Kumpulkan satu jurnal yang membahas FP-Growth

International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org || PP. 24-31

Research of Improved FP-Growth (IFP) Algorithm in Association Rules Mining

M.Sinthuja[1], Dr. N. Puviarasan[2], Dr. P.Aruna[3]
[1, 3](Department of Computer Science and Engineering,Annamalai University, Tamil Nadu)
[2](Department of Computer and Information Science, Annamalai University, Tamil Nadu)

Abstract: *Exploring frequent itemset from huge transactional database has been the most time consuming process of association rule mining.Up-to-date, various algorithms have been popularized in the area of frequent itemset generation. The FP-growth algorithms are the most familiar algorithms. FP-growth algorithm adopts tree structure for storing information producing in longer runtime. FP-growth algorithm faces the problem in complexity of space and time complexity. To develop the efficiency of the FP-growth algorithm, Improvised FP-growth (IFP) algorithm was introduced. In this paper, the benchmark databases considered for comparison are Chess, Connect and Mushroom. It was found out that the IFP-Growth algorithm outperforms FP-growth algorithms for all databases in the criteria of runtime and memory usage.*

Keywords: *Association Rule Mining Algorithm; Data Mining; FP-Growth; IFP-Growth; Minimum Support; Pruning*

I. Introduction

The era of automatized data collection applications and advanced database systems have given rise to tonnes of data for research in data warehouses, repositories and databases [1] [3]. Data should lead to useful information. The concept of datamining is used to extract useful and interesting information such as regularities, patterns, limitations and rules in databases. Data mining generates information about formerly accurate independent itemsets and their connection in a big database.

The items that are represented often in a database is called frequent itemsets. Frequent item sets are a group of items that occur often like notebook and pen in a transaction database. Frequent pattern mining bats for robust relationships in a data set. Frequent item sets may be transactional or relational.

This data industry has given birth to lot of data mining applications that can be applied in retail shopping for product showcasing and in internet for search of common words in pair. These applications work out for all items which have specific attributes-patient/symptoms, restaurants, /menu, internet/keywords, basket/products, railway/timings etc.

Market basket analysis tells about the products that are often bought in pair and it might provide data for promotion strategies. For example, purchase of a shampoo is accompanied with the purchase of conditioner. Promotion of shampoo might increase the sale of the conditioner. Promotion of both the items in combination might increase the profit.

II. Related Work And Contribution

Association rule learning is a predominant and thoroughly studied method for examining rules. R.S. Agrawal et al independently introduced Association rule mining [2] [3] [4]. Discovery of frequent item sets is the most pivotal aspect of data mining. Here frequent itemsets are sets of items that are purchased together frequently in a transaction. Mining frequent itemsets are improved by numerous new methods and an agile data structure is also introduced. Numerous algorithm for frequent itemset mining is discussed elaborately and the performance can be seen in the literature survey of frequent itemset mining [2] [13].

Apriori is one of the earlier algorithms which find frequent itemsets from big transactional databases [2]. It is a process to pinpoint frequent particular items in the transactional database. It in a later stage it extends them to larger itemsets that occur frequently. The frequent itemsets found by Apriori is used to explore association rules. This can be used in the application domain market basket analysis.

It is notable that the average search space for all frequent itemset is huge. Instead of exploring and calculating the support of the itemsets which are certainly impractical, numerous solutions have been introduced [1]. Partitioning and sampling methods of algorithms were introduced. The study includes algorithms like Apriori-TID [20], Apriori-Hybrid [5], Partition [6], Sampling [7], CARMA [8], ECLAT [9], Top-Down [10], FP-Growth [11] among others.

UT-Miner [11] is an advanced Apriori algorithm suited for sparse data. Sparse data are mostly varied and unique. It has a better performance with a structure of array. It is unreliable in terms of runtime and memory usage.

Predictive Apriori algorithm is another form of the association rule mining algorithms [12]. It is different from Apriori algorithm in both support and confidence. In this algorithm support and confidence are united into a single combination called predictive accuracy. {Support, Confidence}=> Predictive Accuracy. This algorithm performs with an increasing support threshold for a number of best rules.

Another achievement in the frequent pattern mining is FP-Growth algorithm. An effective algorithm called FP-Growth which forms a frequent pattern tree construction called FP-Tree [7]. In comparison to Apriori algorithm, candidate patterns are not searched and database is scanned only twice.

III. Proposed Research Work Improvised Fp-Growth Algorithm

In this research paper, an algorithm of Improvised FP-growth is introduced which is an advanced form of FP-growth algorithm. The structure of FP-growth algorithm has only two attributes of item name and count. In the proposed IFP-growth algorithm the structure of the node is modified. It has four attributes of item name, count, node link and flag. Here, node link plays an act of connect nodes with the similar item. It helps to search for particular node quickly. Thus the proposed algorithm quickly traverses the tree in relation to FP-growth algorithm. It reduces the runtime and memory of IFP-growth algorithm. IFP-growth algorithm shortens the database into IFP-tree. It also maintains the data between itemsets. IFP-growth is a divide and- conquer algorithm that uses two stages: build and mine. Structure of proposed IFP-growth algorithm is as follows:

An FP-tree is of three attributes namely:

i. Root node
ii. Child node
iii. Header Table

Attributes of each node are:

i. Item Name
   The names of the item are stored in this fie
ii. Count
   Number of times the item occur are updated in this field.
iii. Node Link
   Node-link that points the node that contains the same item name
iv. Flag
   The branch information are noted in this field.

Each entry of the item in header table consists of two Characteristics:

i. Item-name
ii. Head of node-link

Transaction Database is displayed in Table 1. The database is scanned early and each item in the database is counted. Items are assembled in descending sequence of support count. Items with lesser support value are pruned.

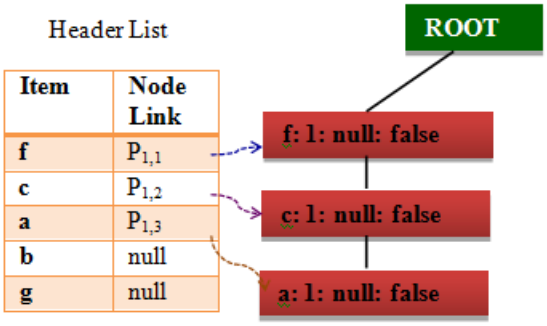
Table I: Transaction Database

TID	Item bought	Ordered item
100	d, c, a, f	f, c, a
200	g, c, a, f, e	f, c, a, g
300	b, a, c, f, h	f, c, a, b
400	g, b, f	f, b, g
500	c, b	c, b

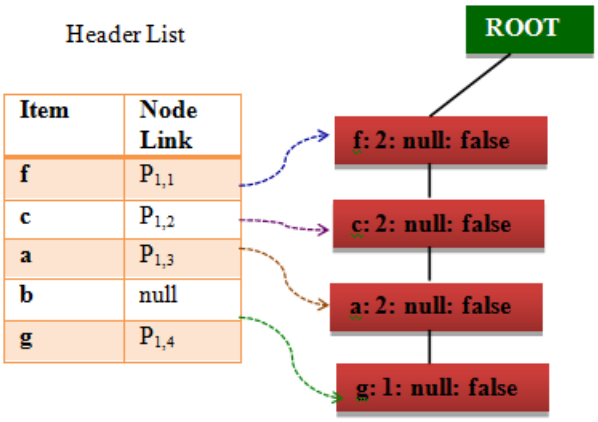
Table II: Prioritize the items

Items	Support
{f}	4
{c}	4
{a}	3
{b}	3
{g}	2
{d}	1
{e}	1
{h}	1

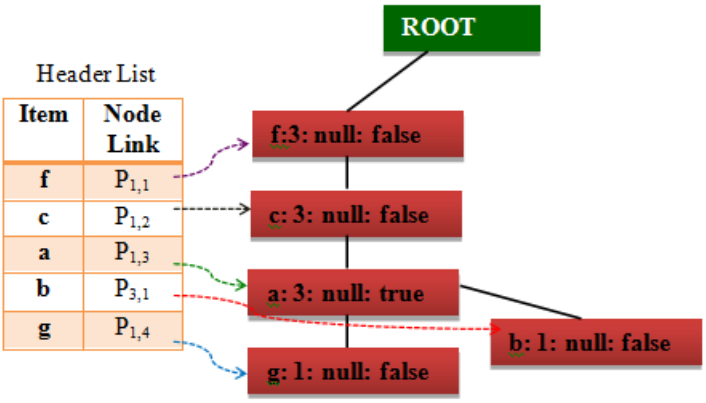
Step 1: Insertion of itemset {f, c, a}



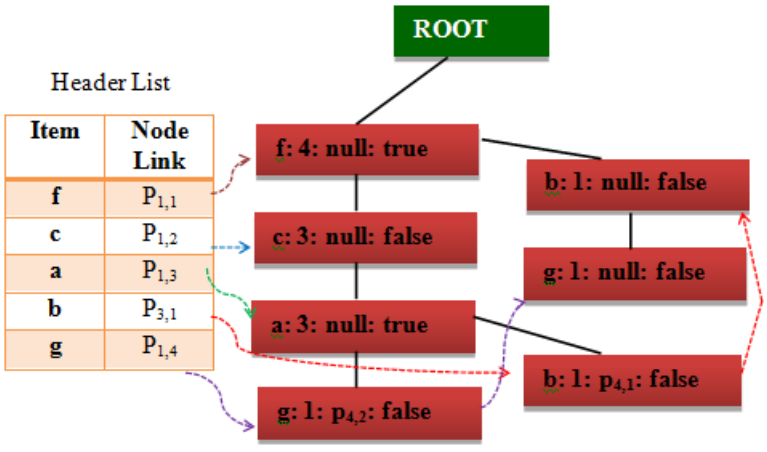
Step 2: Insertion of itemset {f, c, a, g}



Step 3: Insertion of itemset {f, c, a, b}



Step 4: Insertion of itemset {f, b, g}



Step 5: Insertion of itemset {c, b}

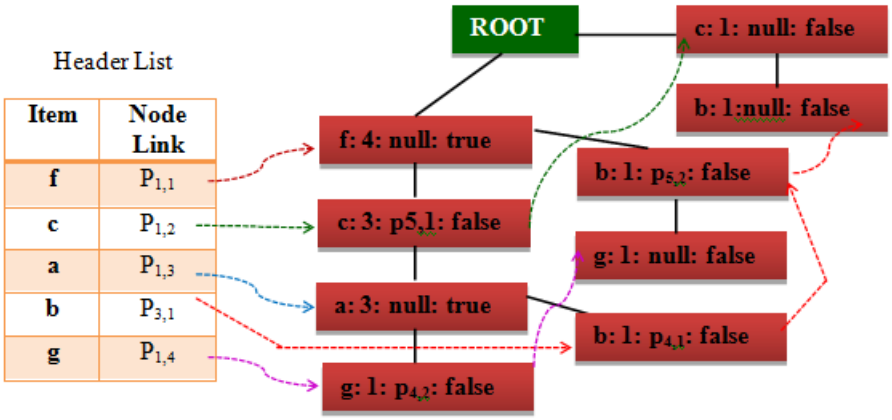


Fig. 1. Construction of IFP-tree

IFP-tree is constructed. Early, the proposed IFP-growth algorithm generates a node named 'root' with the tag value 'null'. For the second scan, it starts generating nodes for each transaction. For example, the first transaction of the database contains three items {f, c, a} based on the arrangement of descending order, building the early wing {(f:1) (c:1) (a:1)} in IFP-tree with three nodes. Here, item 'f' is the child of "root", item 'c' links to item 'f', item 'a' links to item 'c'. Now, the node link to 'null' is updated, as it is not related to any link and the flag value to 'false' is tagged, as it is not related to any branch nodes. Node link in the header table has to be updated for each item when an item is inserted in the IFP-tree. The second transaction contains four items {(f: 1) (c: 1) (a: 1) (g:1)} where four nodes are generated. Here, this branch shares the prefix 'f', 'c', 'a' with the existing path of transaction '01'. In this way the count of the nodes are incremented to '2'. For item 'g' new node is generated from item 'a'. The node link of item 'f',' c', 'a', 'g' to 'null' is updated as it does not have any link and fix its flag value. Third transaction of the database contains {f, c, a, b}. It is similar to that of previous transaction. Fourth transaction of the database contains {f, b, g}. In this scenario, the item 'f' shares its path from first transaction and increment its count to '4'. Node link value is noted down and flag value is fixed to true as it has branch node. For the insertion of item 'b' and 'g' a new node is generated from item 'f' where item 'b' is linked to item 'f' and item 'g' is linked to item 'b'. It is notable that the node link of item 'g' has to be updated to $P_{4,2}$and item 'b' to $P_{4,1}$ respectively along with an update in flag value. In case of fifth transaction the database contains {c, b}. In this scenario, new node has to be generated from root node as it does not share its path with existing transactions. Here, item 'c' is linked to 'root' and item 'b' is linked to item 'c'. The node links for all items are updated. The complete IFP-tree is portrayed in Fig 1.

Table III. Generated Frequent Patterns

Items	Conditional pattern base	ConditionalFP-tree	Frequent patterns generated
h	{f, c, b, a:1}	-	**No**
e	{f, c, a, g:1}	-	**No**
d	{ f, a, c:1}	-	**No**
g	{f,b:1} {f, c, a:1}	{f:2}	**{f, g:2}**
b	{c:1}{f:1} {f, c, a:1}	{f, c:2}	**{f, c, b:2}**
a	{f, c:3}	{f,c:3}	**{f,c,a:3}**
c	{f:3}	{f:3}	**{f, c:3}**
f	-	-	**No**

The system of exploring frequent itemsets from IFP-tree is from last to first using "Bottom Up" approach to explore frequent itemsets. Generated frequent itemsets is displayed in Table III. FP growth algorithm is great in achieving three important aspirations: database is scanned only two times and computational cost is decreased dramatically, Candidate itemsets are not generated. The search space is reduced by using divide and conquer approach. The limitation of FP-growth algorithm is crucial to use in incremental mining, as new transactions are added to the database.

IV. Experimental Results Analysis

Inthisexperiment,theperformanceofimproved FP-growthiscomparedwithFP-growthalgorithms.Inordertoshowthattheexperimentsarereasonable,theperformancebasedontwoimportantcriteria:runtimeandmemoryusage

A. Data Description

Theproposed algorithm iscarried outusingstandarddatasetsofvariousdomainsnamelyChess, Connect, Mushroom.TableIVdepictstheentireinformationofdatasetsusedfortheimplementationpurpose.

TABLE IV.DatasetDescription

Datasets	File Size (kb)	Tuples	Items	Type
Chess	335	3196	75	Dense
Mushroom	9105	8124	58	Dense
Connect	558	67557	116	Dense

B. Runtime Assessment

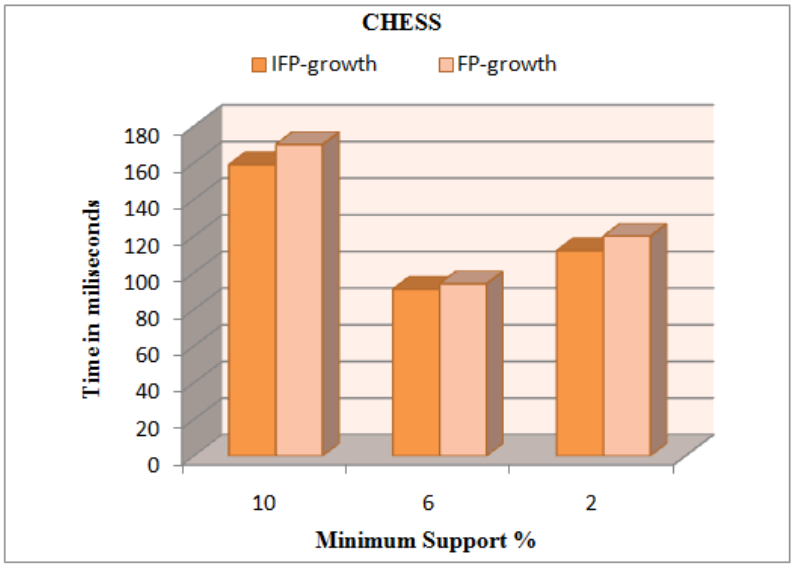


Fig. 2. Runtime of proposed IFP-growth algorithm

From theFig.2y-axisdisplayruntimeinmillisecondandx-axisdisplaythedifferent minimumsupportvalues.Itincludestheresult ofexecutiontimeof proposed IFP-growthandFP-growthalgorithmsonthedatasetChesswithdifferentminimumsupport.

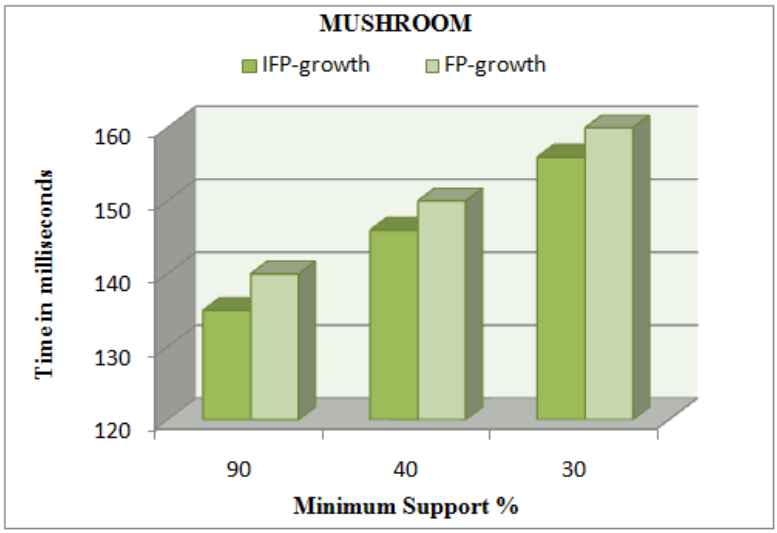


Fig. 3. Runtime of proposed IFP-growth algorithm

Runtimemeansthetotalexecutiontimebetweeninputandoutput.Here,theruntimeofIFP-growthis170msandFP-growthis150msforminimumsupport10.ItshowsthatIFP-growthisquickerthanFP-growthalgorithm.Astheminimum supportdiminishestheruntimeofFP-growthincreaseswhencomparedtoproposed IFP-growthalgorithm.

With reference to the Fig.3, itshowstheresultofrunningtimeof proposed IFP-growthandFP-growthalgorithmsonthedatasetMushroom withdifferentminimumsupport.TheruntimeofFP-growthis160msandIFP-growthis156msforminimumsupport30. In some cases, the processing to time of FP-growth is more or less similar to the proposed IFP-growth algorithm. Even though the proposed IFP-growth algorithm is better than FP-growth algorithm.

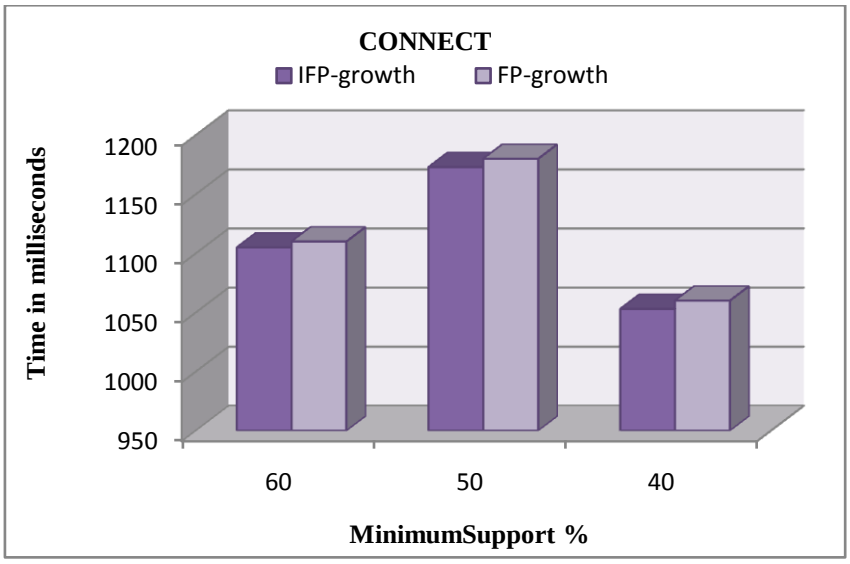


Fig. 4. Runtime of proposed IFP-growth algorithm

The graph illustrated in Fig 4 shows theruntimeof all algorithms for the dataset Connect.FP-growthis1110msandIFP-growthis1105msfor the dataset Connect with minimumsupport60.ItshowsthatproposedIFP-growthisrapidthanFP-growthalgorithm.IFP-growthshowsit'sthe beston thedatasetsConnectforbothlowerandhigherminimumsupportwhencomparedtoFP-growthalgorithm.

C. Memory Consumption

Fig.5portraysthecomparativeresultsof IFP-growthandFP-growthalgorithmincaseofmemoryusage for the dataset Chess.Fromthegraphlongitudinalaxisshowsthememoryin MBand latitudinalaxisshowsthedifferentsupportthresholdvalues.ThememoryusageofFP-growthis 107.86 MBandproposed IFP-growthis101MB forminimumsupport 6 .ItshowsthatIFP-growthisrapidthanFP-growthalgorithm.IFP-growthrequireslessmemorythanFP-growthalgorithm while the minimum support is higher. When the minimumsupport is lower memoryrequirementforIFP-growthislesswhencomparedtoFP-growthalgorithm.

FromFig.6memoryusageofFP-growthalgorithmishigherascomparedtoIFP-growthalgorithmsincaseofbothhigherand lower minimumsupportforthedataset Mushroom. Here, the consumption of memory of proposed IFP-growth algorithm is 120MB and 123.9MB for FP-growth algorithm with the minimum support threshold values of 40.

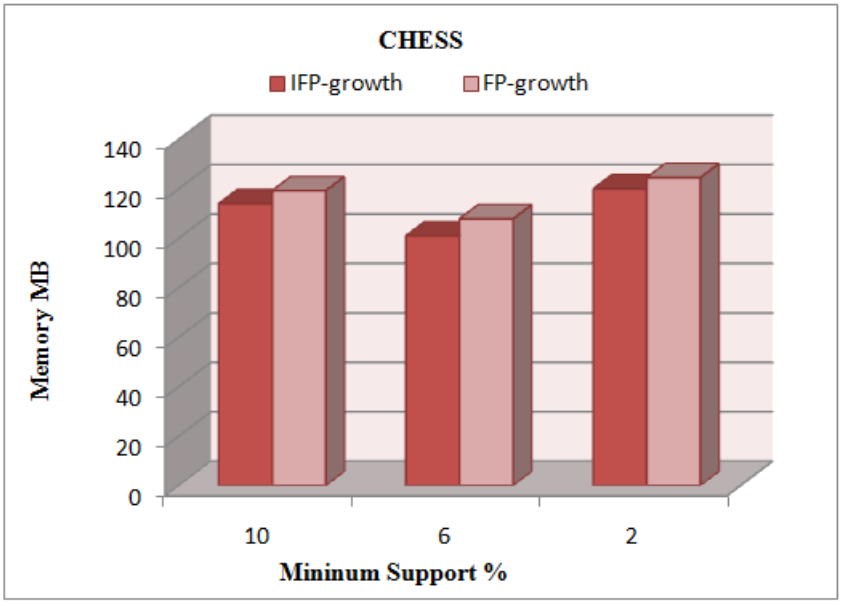


Fig. 5. Memory usage of proposed IFP-growth algorithm

From the graph it is come to know that theIFP-growthalgorithmgivesbetterperformanceascomparedtoFP-growthalgorithm.

The experimental results of memory usage all algorithms for the dataset Connect are reported in Fig 7. FP-growth algorithm consumes lot of memory when compared to other algorithm. Meanwhile, IFP-growth algorithm consumes lower memory space when compared to FP-growth algorithm. It is efficient in consumption of memory. For the minimum support threshold value of 60 the IFP-growth consumes 99MB FP-growth consumes 103.59MB. It is clearly observed from the graph that the proposed IFP-growth algorithm is efficient than FP-growth algorithm.

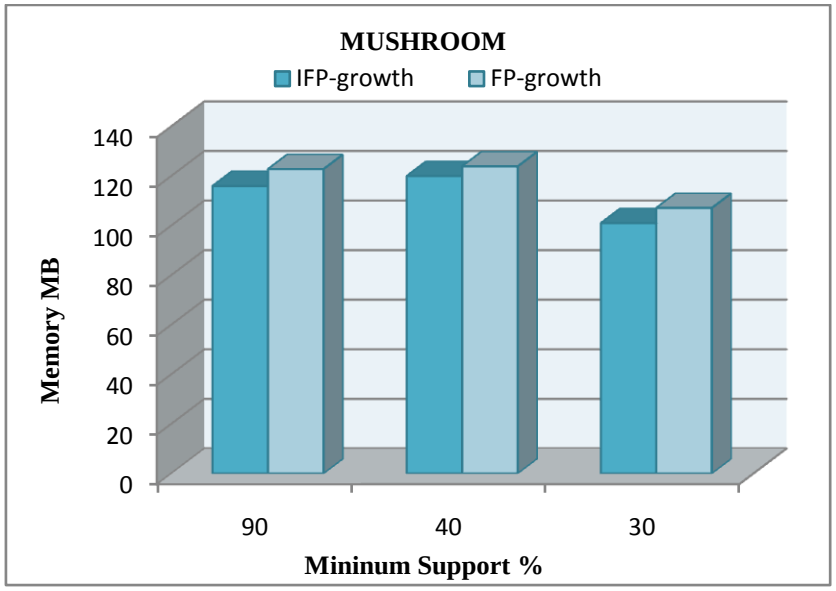


Fig. 6. Memory usage of proposed IFP-growth algorithm

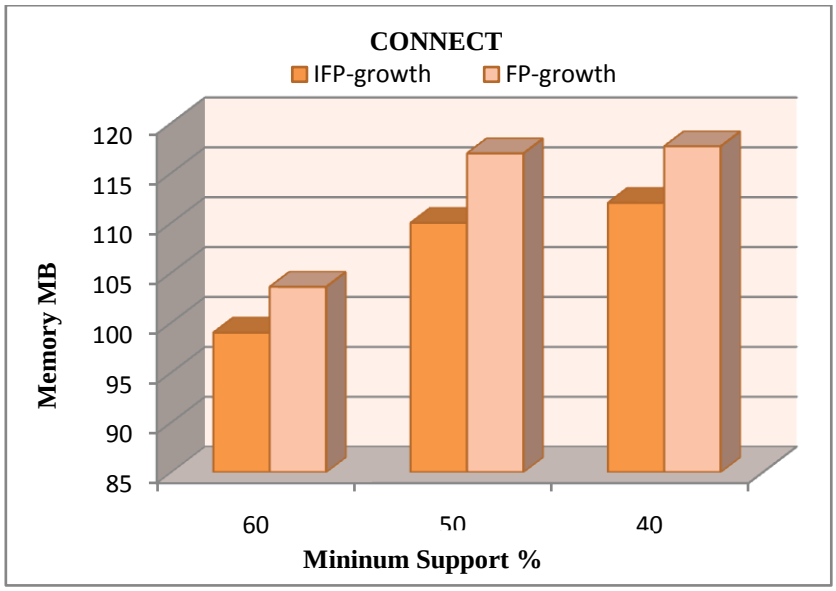


Fig. 7. Memory usage of proposed IFP-growth algorithm

V. Conclusion

In this research paper, improvised version of Frequent Patten growth (IFP) is proposed for exploring frequent itemsets. In the light of results observed above from the proposed Improvised Frequent Pattern growth algorithm. It has been finalized that the proposed Improvised Frequent Pattern growth algorithm is better than the FP-growth algorithm for large benchmark datasets like Chess, Mushroom and Connect for mining frequent itemsets. It is determined from the experimental evaluation that the proposed Improvised FP-growth is a suitable algorithm for dense datasets with respect to speed and memory consumption. It is more efficient for dense datasets. The proposed Improvised Frequent Pattern growth algorithm is found to be the best algorithm than FP-growth algorithm in terms of processing time and consumptionof memory.

Acknowledgment

The authors would like to thank the Department of Computer Science and Engineering, Annamalai University for providing the lab facilities. We gratefully acknowledge the funding agency, the University Grant Commission (UGC) of the Government of India, for providing financial support. Also, authors would like to thank to Dr. N. Puviarasan and Dr. P.Aruna for guiding to do new analysis for revising the article.

REFERENCES

[1] S. Langenfeld, *"CRM and the Customer Driven Demand Chain,"* vol. *2004*, 2004.
[2] R. Agarwal, T. Imielinski, and A. Swami, "*Mining Association Rules Between Sets of Items in Large Datasets,"* in Proceedings of the ACM SIGMODConference on Management of Data. Washington, D.C.: ACM, 1993, pp. 207-216.
[3] J. Schafer, J. Konstan, and J. Riedl, "*Electronic Commerce Recommender Applications*," Journal ofData Mining and Knowledge Discovery, *vol. 5*, pp. 115-152, 2001.
[4] W. Lin, "*Association Rule Mining for Collaborative Recommender Systems*," in Computer Science.
[5] R. Agrawal and R. Srikant, "*Mining Sequential Patterns*," in Proceedings of the 11th IEEEInternational Conference on Data Engineering. Taipei, Taiwan: IEEE, 1995.
[6] A. Savasere, E. Omiecinski, and S. B. Navathe, "*An Efficient Algorithm for Mining Association Rules in Large Databases*," in Proceedings of the 21ndInternational Conference on Very Large Databases. Zurich, Switzerland, 1995, pp. 432-444.
[7] H. Toivonen, "*Sampling Large Databases for Association Rules,"* in Proceedings of the 22nd International Conference on Very Large Databases. Mumbai, India, 1996, pp. 134-145.
[8] C. Hidber, "*Online Association Rule Mining,"* in Proceedings ACM SIGMOD InternationalConference on Management of Data. Philadephia, Pennsylvania: ACM, 1999, pp. 145-156.
[9] M.J.Zaki, "*Scalable Algorithms for Association Mining*," IEEE Transactions on Knowledge and DataEngineering, *vol. 12*, pp. 372-390, 2000.
[10] J. Han, J. Pei, and Y. Yin, "*Mining Frequent Patterns Without Candidate Generation,"* in Proceedings of the2000 ACM SIGMOD International Conference onManagement of Data. Dallas,TX: ACM, 2000.
[11] J. Hipp, U. Guntzer, and G. Nakhaeizadeh, *"Algorithms for Association Rule Mining -- A general Survey and Comparision*," in Proceedings of ACMSIGKDD, *vol. 2*: ACM, 2000, pp. 58-64.
[12] M.Sinthujaet al, *"Comparative Analysis of Association Rule MiningAlgorithms in Mining Frequent Patterns*"International Journal of Advanced Research in Computer Science, *vol 8 (5)*, May-June 2017,1839-1846.
[13] B. Goethals, "*Survey on Frequent Pattern Mining," vol. 2004*, 2003.

Jurnal Sarjana Teknik Informatika
Volume 3 Nomor 1, Februari 2015
e-ISSN: 2338-5197

ANALISIS PERBANDINGAN ALGORITMA APRIORI DAN ALGORITMA HASH BASED PADA MARKET BASKET ANALYSIS DI APOTEK UAD

[1]Orisky Sitra Arifah Destiyati (08018359), [2]Eko Aribowo (0006027001)

[1,2]Program Studi Teknik Informatika
Universitas Ahmad Dahlan
Prof. Dr. Soepomo, S.H., Janturan, Umbulharjo, Yogyakarta 55164
[1]Email: oriskysad@gmail.com
[2]Email: ekoab@tif.uad.ac.id

ABSTRAK

Dalam aturan asosiasi data mining, terdapat beberapa algoritma diantaranya adalah algoritma apriori, algoritma hash base, algoritma fp growth, dan algoritma fuzy c covering. Masing-masing algoritma memiliki kelebihan dan kekurangan. Penelitian ini bertujuan untuk membandingkan kedua algoritma aturan asosiasi yaitu algoritma apriori dan algoritma hash based dengan studi kasus pemesanan obat di Apotek UAD dimana apotek tersebut secara periodik diharuskan menyuplai obat untuk mahasiswa yang akan melaksanakan KKN. Oleh karena itu, penelitian ini juga dilaksanakan untuk mengetahui kecenderungan mahasiswa KKN UAD dalam memesan obat untuk melaksanakan KKN, sehingga nantinya manager akan menyuplai obat-obat tertentu berdasarkan pemesanan mahasiswa yang akan melaksanakan KKN.

Penelitian dilakukan dengan membangun sebuah sistem yang menghasilkan waktu penambangan data dari masing-masing algoritma. Dari hasil pencatatan waktu tersebut dapat digunakan untuk menganalisis kecepatan masing-masing algoritma dalam menambang data. Teknik pengumpulan data dilakukan dengan wawancara dan observasi langsung ke Apotek UAD. Data diolah dan dikelompokkan berdasarkan merk masing-masing obat.

Hasil penelitian menunjukkan bahwa, pada beberapa uji coba yang telah dilakukan, diketahui algoritma apriori memiliki waktu yang lebih cepat dalam menambang data dibandingkan algoritma hash based. Dan telah diketahui kecenderungan pemesanan obat mahasiswa KKN, dengan nilai support lebih dari 30% dan nilai confidence lebih dari 70%.

Kata kunci : *Algoritma Apriori, Algoritma Hash Based, Market Basket Analysis*

 e-ISSN: 2338-5197

1. PENDAHULUAN

Data mining adalah suatu proses menemukan hubungan yang berarti, pola, dan kecenderungan dengan memeriksa dalam sekumpulan besar data yang tersimpan dalam penyimpanan dengan menggunakan teknik pengenalan pola seperti teknik statistic dan matematika [1]. Data mining menggunakan *discovery based* yaitu dimana pencocokan pola dan algoritma digunakan untuk menentukan relasi-relasi kunci di dalam data yang akan dieksplorasi. Dalam data mining terdapat beberapa teknik penambangan data,salah satunya adalah *association rule mining* atau dikenal dengan aturan asosiasi. *Assocation rule mining* adalah teknik *mining* untuk menemukan aturan assosiatif antara suatu kombinasi *item*. Dalam aturan asosiasi sendiri terdapat beberapa algoritma diantaranya adalah algoritma *apriori*, algoritma *hash based*, algoritma *fp growth*, dan algoritma *fuzy c covering*. Masing-masing algoritma memiliki kelebihan dan kekurangan.

Dalam penelitian ini, dilakukan perbandingan antara dua algoritma aturan asosiasi, yaitu algoritma *apriori* dan algoritma *hash based*. Parameter yang digunakan adalah kecepatan masing-masing algoritma dalam menambang data. Selain itu, akan dilakukan penelitian pola konsumsi obat, dengan objek adalah data transaksi penjualan obat di sebuah apotek. Sebagai pembandingnya akan didasarkan pada pola pemesanan obat oleh mahasiswa UAD yang akan melaksanakan KKN. Dari hasil informasi tersebut, diharapkan dapat membantu pengelola apotek, dalam hal ini adalah *manager* dalam menentukan kebijakan-kebijakan yang nantinya dapat menguntungkan apotek.

2. KAJIAN PUSTAKA

2.1 Kajian Terdahulu

Penelitian yang dilakukan, mengacu pada penelitian terdahulu yang berjudul "Analisis *Market Basket* dengan Algoritma *Apriori* dan *FP-Growth*" [3]. Penelitian ini bertujuan untuk membandingkan kompleksitas waktu antara kedua algoritma yang digunakan. Dari penelitian ini, kesimpulan yang didapat adalah algoritma *apriori* membutuhkan waktu komputasi yang lama untuk mendapatkan *frequent itemsets* karena berulang kali melakukan pemindaian data. Sedangakan algoritma *fp-growth* hanya membutuhkan dua kali *scanning database* dalam mencari *frequent itemset* sehingga waktu yang dibutuhkan pun menjadi relatif singkat dan efisien.

Dalam penelitian dengan judul "Penggunaan Metode *Fuzzy C-Covering* untuk Analisa *Market Basket* pada Supermarket" [4], dibahas mengenai bagaimana performasi dan rule atau aturan yang dihasilkan algoritma *fuzzy c-covering.* Kesimpulan yang didapat dari kesimpulan ini adalah semakin kecil *minimum support* dan *confidence* yang ditentukan, semakin banyak pula *rule* yang dihasilkan, namun waktu yang diperlukan relatif lebih lama dibandingkan jika *minimum support* lebih besar. Selan itu ditemukan pula bahwa semakin tinggi jumlah kombinasi yang dicari, waktu yang dibutuhkan relatif lebih cepat. Hal ini terjadi karena semakin lama item yang tersaring semakin banyak sehingga kombinasi yang dicari semakin sedikit.

2.2 Landasan Teori

2.2.1. Algoritma Apriori

Algoritma *apriori* menggunakan *knowledge* mengenai frequent itemset yang telah diketahui sebelumnya, untuk memproses informasi selanjutnya. Pada algoritma *apriori* untuk menentukan kandidat-kandidat yang mungkin muncul dengan cara memperhatikan minimum support.

Adapun dua proses utama yang dilakukan dalam algoritma *apriori*, yaitu [3]:

a. *Join* (Penggabungan)
   Pada proses ini, setiap *item* dikombinasikan dengan *item* yang lainnya sampai tidak terbentuk kombinasi lagi.
b. *Prune* (Pemangkasan)
   Pada proses ini, hasil dari *item* yang telah dikombinasikan tadi lalu dipangkas dengan menggunakan *minimum support* yang telah ditentukan *user*.

Algoritma *apriori* dibagi menjadi beberapa tahapan yang disebut *iterasi*. Pada *iterasi* pertama dihasilkan pola frekuensi tinggi dengan panjang 1-*itemset*. Pada *iterasi* pertama pula, nilai *support* dari setiap *item* dihitung dengan melihat *database* yang ada. Setelah didapat nilai *support*, *item* yang memiliki *support* di atas *minimum support* (minsup) dipilih sebagai acuan untuk *iterasi* selanjutnya. *Iterasi* kedua menghasilkan 2-*itemset* yang tiap setnya memiliki dua *item*. Tiap kandidat 2-*itemset* dihitung nilai *support*nya dengan melihat *database* sebelumnya. *Support* disini dimaksudkan untuk jumlah transaksi dalam *database* yang mengandung kedua *item* dalam kandidat 2-*itemset*. Sama seperti 1-*itemset*, setelah didapat nilai *support*nya, maka kandidat 2-*itemset* yang memenuhi minsup dapat ditetapkan sebagai 2-*itemset* yang merupakan pola frekuensi tinggi.

Untuk selanjutnya pada *itersai* ke-k dapat dibagi menjadi beberapa bagian [5] :

a. Pembentukan kandidat *itemset*
   Kandidat k-*itemset* dibentuk dari kombinasi (k-1)-*itemset* yang didapat dari *iterasi* sebelumnya.
b. Penghitungan *support* dari tiap kandidat k-*itemset*
   Support dari tiap kandidat k-*itemset* didapat dengan melakukan *scan database* untuk menghitung jumlah transaksi yang memuat semua *item* di dalam kandidat k-*itemset* tersebut.
c. Tetapkan pola frekuensi tinggi
   Pola frekuensi tinggi yang memuat k-*item* atau k-*itemset* ditetapkan dari kandidat k-*itemset* yang supportnya lebih besar dari minsup.
d. Bila tidak didapat pola frekuensi tinggi baru, maka seluruh proses dihentikan. Bila tidak, maka k ditambah satu dan kembali ke bagian a.

2.2.2. Algoritma Hash-Based

Algoritma *Hash Based* menggunakan teknik *hashing* untuk menyaring keluar *itemset* yang tidak penting untuk pembangkitan *itemset* selanjutnya. Ketika *support count* untuk kandidat k-*itemset* dihitung dengan menelusuri basis data, algoritma *hash based* mengumpulkan informasi mengenai (k+1)-*itemset* dengan cara seluruh kemungkinan (k+1)-*itemset* di*hash* ke dalam *hash table* dengan menggunakan fungsi *hash* (menggunakan sebuah bilangan prima untuk operasi modulo) [5].

Algoritma *Hash Based* terbagi menjadi tiga bagian utama yang masing-masing bagian melakukan proses yang berbeda. Bagian pertama akan menghasilkan kandidat 1-*itemset* yang disebut C_1 dan *large* 1-*itemset* yang disebut L_1 dari basis data. Untuk kandidat 1-*itemset*, seluruh transaksi ditelusuri untuk menghitung *support* count dari *itemset* ini. Pada tahap ini *hash tree* untuk C_1

 e-ISSN: 2338-5197

dibangun dengan tujuan mengefisienkan penghitungan *support count*. Pada bagian ini juga algoritma akan membangun *hash table* (dengan fungsi *hash*) untuk 2 *itemset* yang akan berguna mengurangi banyaknya kandidat *2-itemset* C_2.

Pada bagian kedua, kumpulan kandidat *itemset* C_k dibangkitkan berdasarkan *hash table* yang telah dibuat pada *iterasi* sebelumnya. Lalu ditentukan *large itemset* L_k dan mengurangi ukuran basis data untuk pembangkitan *itemset* selanjutnya. Bagian algoritma ini terbagi menjadi dua fase. Fase pertama untuk membangkitkan kandidat k-*itemset* berdasarkan *hash* table. Fase kedua akan menghitung *support* pada kandidat *itemset* dan mengurangi ukuran dari setiap transaksi.

Bagian ketiga sama seperti bagian kedua tetapi tidak menggunakan *hash table* sehingga mirip dengan algoritma *apriori*. Bagian kedua dilakukan selama nilai *hash* buket lebih besar dari *minimum support*. Setelah batasan ini terlewati, algoritma *hash based* diganti dengan algoritma *apriori* [2].

Berikut penjelasan dari rumus *hash table* (*hash bucket*) [5].

H(X,Y) =[(*order of* X)* penambahan ctr *hash table* + (*order of* y)] mod prima

Keterangan :

- *Order of* X adalah perwakilan nilai X
- Penambahan ctr *hash table* adalah nilai modulus bilangan prima, apabila terjadi *collision* nilai tersebut ditambah 1 (+ 1) sampai tidak terjadi *collision*.
- *Order of* Y adalah perwakilan nilai Y
- Prima adalah bilangan prima yang terdekat dan yang lebih besar dari jumlah kombinasi 2-*itemet* (C2).

3. METODE PENELITIAN

Subyek penelitian dalam penelitian ini adalah dua algoritma data mining yaitu algoritma *apriori* dan algoritma *hash based* juga data transaksi obat di apotek uad. Adapun alat yang digunakan dalam penelitian adalah Borland Delphi 7 dan untuk database menggunakan My SQL. Data transaksi obat tersebut diperoleh langsung dari apoteker dengan melakukan wawancara dan observasi data obat. Selain itu dilakukan studi kepustakaan dengan menggunakan buku-buku dan referensi dari penelitian-penelitian yang telah dilakukan sebelumnya terhadap algoritma-algoritma dalam aturan asosiasi untuk membandingkan data dan fakta-fakta yang ada guna memperkuat landasan teori.

4. HASIL DAN PEMBAHASAN

Data utama yang digunakan pada penelitian ini adalah transaksi data obat yang terdiri dari 129 transaksi. Pada Tabel 1 dapat dilihat sampel data transaksi obat di apotek UAD. Sebelumnya data telah diinisiasi sehingga tidak memakan banyak ruang.

Tabel 1 : Tabel Transaksi Obat Apotek UAD

Transaksi	Item
1	BS,KS,ALK,PCM,AM,DMC,CAV,OBH,HP,NDT,VO,PMG,VIT C,SG,MKP,CTM,BP,MFB
2	AM,PCM,MKP,EVC,PND,PMG,BS,HP,BKP,SKH,PRM,NDT,T AM,KMK,SGB,KP,BP
3	MS,BS,HP,DCG,VIC,RHT,TAM,MYL,AS,BM,SGB,MKP,PND,

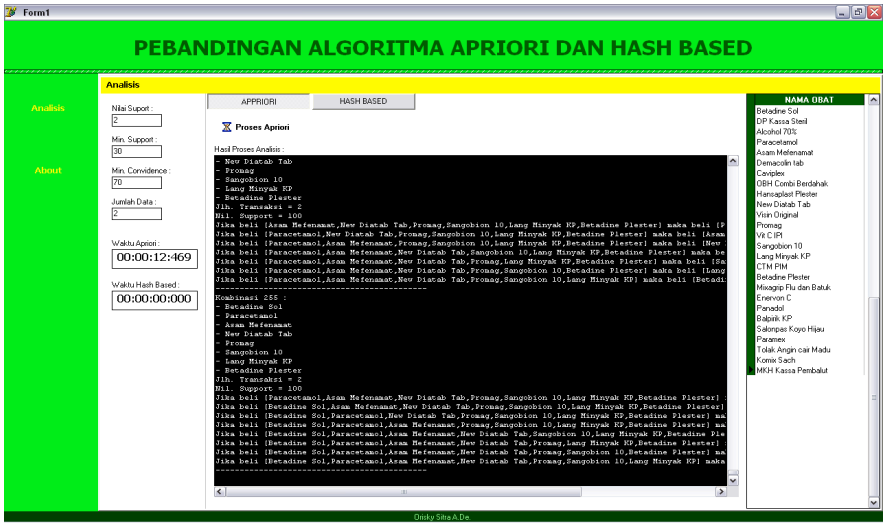
 e-ISSN: 2338-5197

	ECS,BP
4	MKP,FTG,MFB,PND,BS,ASK,BP,KS,HP,MS,PMG,AM,NDT,SGB,OBH
5	MKP,MYL,ULT,GGB,BS,ASM,CTS,RTD,EPX,PCM,OBH,MK,NETS,AM,CAV,SYN,RHT,BP,DP

Setelah dilakukan *running* program, maka dihasilkan output sebagai berikut.

4.1. Hasil Algoritma *Apriori*

Dalam form ini terdapat parameter nilai *support* (f), nilai *minimum support*, nilai *minimum confidence*, dan jumlah data. Parameter tersebut juga akan digunakan pada analisis algoritma *hash based*. Data yang ditampilkan pada form ini merupakan keseluruhan data yang memiliki nilai *support* dan *confidence* sebesar 100%, mulai dari kombinasi pertama hingga kombinasi terakhir yang dapat dilakukan. Hal tersebut mengindikasikan bahwa data obat tersebut adalah obat yang sering dipesan oleh mahasiswa dalam pelaksanaan KKN. Hasil penambangan data manggunakan algoritma *apriori* dapat dilihat pada Gambar 1.

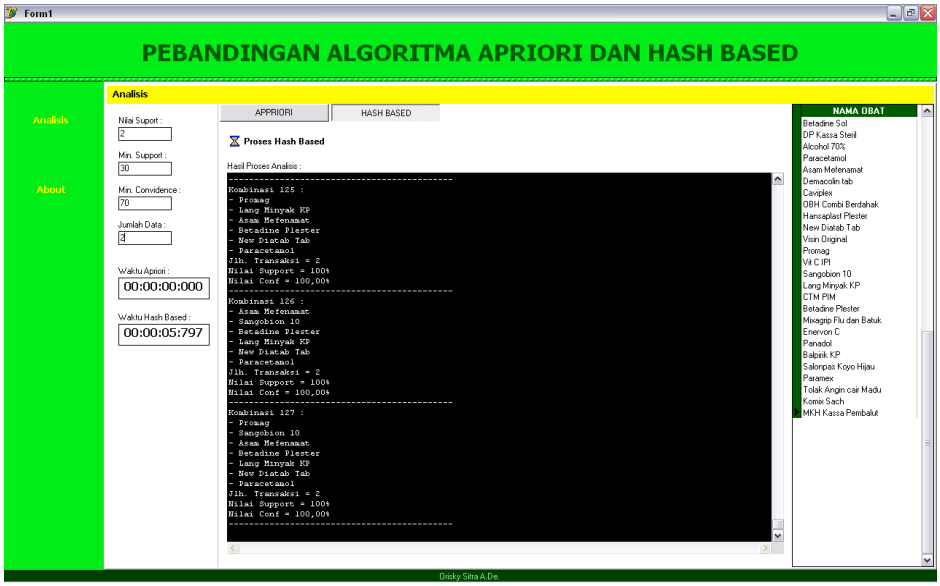


Gambar 1: Form Analisis Algoritma *Apriori*

4.2. Hasil Algoritma *Hash Based*

Sama seperti pada algoritma *apriori*, Dalam form ini terdapat parameter nilai *support* (f), nilai *minimum support*, nilai *minimum confidence*, dan jumlah data. Parameter tersebut juga akan digunakan pada analisis algoritma *hash based*. Data yang ditampilkan pada form ini merupakan keseluruhan data yang memiliki nilai *support* dan *confidence* sebesar 100%, mulai dari kombinasi pertama hingga kombinasi terakhir yang dapat dilakukan. Hal tersebut mengindikasikan bahwa data obat tersebut adalah obat yang sering dipesan oleh mahasiswa dalam pelaksanaan KKN. Hasil dari penambangan data menggunakan algoritma *hash based* dapat dilihat pada Gambar 2 di bawah ini.

 e-ISSN: 2338-5197



Gambar 2: Form Analisis Algoritma *Hash Based*

4.3. Analisis Hasil

Hasil dari penelitian ini adalah dengan menjalankan sistem yang telah dibangun untuk melakukan penghitungan waktu dari kedua algoritma yang digunakan. Dengan menggunakan nilai minimun yang telah ditentukan sebesar 8 dan 9, hasil dari *running* program dapat dilihat pada Tabel 2 dan Tabel 3 di bawah ini.

Tabel 2 : Hasil Uji Coba I dengan nilai ϕ = 8

Jumlah Data	Waktu Apriori	Waktu Hash Based
2	01 : 19 : 547	03 : 31 : 390
5	00 : 01 : 844	00 : 16 : 156
8	00 : 00 : 297	00 : 30 : 406
48	00 : 02 : 183	01 : 38 : 764
53	00 : 02 : 946	02 : 56 : 218
58	00 : 04 : 527	05 : 20 : 813
96	00 : 05 : 642	06 : 51 : 219
112	00 : 06 : 108	07 : 38 : 813
129	00 : 06 : 265	08 : 54 : 421

Tabel 3 : Hasil Uji Coba II dengan nilai ϕ = 9

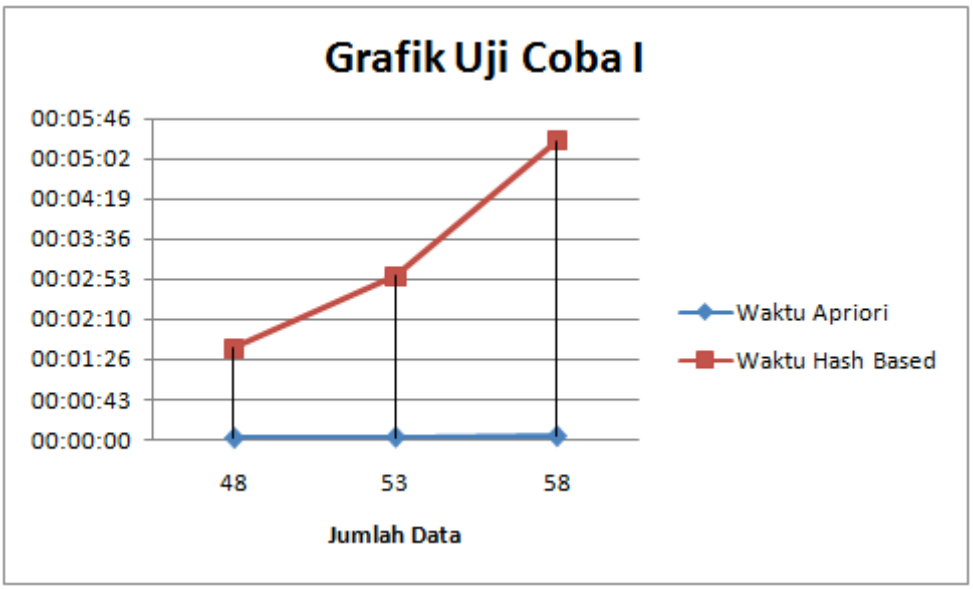
Jumlah Data	Waktu Apriori	Waktu Hash Based
2	01 : 58 : 219	05 : 00 : 235
5	00 : 01 : 906	00 : 22 : 656
8	00 : 00 : 328	00 : 37 : 937
48	00 : 06 : 824	08 : 19 : 590

 e-ISSN: 2338-5197

53	00 : 07 : 540	08 : 58 : 942
58	00 : 09 : 754	09 : 34 : 670
96	00 : 10 : 183	10 : 12 : 408
112	00 : 10 : 920	12 : 31 : 917
129	00 : 12 : 057	12 : 58 : 672

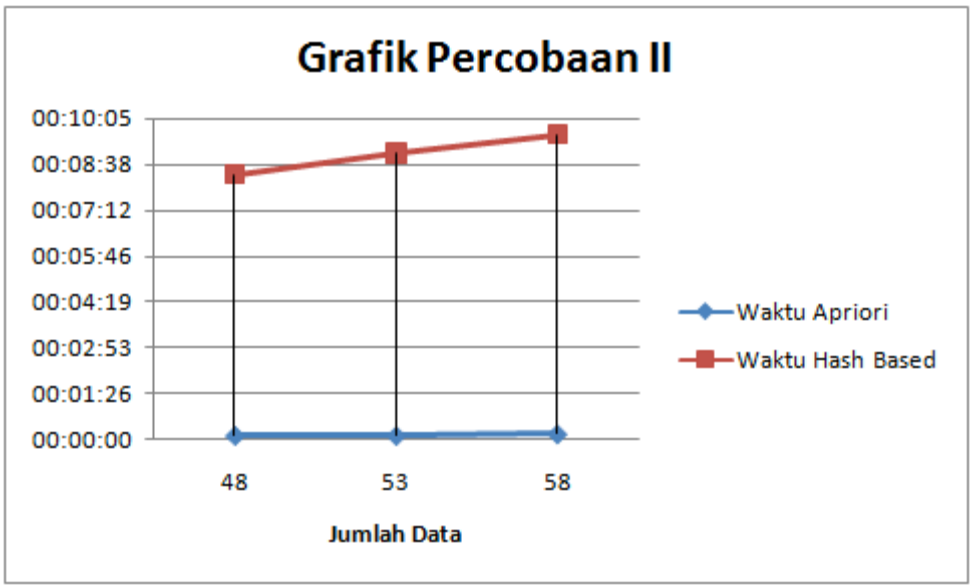
Dapat dilihat dari kedua percobaan di atas waktu yang dibutuhkan algoritma *apriori* lebih cepat dibandingkan waktu yang dibutuhkan algoritma *hash based*. Pada algoritma *apriori* waktu yang digunakan untuk penambangan data menunjukkan masih berada di hitungan detik, sedangkan pada algoritma *hash based* waktu menunjukkan berada di hitungan menit.

Selain itu, dari hasil kedua percobaan yang dapat dilihat pada Tabel 2 dan Tabel 3, dibuat juga grafik yang mewakili masing-masing percobaan. Grafik dapat dilihat pada Gambar 3 dan Gambar 4 di bawah ini.



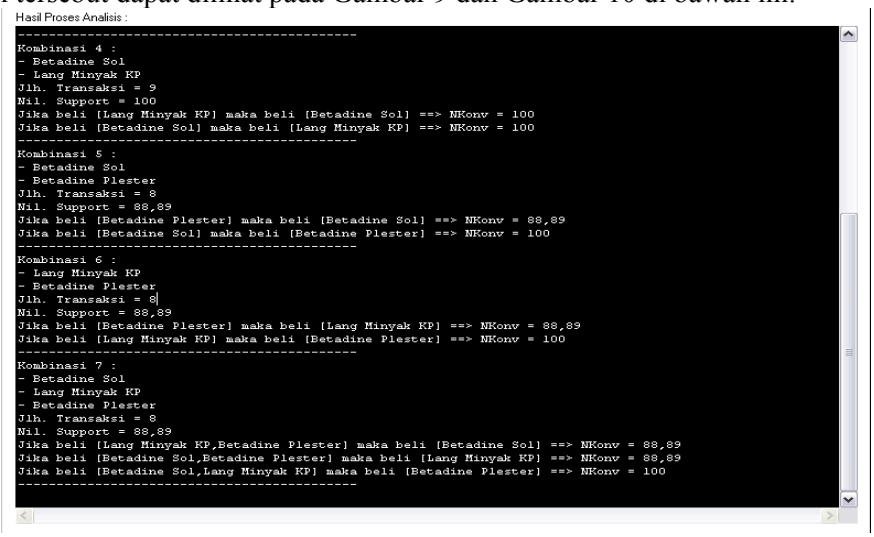
Gambar 3 : Grafik Percobaan I

 e-ISSN: 2338-5197



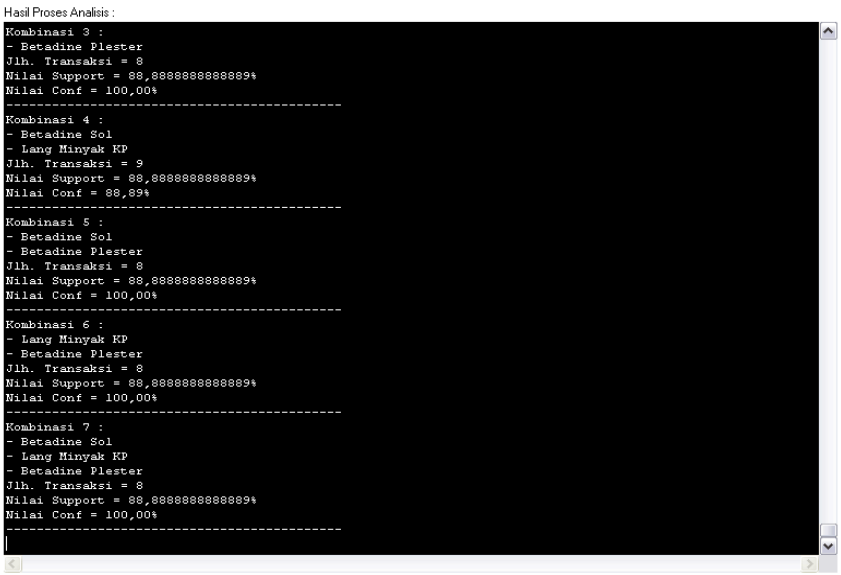
Gambar 4 : Grafik Percobaan II

Selain perbandingan waktu yang dapat dilihat pada tabel-tabel percobaan dan grafik di atas, dapat dilihat pula dari hasil penambangan data yang dilakukan oleh kedua algoritma yang digunakan, kecenderungan pemesanan obat di Apotek UAD yang dilakukan oleh mahasiswa KKN. Hasil tersebut dapat dilihat pada Gambar 9 dan Gambar 10 di bawah ini.



Gambar 9 : Hasil Penambangan Algoritma *Apriori*

 e-ISSN: 2338-5197



Gambar 10: Hasil Penambangan Algoritma *Hash Based*

Dari tabel-tabel percobaan, grafik, serta hasil penambangan data terlihat bahwa :

a. Waktu yang digunakan oleh algoritma *apriori* dalam melakukan penambangan data relatif lebih pendek dari waktu yang digunakan oleh algoritma *hash based*. (Waktu algoritma *apriori* masih dalam hitungan detik untuk menghasilkan keputusan, sedangkan algoritma *hash based* sudah memasuki hitungan menit untuk mengdailkan keputusan)
b. Waktu yang dibutuhkan algoritma *hash based* relatif lebih lama dikarenakan dalam melakukan penambangan data, terdapat langkah yaitu ketika mencari alamat *hash* jika terjadi *collision* pada alamat tiap *item*, maka proses dilakukan secara berulang sampai tidak terjadi *collision*.
c. Kecenderungan mahasiswa KKN UAD dalam memesan obat di Apotek UAD adalah Betadine sol, Lang minyak kayu putih, dan Betadine plester dengan support sebesar 88% dan confidence sebesar 100%.

5. PENUTUP

5.1. Kesimpulan

Karena pada algoritma *hash based* terjadi *collision*, maka waktu yang dibutuhkan dalam menambang data lebih lama dibandingkan waktu yang dibutuhkan algoritma *apriori*. Selain itu setelah langkah menempatkan *item* pada *hash tabel* sesuai dengan alamatnya, langkah yang dilakukan *hash based* sama dengan langkah yang dilakukan *apriori*. Hal ini membuat algoritma *hash based* tidak lebih efisien dibandingkan algoritma *apriori.*

5.2. Saran

Saran dalam pelaksanaan penelitian ini adalah sebagai berikut :

1. Dapat dilakukan penelitian dengan membandingan masing-masing algoritma dengan algoritma lain (misalnya algoritma *Hash Based* dan Algoritma *FP Growth*)
2. Penelitian dapat dilakukan dengan volume data yang lebih besar dan dengan data transaksi lain (misalnya data transaksi pada toserba/supermarket).

6. DAFTAR PUSTAKA

[1] Kusrini. 2009. *Algoritma Data Mining*. CV Andi Offset. Yogyakarta
[2] Yulita Marsela, S. Moertini Veronica. 2004. Analisis Keranjang Pasar Dengan Algoritma Hash-Based Pada Dara Transaksi Penjualan Apotek. Skripsi. Jurusan Ilmu Komputer. Universitas Katolik Parahyangan. Bandung.
[3] Analisis Market Basket dengan Algoritma Apriori dan Algoritma Hash Based http://eprints.unsri.ac.id/83/1/6-Erwin.pdf(diunduh pada tanggal 18 April 2012)
[4] Penggunaan Metode Fuzzy C-Covering untuk Analisa Market Basket pada Supermarket http://puslit2.petra.ac.id/gudangpaper/files/1091.pdf(diunduh pada tanggal 20 April 2012)
[5] Setyawan, Bambang. 2011. Analisa Keranjang Pasar Pada Toko Apotek Menggunakan Algoritma Hash Based. Skripsi. Jurusan Ilmu Komputer. Universitas Pembangunan Nasional Veteran Jakarta. Jakarta.

An Implementation of the FP-growth Algorithm

Christian Borgelt

Department of Knowledge Processing and Language Engineering
School of Computer Science, Otto-von-Guericke-University of Magdeburg
Universitätsplatz 2, 39106 Magdeburg, Germany

borgelt@iws.cs.uni-magdeburg.de

ABSTRACT

The FP-growth algorithm is currently one of the fastest approaches to frequent item set mining. In this paper I describe a C implementation of this algorithm, which contains two variants of the core operation of computing a projection of an FP-tree (the fundamental data structure of the FP-growth algorithm). In addition, projected FP-trees are (optionally) pruned by removing items that have become infrequent due to the projection (an approach that has been called FP-Bonsai). I report experimental results comparing this implementation of the FP-growth algorithm with three other frequent item set mining algorithms I implemented (Apriori, Eclat, and Relim).

1. INTRODUCTION

One of the currently fastest and most popular algorithms for frequent item set mining is the FP-growth algorithm [8]. It is based on a prefix tree representation of the given database of transactions (called an FP-tree), which can save considerable amounts of memory for storing the transactions. The basic idea of the FP-growth algorithm can be described as a *recursive elimination* scheme: in a preprocessing step delete all items from the transactions that are not frequent individually, i.e., do not appear in a user-specified minimum number of transactions. Then select all transactions that contain the least frequent item (least frequent among those that are frequent) and delete this item from them. Recurse to process the obtained reduced (also known as *projected*) database, remembering that the item sets found in the recursion share the deleted item as a prefix. On return, remove the processed item also from the database of all transactions and start over, i.e., process the second frequent item etc. In these processing steps the prefix tree, which is enhanced by links between the branches, is exploited to quickly find the transactions containing a given item and also to remove this item from the transactions after it has been processed.

In this paper I describe an efficient C implementation of the FP-growth algorithm. In Section 2 I briefly review how the transaction database is preprocessed in a way that is common to basically all frequent item set mining algorithms. Section 3 explains how the initial FP-tree is built from the (preprocessed) transaction database, yielding the starting point of the algorithm. The main step is described in Section 4, namely how an FP-tree is projected in order to obtain an FP-tree of the (sub-)database containing the transactions with a specific item (though with this item removed). The projection step is the most costly in the algorithm and thus it is important to find an efficient way of executing it. Section 5 considers how a projected FP-tree may be further pruned using a technique that has been called FP-Bonsai [4]. Such pruning can sometimes shrink the FP-tree considerably and thus lead to much faster projections. Finally, in Section 6 I report experiments with my implementation, comparing it with my implementations [5, 6] of the Apriori [1, 2] and Eclat [10] algorithms.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
OSDM'05, August 21, 2005, Chicago, Illinois, USA.
Copyright 2005 ACM 1-59593-210-0/05/08 ...$5.00.

2. PREPROCESSING

Similar to several other algorithms for frequent item set mining, like, for example, Apriori or Eclat, FP-growth preprocesses the transaction database as follows: in an initial scan the frequencies of the items (support of single element item sets) are determined. All infrequent items—that is, all items that appear in fewer transactions than a user-specified minimum number—are discarded from the transactions, since, obviously, they can never be part of a frequent item set.

In addition, the items in each transaction are sorted, so that they are in *descending* order w.r.t. their frequency in the database. Although the algorithm does not depend on this specific order, experiments showed that it leads to much shorter execution times than a random order. An *ascending* order leads to a particularly slow operation in my experiments, performing even worse than a random order. (In this respect FP-growth behaves in exactly the opposite way as Apriori, which in my implementation usually runs fastest if items are sorted ascendingly, but in the same way as Eclat, which also profits from items being sorted descendingly.)

This preprocessing is demonstrated in Table 1, which shows an example transaction database on the left. The frequencies of the items in this database, sorted descendingly, are shown in the middle of this table. If we are given a user specified minimal support of 3 transactions, items f and g can be discarded. After doing so and sorting the items in each transaction descendingly w.r.t. their frequencies we obtain the reduced database shown in Table 1 on the right.

a d f	d 8	d a
a c d e	b 7	d c a e
b d	c 5	d b
b c d	a 4	d b c
b c	e 3	b c
a b d	f 2	d b a
b d e	g 1	d b e
b c e g		b c e
c d f		d c
a b d		d b a

Table 1: Transaction database (left), item frequencies (middle), and reduced transaction database with items in transactions sorted descendingly w.r.t. their frequency (right).

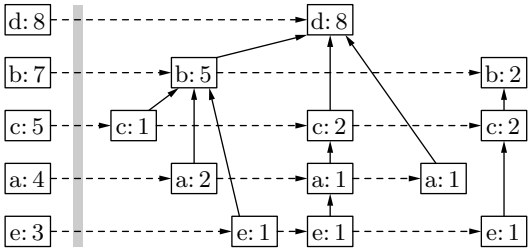


Figure 1: FP-tree for the (reduced) transaction database shown in Table 1.

3. BUILDING THE INITIAL FP-TREE

After all individually infrequent items have been deleted from the transaction database, it is turned into an FP-tree. An FP-tree is basically a prefix tree for the transactions. That is, each path represents a set of transactions that share the same prefix, each node corresponds to one item. In addition, all nodes referring to the same item are linked together in a list, so that all transactions containing a specific item can easily be found and counted by traversing this list. The list can be accessed through a head element, which also states the total number of occurrences of the item in the database. As an example, Figure 1 shows the FP-tree for the (reduced) database shown in Table 1 on the right. The head elements of the item lists are shown to the left of the vertical grey bar, the prefix tree to the right of it.

In my implementation the initial FP-tree is built from a main memory representation of the (preprocessed) transaction database as a simple list of integer arrays. This list is sorted lexicographically (thus respecting the order of the items in the transactions, which reflects their frequency). The sorted list can easily be turned into an FP-tree with a straightforward recursive procedure: at recursion depth k, the k-th item in each transaction is used to split the database into sections, one for each item. For each section a node of the FP-tree is created and labeled with the item corresponding to the section. Each section is then processed recursively, split into subsections, a new layer of nodes (one per subsection) is created etc. Note that in doing so one has to take care that transactions that are only as long as the current recursion depth are handled appropriately, that is, are removed from the section before going into recursion.

Of course, this is not the only way in which the initial FP-tree can be built. At first sight it may seem to be more natural to build it by inserting transaction after transaction into an initially empty FP-tree, creating the necessary nodes for each new transaction. Indeed, such an approach even has the advantage that the transaction database need not be loaded in a simple form (for instance, as a list of integer arrays) into main memory. Since only one transaction is processed at a time, only the FP-tree representation and one new transaction is in main memory. This usually saves space, because an FP-tree is often a much more compact representation of a transaction database.

Nevertheless I decided against such a representation for the following reasons: in order to build a prefix tree by sequentially adding transactions, one needs pointers from parent nodes to child nodes, so that one can descend in the tree according to the items present in the transaction. However, this is highly disadvantageous. As we will see later on, the further processing of an FP-tree, especially the main operation of projecting it, does not need such parent-to-child pointers in my implementation, but rather child-to-parent pointers. Since each node in an FP-tree (with the exception of the roots) has exactly one parent, this, in principle, makes it possible to work with nodes of constant size. If, however, we have to accomodate an array of child pointers per node, the nodes either have variable size or are unnecessarily large (because we have pointers that are not needed), rendering the memory management much less efficient.

It has to be conceded, though, that instead of using an array of child pointers, one may also link all children into a list. This, however, has the severe disadvantage that when inserting transactions into the FP-tree, one such list has to be searched (linearly!) for each item of the transaction in order to find the child to go to—a possibly fairly costly operation.

In contrast to this, first loading the transaction database as a simple list of integer arrays, sorting it, and building the FP-tree with a recursive function (as outlined above), makes it possible to do without parent-to-child pointers entirely. Since the FP-tree is built top down, the parent is already known when the children are created. Thus it can be passed down in the recursion, where the parent pointers of the children are set directly. As a consequence, the nodes of the FP-tree can be kept very small. In my implementation, an FP-tree node contains only fields for (1) an item identifier, (2) a counter, (3) a pointer to the parent node, (4) a pointer to the successor node (referring to the same item) and (5) an auxiliary pointer that is used when projecting the FP-tree (see below). That is, an FP-tree node needs only 20 bytes (on a 32 bit machine).

However, if we used the standard memory management, allocating a block of memory for each node, there would be an additional overhead of 4 to 12 bytes (depending on the memory system implementation) for each node for bookkeeping purposes (for instance, for storing the size of the memory block). In addition, allocating and deallocating a large number of such small memory blocks is usually not very efficient. Therefore I use a specialized memory management in my implementation, which makes it possible to efficiently handle large numbers of equally sized small mem-

ory objects. The idea is to allocate larger arrays (with several thousand elements) of these objects and to organize the elements into a "free" list (i.e., a list of available memory blocks of equal size). With such a system allocating and deallocating FP-tree nodes gets very efficient: the former retrieves (and removes) the first element of the free list, the latter adds the node to deallocate at the beginning of the free list. As experiments showed, introducing this specialized memory management led to a considerable speed-up.

4. PROJECTING AN FP-TREE

The core operation of the FP-growth algorithm is to compute an FP-tree of a projected database, that is, a database of the transactions containing a specific item, with this item removed. This projected database is processed recursively, remembering that the frequent item sets found in the recursion share the removed item as a prefix.

My implementation of the FP-growth algorithm contains two different projection methods, both of which proceed by copying certain nodes of the FP-tree that are identified by the deepest level of the FP-tree, thus producing a kind of "shadow" of it. The copied nodes are then linked and detached from the original FP-tree, yielding an FP-tree of the projected database. Afterwards the deepest level of the original FP-tree, which corresponds to the item on which the projection was based, is removed, and the next higher level is processed in the same way. The two projections methods differ mainly in the order in which they traverse and copy the nodes of the FP-tree (branchwise vs. levelwise).

The first method is illustrated in Figure 2 for the example FP-tree shown in Figure 1. The red arrows show the flow of the processing and the blue "shadow" FP-tree is the created projection. In an outer loop, the lowest level of the FP-tree, that is, the list of nodes corresponding to the projection item, is traversed. For each node of this list, the parent pointers are followed to traverse all ancestors up to the root. Each encountered ancestor is copied and linked from its original (this is what the auxiliary pointer in each node, which was mentioned above, is needed for). During the copying, the parent pointers of the copies are set, the copies are also organized into level lists, and a sum of the counter values in each node is computed in head elements for these lists (these head elements are omitted in Figure 2).

Note that the counters in the copied nodes are determined only from the counters in the nodes on the deepest level, which are propagated upwards, so that each node receives the sum of its children. Note also that due to this we cannot stop following the chain of ancestors at a node that has already been copied, even though it is clear that in this case all ancestors higher up in the FP-tree must already have been copied. The reason is that one has to update the number of transactions in the copies, adding the counter value from the current branch to all copies of the ancestors on the path to the root. This is what the second projection method tries to improve upon.

In a second traversal of the same branches, carried out in exactly the same manner, the copies are detached from their originals (the auxiliary pointers are set to null), which yields the independent projected FP-tree shown in Figure 3. This

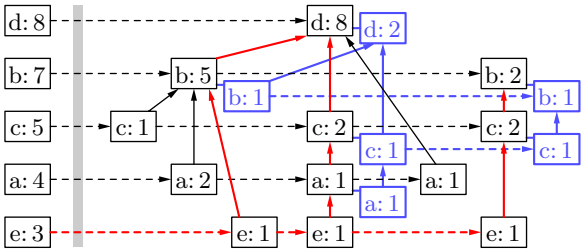


Figure 2: Computing a projection of the database w.r.t. the item e by traversing the lowest level and following all paths to the root.

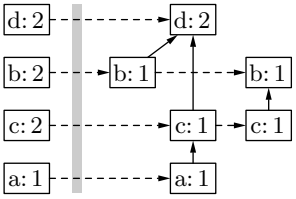


Figure 3: Resulting projected FP-tree after it has been detached from the original FP-tree.

FP-tree is then processed recursively with the prefix e. Note, however, that in this FP-tree all items are infrequent (and thus all item sets containing item e and one other item are infrequent). Hence in this example, no recursive processing would take place. This is, of course, due to the chosen example database and the support threshold.

The second projection method also traverses, in an outer loop, the deepest level of the FP-tree. However, it does not follow the chain of parent pointers up the root, but only copies the parent of each node, not its higher ancestors. In doing so, it also copies the parent pointers of the original FP-nodes, thus making it possible to find the ancestors in later steps. These later steps consist in traversing the levels of the (partially constructed) "shadow" FP-tree (not the levels of the original one!) from bottom to top. On each level the parents of the copied nodes (which are nodes in the original tree) are determined and copied, and the parent pointers of the copies are set. That is, instead of branch by branch, the FP-tree is rather constructed level by level (even though in each step nodes on several levels may be created). The advantage of this method over the one described above is that for branches that share a path close to the root, this common path has to be traversed only once with this method (as the counters for all branches are summed before they are passed to the next higher level). However, the experiments reported below show that the first method is superior in practice. As it seems, the additional effort needed for temporarily setting another parent etc. more than outweighs the advantage of the better combination of the counter values.

5. PRUNING A PROJECTED FP-TREE

After we obtained an FP-tree of a projected database, we may carry out an additional pruning step in order to simplify the tree, thus speeding up projections. I got this idea from [4], which introduces pruning techniques in a slightly different context than pure frequent item set mining (suffice it to say that there are additional constraints). One of these techniques, however, can nevertheless be used here,

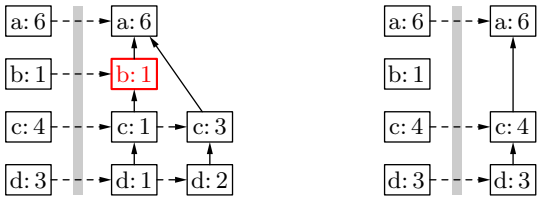


Figure 4: α-pruning of a (projected) FP-tree.

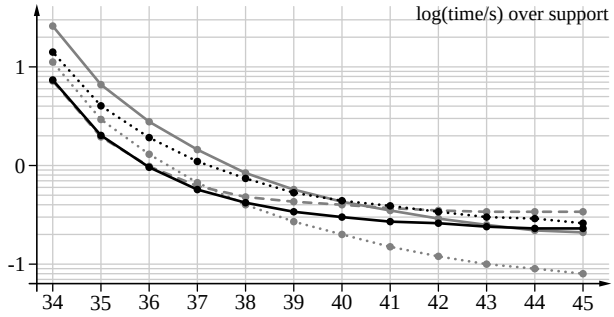


Figure 5: Results on BMS-Webview-1

namely the so-called α-pruning. The idea of this pruning is illustrated with a very simple example in Figure 4. Suppose that the FP-tree shown on the left resulted from a projection and that the minimum support is either 2 or 3. Then item b is infrequent and is not needed in projections. However, it gives rise to a branching in the tree. Hence, by removing it, the tree can be simplified and actually turned into a simple list, as it is shown on the right in Figure 4.

This pruning is achieved by traversing the levels of the FP-tree from top to bottom. The processing starts at the level following the first level that has a non-vanishing support less than the minimum support. (Items having vanishing support can be ignored, because they have no nodes in the FP-tree.) This level and the following ones are traversed and for each node the first ancestor with an item having sufficient support is determined. The parent pointer is then updated to this ancestor, bypassing the nodes corresponding to infrequent items. If by such an operation neighboring nodes receive the same parent, they are merged. They are also merged, if their parents were different originally, but have been merged in a preceding step. As an illustration consider the example Figure 4: after item b is removed, the two nodes for item c can be merged. This has to be recognized in order to merge the two nodes for item d also.

6. EXPERIMENTAL RESULTS

I ran experiments on the same five data sets that I already used in [5, 6], namely BMS-Webview-1 [9], T10I4D100K [11], census, chess, and mushroom [3]. However, I used a different machine and an updated operating system, namely a Pentium 4C 2.6GHz system with 1 GB of main memory running S.u.S.E. Linux 9.3 and gcc version 3.3.5). The results were compared to experiments with my implementations of Apriori, Eclat, and Relim. All experiments were rerun to ensure that the results are comparable.

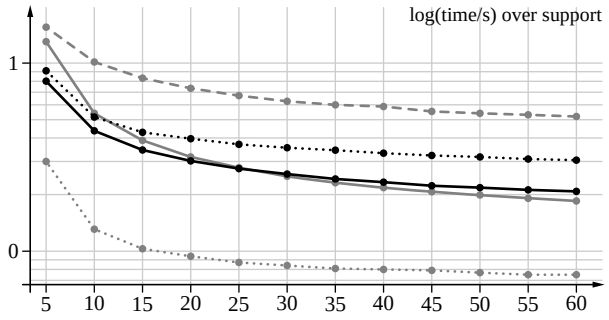


Figure 6: Results on T10I4D100K

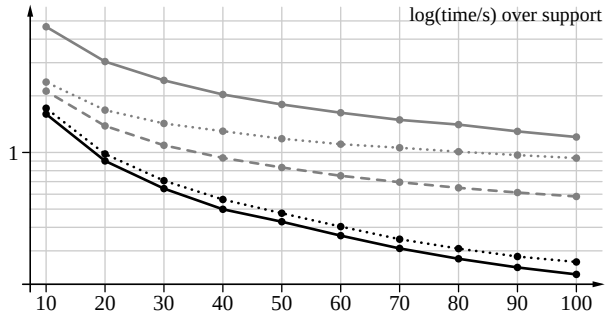


Figure 7: Results on census

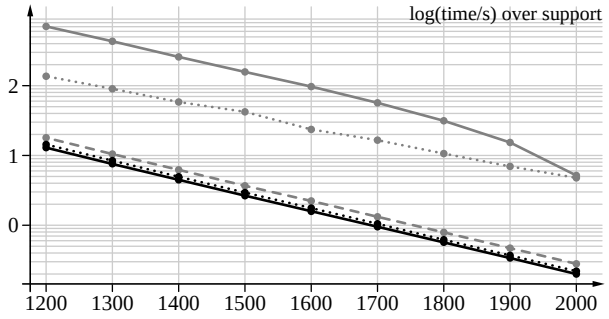


Figure 8: Results on chess

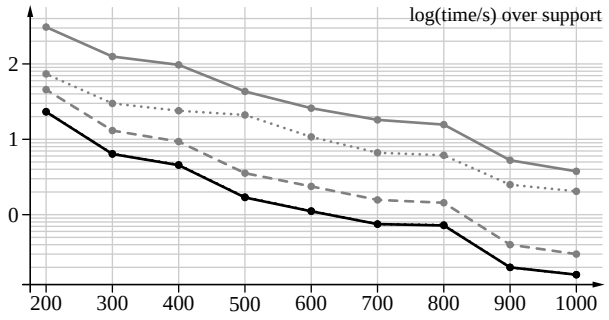


Figure 9: Results on mushroom

Figures 5 to 9 show, each for one of the five data sets, the decimal logarithm of the execution time over different (absolute) minimum support values. The solid black line refers to the implementation of the FP-growth algorithm described here, the dotted black line to the version that uses the alternative projection method. The grey lines represent the corresponding results for Apriori (solid line), Eclat (dashed line), and Relim (dotted line).[1]

Among these implementations, all of which are highly optimized, FP-growth clearly performs best. With the exception of the artificial dataset T10I4D100K, on which it is bet by a considerable margin by Relim, and for higher support values on BMS-Webview-1, where Relim also performs slightly better (presumably, because it does not need to construct a prefix tree), FP-growth is the clear winner. Only on chess, Eclat can come sufficiently close to be called competitive.

The second projection methods for FP-trees (dotted black line) generally fares worse, although there is not much difference between the two methods on chess and mushroom. This is a somewhat surprising result, because there are good reasons to believe that the second projection method may be able to yield better results than the first. I plan to examine this issue in more detail in the future.

7. CONCLUSIONS

In this paper I described an implementation of the FP-growth algorithm, which contains two methods for efficiently projecting an FP-tree—the core operation of the FP-growth algorithm. As the experimental results show, this implementation clearly outperforms Apriori and Eclat, even in highly optimized versions. However, the performance of the two projection methods, especially, why the second is sometimes much slower than the first, needs further investigation.

8. PROGRAM

The implementation of the FP-growth algorithm described in this paper (Windows™ and Linux™ executables as well as the source code, distributed under the LGPL) can be downloaded free of charge at

http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html

At this URL my implementations of Apriori, Eclat, and Relim are also available as well as a graphical user interface (written in Java) for finding association rules with Apriori.

9. REFERENCES

[1] R. Agrawal, T. Imielienski, and A. Swami. Mining Association Rules between Sets of Items in Large Databases. *Proc. Conf. on Management of Data*, 207–216. ACM Press, New York, NY, USA 1993

[2] A. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Verkamo. Fast Discovery of Association Rules. In: [7], 307–328

[3] C.L. Blake and C.J. Merz. *UCI Repository of Machine Learning Databases*. Dept. of Information and Computer Science, University of California at Irvine, CA, USA 1998
http://www.ics.uci.edu/~mlearn/MLRepository.html

[4] F. Bonchi and B. Goethals. FP-Bonsai: the Art of Growing and Pruning Small FP-trees. *Proc. 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'04, Sydney, Australia)*, 155–160. Springer-Verlag, Heidelberg, Germany 2004

[5] C. Borgelt. Efficient Implementations of Apriori and Eclat. *Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL)*. CEUR Workshop Proceedings 90, Aachen, Germany 2003.
http://www.ceur-ws.org/Vol-90/

[6] C. Borgelt. Recursion Pruning for the Apriori Algorithm. *Proc. 2nd IEEE ICDM Workshop on Frequent Item Set Mining Implementations (FIMI 2003, Brighton, United Kingdom)*. CEUR Workshop Proceedings 126, Aachen, Germany 2004.
http://www.ceur-ws.org/Vol-126/

[7] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds. *Advances in Knowledge Discovery and Data Mining*. AAAI Press / MIT Press, Cambridge, CA, USA 1996

[8] J. Han, H. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In: *Proc. Conf. on the Management of Data (SIGMOD'00, Dallas, TX)*. ACM Press, New York, NY, USA 2000

[9] R. Kohavi, C.E. Bradley, B. Frasca, L. Mason, and Z. Zheng. KDD-Cup 2000 Organizers' Report: Peeling the Onion. SIGKDD Exploration 2(2):86–93. 2000.

[10] M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New Algorithms for Fast Discovery of Association Rules. *Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD'97)*, 283–296. AAAI Press, Menlo Park, CA, USA 1997

[11] Synthetic Data Generation Code for Associations and Sequential Patterns. Intelligent Information Systems, IBM Almaden Research Center
http://www.almaden.ibm.com/software/quest/Resources/index.shtml

[1] Relim is described in a sibling paper that has also been submitted to this workshop.

ANALISIS ALGORITMA *FP-GROWTH* UNTUK REKOMENDASI PRODUK PADA DATA RETAIL PENJUALAN PRODUK KOSMETIK (STUDI KASUS : MT SHOP KELAPA GADING)

Sigit Kurniawan[1], Windu Gata[2], Hari Wiyana[3]
[1,2]*Program Studi Magister Ilmu Komputer, STMIK Nusa Mandiri Jakarta*
Jl. Kramat Raya No. 18, Jakarta 10450
[3]*Program Studi Sistem Informasi, STMIK Mercusuar Bekasi*
Jl. Raya Jatiwaringin No.144, Kota Bekasi 17411
E-mail : kurniawan.sgt@gmail.com, windu.gata@gmail.com, hary_wiyana@yahoo.com

ABSTRAKS

Semakin pesatnya pertumbuhan perusahaan kosmetik, terutama di Indonesia, membuat persaingan bisnis dibidang ini juga semakin ketat. Setiap perusahaan berusaha menghasilkan produk-produk kosmetik terbaik demi memberikan pelayanan memuaskan kepada pelanggan. Namun terkadang karena semakin besarnya data transaksi penjualan yang terjadi, membuat semakin sulit dalam menentukan strategi bisnis dan promosi. Pada kondisi tersebut dibutuhkan sebuah teknik pengolahan data yang baik, salah satunya dengan teknik data mining. Salah satu yang dapat digunakan pada teknik tersebut adalah dengan algoritma FP-Growth, yaitu sebuah algoritma yang menghasilkan frequent itemset yang nantinya akan digunakan dalam proses penentuan aturan yang dapat menghasilkan sebuah pilihan. Pada penelitian ini mengambil sebuah objek data transaksi penjualan sebuah toko kosmetik, dari data transaksi tersebut diolah dengan algoritma FP-Growth. Dari hasil pengujian yang dilakukan didapatkan sebuah aturan (rule) yang memiliki nilai confidence terbaik adalah 89% dengan aturan pada setiap pembelian produk MASKER BERAS PUTIH dapat dipastikan akan membeli PUTIH LANGSAT FACIAL FOAM.

Kata Kunci : *kosmetik, penjualan, data mining, frequent itemset, FP-Growth*

1. PENDAHULUAN

Persaingan dunia bisnis pada saat ini sangatlah ketat, terutama di bidang bisnis retail kosmetik. Beberapa tahun belakangan ini di Indonesia sudah banyak perusahaan-perusahaan kosmetik bermunculan dengan berbagai jenis merk dan variasi produk yang ditawarkan. Tidak hanya menghasilkan produk-produk kosmetik saja, namun perusahaan-perusahaan tersebut juga mendirikan bisnis-bisnis retail mereka guna mendukung penjualan produk-produk yang mereka sudah produksi.

Dalam membangun bisnis retail tersebut, perusahaan kosmetik tidak luput untuk menggunakan teknologi informasi guna mendukung kelancaran penjualan produk-produk mereka tersebut. Tidak dipungkiri bahwa penggunaan teknologi informasi menjadi sebuah keharusan pada saat ini, dengan melihat pesat dan ketatnya persaingan bisnis di bidang tersebut.

Semakin banyaknya bisnis retail penjualan produk kosmetik yang ada, maka secara otomatis persaingan bisnis tersebut akan semakin ketat. Setiap perusahaan akan berusaha untuk tidak saja menghasilkan produk-produk kosmetik yang berkualitas, namun juga berusaha untuk memberikan pelayanan pemasaran terbaik terhadap pasar atau pelanggan.

Dalam hal memberikan pelayanan terbaik terhadap pasar atau pelanggan perusahaan akan menggunakan strategi bisnis yang terbaik. Terkadang perusahaan terkendala dalam hal menentukan strategi, banyak faktor yang menyebabkan hal tersebut. Salah satu faktor penyebabnya adalah sulitnya menghasilkan analisa terkait dengan data penjualan pelanggan yang sudah ada.

Adapun pada penelitian ini mengangkat masalah yang ada di toko kosmetik MT Shop Kelapa Gading, yaitu masih belum teroganisirnya produk-produk yang paling banyak terjual dan saling berhubungan, yang diletakkan pada rak-rak produk, kemudian belum maksimalnya promosi produk-produk yang dibuat dalam satu paket yang saling berhubungan. Selain itu masih banyak produk-produk yang di produksi namun proses penjualannya tidak atau belum maksimal.

Dari kasus yang terjadi di MT Shop Kelapa Gading tersebut penelitian ini akan mencoba memberikan usulan terkait sebuah cara bagi penganalisaan data penjualan produk untuk memaksimalkan strategi pemasaran dan promosi produk dari sebuah retail kosmetik. Tidak hanya memaksimalkan strategi pada pemasaran dan promosi produk saja, namun dapat juga memberikan rekomendasi untuk bagian terkait. Seperti pada bagian produksi dalam memberikan rekomendasi jumlah produk yang akan diproduksi seusai dengan hasil analisa data penjualan pelanggan tersebut. Selain itu pula mencegah terjadinya *over stock* yang akan berdampak pada jumlah produk yang kadaluarsa menjadi berlebihan.

Salah satu cara yang dapat digunakan adalah dengan menerapkan penggunaan *data mining*. Karena di dalam *data mining* terdapat cara dan teknik dalam pemenuhan kebutuhan informasi yang luas dan informasi tersebut dapat digunakan sebagai bahan pengambilan keputusan. Penerapan *data mining* tersebut salah satunya dengan

cara mengimplementasikan salah satu fungsi *association analysist* yaitu algoritma *FP-Growth*. Adapun *FP-Growth* sendiri adalah alternatif algoritma yang dapat digunakan dalam hal menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sekumpulan data.

Berbagai penelitian yang telah menggunakan algoritma yang serupa dengan penelitian ini sudah banyak dilakukan oleh para peneliti lain, ada yang menggunakan algoritma *FP-Growth* untuk penentuan pola pada *semantic web*, ada juga penelitian yang menggunakan objek data transaksi bank dan ada pula penelitian yang menggunakan objek pada perpustakaan kampus. Penelitian ini tidak jauh berbeda dengan penelitian-penelitian yang sudah dilakukan sebelumnya, hanya saja yang berbeda berada pada objek yang digunakan. Penelitian ini berfokus pada objek data transaksi retail penjualan produk kosmetik, dengan melihat semakin meningkatnya bisnis dibidang ini, semakin meningkat pula jumlah data transaksi penjualan yang dapat dijadikan sebagai objek penelitian.

1.1. Tinjauan Pustaka

Data mining sebagai sebuah disiplin ilmu yang sebagian besar terbuka bagi dunia. Sebagian besar kita bahkan tidak pernah menyadari bahwa hal itu terjadi. Tapi setiap kali kita mendaftarkan diri sebagai anggota toko grosir, berbelanja menggunakan kartu kredit, atau menjelajahi web, sebenarnya kita sedang membuat data. Data ini disimpan dalam komputer besar dan kuat milik sebuah perusahaan yang kita setujui setiap harinya. Berlandaskan pada kumpulan data tersebut ada pola-indikator kepentingan, kebiasaan, dan perilaku kita. *Data mining* memungkinkan orang untuk menemukan dan menafsirkan pola tersebut, untuk membantu mereka membuat keputusan yang lebih tepat dan lebih baik dalam melayani pelanggan mereka. (Matthew North, 2012)

Menurut (Susanto & Suryadi, 2010) *data mining* memiliki beberapa fungsi utama yang pada setiap fungsinya memiliki tujuan penggunaan yang berbeda, yaitu:

A. *Description* (deskripsi), untuk memberi gambaran secara ringkas bagi sekumpulan data yang jumlahnya sangat besar dan banyak jenisnya.
B. *Estimation* (estimasi), untuk menerka sebuah nilai yang belum diketahui, misal menerka penghasilan seseorang ketika informasi mengenai orang tersebut diketahui.
C. *Prediction* (prediksi), untuk memperkirakan nilai masa mendatang, misal memprediksi stok barang satu tahun ke depan.
D. *Classification* (klasifikasi), merupakan proses penemuan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui.
E. *Clustering* (pengelompokan), yaitu pengelompokan mengidentifikasi data yang memiliki karakteristik tertentu.
F. *Association* (asosiasi), dinamakan juga analisis keranjang pasar dimana fungsi ini mengidentifikasi produk yang kemungkinan dibeli konsumen bersamaan dengan produk lain.

Algoritma *FP-Growth* merupakan pengembangan dari algoritma *Apriori*. Algoritma *Frequent Pattern Growth (FP-Growth)* adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. (Samuel D, 2008)

Pada algoritma *FP-Growth* menggunakan konsep pembangunan *tree*, yang biasa disebut *FP-Tree*, dalam pencarian *frequent itemset* bukan menggunakan *generate candidate* seperti yang dilakukan pada algoritma *Apriori*. Dengan menggunakan konsep tersebut, algoritma *FP-Growth* menjadi lebih cepat daripada algoritma *Apriori*. (Erwin, 2009)

Algoritma *FP-Growth* memiliki tahapan-tahapan yang harus dilewati agar dapat memberikan hasil yang maksimal, tahapan-tahapan tersebut (Jiawei Han, 2006), yaitu:

1. Tahap pembangkitan *conditional pattern base.*
2. Tahap pembangkitan *conditional FP-Tree.*
3. Tahap pencarian *frequent itemset.*

Association rules merupakan suatu proses pada *data mining* untuk menentukan semua aturan asosiatif yang memenuhi syarat minimum untuk *support* (*minsup*) dan *confidance* (*minconf*) pada sebuah *database*. Kedua syarat tersebut akan digunakan untuk *interesting association rules* yang dibandingkaan dengan batasan yang telah ditentukan, yaitu *minsup* dan *minconf*.

Dalam menentukan nilai *minimum support* sebuah *item* dapat diperoleh dengan menggunakan rumus persamaan (1) seperti di bawah ini:

$$\text{Support (A)} = \frac{\text{Jumlah Transaksi Mengandung A}}{\text{Total Transaksi}} \quad (1)$$

$$\text{Support } (A \cap B) = \frac{\text{Jumlah Transaksi Mengandung A dan B}}{\text{Total Transaksi}} \quad (1)$$

Sedangkan dalam menentukan *minimum confidence* dapat ditentukan dengan rumus persamaan (2) seperti dibawah ini :

$$\textit{Confidence } (A \rightarrow B) = P(A \mid B) = \frac{\text{Jumlah Transaksi Mengandung A dan B}}{\text{Jumlah Transaksi Mengandung A}} \quad (2)$$

Penggalian *association rules* adalah suatu prosedur untuk mencari hubungan antar *item* dalam suatu *dataset*. Dimulai dengan mencari *frequent itemset*, yaitu kombinasi yang paling sering terjadi dalam suatu *itemset* dan harus memenuhi *minimal support (minsup)*.

Penggalian *itemset* yang sering muncul dari database transaksional yang besar adalah salah satu masalah yang paling menantang dalam *data mining*. Dalam banyak skenario dunia nyata, data tidak diekstraksi dari sumber data tunggal tetapi dari data yang terdistribusi dan heterogen. Pengetahuan yang ditemukan diharapkan bisa membantu operasional bisnis yang lebih baik. Dalam metode *data mining, association rules* adalah salah satu yang paling populer. Namun, penggalian pola informasi yang menggunakan *association rules* seringkali menghasilkan pola individu yang sangat besar, sehingga membiarkan analis menyelesaikan tugas dengan semua peraturan dan menemukan salah satu yang menarik. Dalam penelitian ini, disajikan pembangkitan *association rules* dengan menggunakan algoritma *Apriori* dan *FP-Growth*. (J.Suresh, P.Rushyanth & Ch.Trinath, 2013)

Data mining menemukan pola yang sering terjadi dari database yang besar menjadi tugas yang hebat dan banyak penelitian terus berlanjut. Sebuah studi komparatif dibuat antara algoritma *mining* pola klasik yang sering menggunakan pembentukan dan uji coba sekumpulan kandidat (algoritma *Apriori*) dan algoritma tanpa pembentukan sekumpulan kandidat (algoritma *FP-Growth*). Algoritma *Apriori* menemukan *itemset* yang sering terjadi, maka semua bagian himpunannya juga harus sering muncul. Algoritma *Apriori* menghasilkan *itemset* dan uji coba kandidat jika sering muncul. Teknik *FP-Growth* menggunakan bagian pola yang berkembang untuk menggali pola yang sering terjadi dari *database* yang besar. Diawali dengan memperluas struktur pohon yang digunakan untuk menyimpan informasi penting dan memadatkan pola yang sering muncul. *FP-Growth* menemukan *itemset* yang sering muncul tanpa pembentukan sekumpulan kandidat *itemset*. (M.Kavitha & S.T.Tamil Selvi, 2016)

Algoritma *FP-Growth* mengacu pada penemuan dan analisis pola secara otomatis dalam alur klik pada halaman website dengan keterkaitan data yang dikumpulkan atau dihasilkan sebagai hasil dari interaksi pengguna dengan sumber daya web pada satu atau lebih situs web. Terdiri dari tiga tahap yaitu data *preprocessing*, penemuan pola dan analisis pola. Pada tahap penemuan pola, algoritma pencarian pola sering diterapkan pada data mentah. Pada tahap analisis pola, pengetahuan menarik diekstrak dari pola yang sering muncul dan hasil ini digunakan untuk modifikasi situs web. Algoritma *FP-Growth* digunakan untuk mendapatkan pola akses yang sering muncul dari pencatatan data web dan memberikan informasi berharga tentang minat pengguna. (Bhavesh V. Berani, 2014)

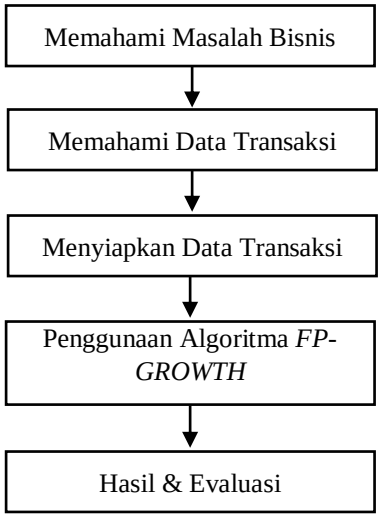
Karena ada peningkatan teknologi informasi yang signifikan, hasil perbaikan memberikan sejumlah data besar yang tersimpan di dalam *database* dan gudang data. *Data mining* digunakan untuk menggali sebagian besar data menggunakan pandangan atau perspektif yang berbeda dan kemudian meringkas data menjadi lebih bernilai. Informasi yang digali ini dapat dimanfaatkan untuk tujuan bisnis yang berbeda guna meningkatkan pembangunan ekonomi. Kini, dalam *data mining*, algoritma *FP-Growth* saat ini merupakan salah satu aturan asosiasi tercepat untuk menggali *itemset* yang sering muncul. Seperti di sektor perbankan, banyak database transaksi mengandung rangkaian transaksi yang sama berkali-kali, dalam penelitian ini menggambarkan penerapan algoritma pohon *FP-Growth* dalam database transaksi bank sehingga menjamin kinerja terbaik. Tujuannya adalah untuk mempertimbangkan dan menguji teknik asosiasi yang ada untuk menggali *itemset* yang sering muncul dan implementasi yang sukses dari algoritma pohon *FP-Growth* dalam database transaksi perbankan untuk meningkatkan kinerja penggalian *itemset* yang sering muncul. (Aroosa Hameed, 2015)

Jumlah penduduk Indonesia diproyeksi akan menembus angka 271 jutaan pada tahun 2020. Jumlah tersebut meningkat sekitar 4,8% dibandingkan dengan total populasi penduduk Indonesia tahun 2016. Dengan angka pertumbuhan yang cukup besar tersebut, Indonesia termasuk pasar yang sangat potensial bagi perusahaan kosmetik. Kementerian Perindustrian bahkan menyebutkan nilai industri ini ditaksir bisa mencapai 100 triliun rupiah. *Euromonitor International* juga memprediksi bahwa Indonesia dan Vietnam akan menjadi pasar kosmetik yang paling cepat bertumbuh di kawasan Asia. (Marsin Indonesia, 2017)

Kemajuan pada industri kecantikan di Indonesia saat ini menunjukkan peningkatan. Berdasarkan data Kementerian Perindustrian (2016), pertumbuhan pasar industri ini rata-rata mencapai 9,67% per tahun dalam

enam tahun terakhir (2009-2015). Diperkirakan besar pasar (*market size*) kosmetik sebesar 46,4 triliun rupiah di tahun 2017 ini. Dengan jumlah tersebut, Indonesia merupakan *potential market* bagi para pengusaha industri kecantikan baik dari luar maupun dalam negeri. (Sigma Research, 2017)

Metodologi Penelitian



Gambar 1. Metode Penelitian CRISP-DM yang telah disesuaikan

Dalam melakukan penelitian ini agar didapatkan hasil yang sesuai seperti yang diinginkan maka, digunakan metode penelitian yang telah ditetapkan sesuai dengan algoritma pada penelitian ini. Adapun metode penelitian yang digunakan adalah CRISP-DM. Berikut adalah langkah-langkah yang digunakan pada objek penelitian ini menggunakan CRISP-DM yang telah disesuaikan, seperti pada gambar 1 diatas. Adapun penjelasan dari masing-masing tahapan diatas adalah sebagai berikut:

a. Memahami Masalah Bisnis
   Masalah yang diidentifikasi pada penelitian ini adalah untuk mendapatkan rekomendasi terkait produk-produk kosmetik yang dipasarkan di MT Shop Kelapa Gading guna menghasilkan penjualan produk yang maksimal juga memaksimalkan promosi produk dengan penjualan terbaik.
b. Memahami Data Transaksi
   Data transaksi yang diambil pada penelitian ini harus terlebih dahulu dipahami isinya, pada penelitian ini data transaksi yang diambil adalah data penjualan terhadap seluruh produk yang ada di MT Shop Kelapa Gading, baik transaksi penjualan promo maupun regular.
c. Menyiapkan Data Transaksi
   Untuk dapat memberikan rekomendasi terhadap manajemen terkait dengan produk-produk yang dipasarkan maka diperlukan data transaksi yang sudah dipahami isinya untuk diolah guna menghasilkan *output* yang dapat digunakan sebagai rekomendasi serta penentuan kebijakan perusahaan. Pada penelitian ini data yang diambil adalah data transaksi selama satu tahun mulai dari Januari 2017 sampai dengan Desember 2017, sebanyak 939 transaksi.
d. Penggunaan Algoritma *FP-Growth*
   Peneletian ini menggunakan salah satu metode algoritma *data mining* untuk mengolah data guna mendapatkan hasil yang digunakan sebagai rekomendasi bagi perusahaan. Algoritma yang digunakan adalah algoritma klasifikasi *FP-Growth*. Algoritma ini dipilih karena selain dapat menentukan kemunculan produk yang sering terjual, juga dapat memberikan rekomendasi bagi produk-produk yang saling terkait yang mungkin akan dibeli oleh pelanggan, yang dapat menjadi referensi bagi pengaturan tata letak produk di toko.
e. Hasil & Evaluasi
   Dari seluruh tahapan kerangka kerja penelitian ini maka, diharapkan dapat memberikan hasil serta evaluasi bagi perusahaan untuk meningkatkan penjualan serta promosi produk-produk kosmetik di kemudian hari. Dan diharapkan penelitian ini juga dapat menjadi referensi bagi peneliti lain.

2. PEMBAHASAN

Sesuai dengan objek masalah dan usulan yang telah dijelaskan sebelumnya, penelitian ini akan menggunakan sebuah *software tools* yang digunakan untuk menghasilkan data analisa dari algoritma *FP-Growth*. *Software tools* tersebut adalah Rapidminer versi 8.0, untuk dapat menggunakan *software tools* tersebut haruslah disiapkan

data transaksi penjualan yang sudah melalui tahap *data cleaning, data integration* dan *data selection* kedalam format file excel atau csv.

Penelitian ini menggunakan data dari transaksi penjualan kosmetik di MT Shop Kelapa Gading pada rentang waktu periode tahun 2017 yaitu, dimulai dari satu Januari 2017 sampai dengan 31 Desember 2017, dengan jumlah data transaksi sebanyak 939 transaksi penjualan dengan hanya mengambil 10 produk terjual paling tertinggi, seperti pada tabel dibawah ini.

Tabel 1. 10 produk terjual paling tertinggi

No	*Produk*	*Jml Terjual*
1	intensive acne care	1,981
2	masker beras putih	929
3	facial foam acne	895
4	putih langsat facial foam	614
5	minyak zaitun	592
6	masker ketan hitam	570
7	putih langsat peel off mask	549
8	wnc gran.cleansing	516
9	pembersih kenanga	503
10	lotion jerawat	501

Seperti yang sudah disebutkan diatas, untuk dapat menghasilkan analisa dari algoritma *FP-Growth* menggunakan s*oftware tools* Rapidminer 8.0, maka terlebih dahulu harus disiapkan *dataset* transaksi penjualan yang sudah melewati tahapan-tahapan *data cleaning, data integration* dan *data selection* yang diambil dari *database* penjualan MT Shop Kelapa Gading yang menjadi objek penelitian.

Tabel dibawah ini menunjukan data *sample dataset* transaksi penjualan dari 10 produk teratas yang akan dijadikan atribut pada analisa menggunakan algoritma *FP-Growth*.

Tabel 2. Data *sample dataset* transaksi penjualan

Tanggal Transaksi	*NoTransaksi*	*Produk*
01/01/2017	SLS-0101170035	putih langsat facial foam
01/01/2017	SLS-0101170035	facial foam acne
01/01/2017	SLS-0101170052	intensive acne care
01/01/2017	SLS-0101170052	lotion jerawat
01/01/2017	SLS-0101170082	intensive acne care
04/01/2017	SLS-0101170306	wnc gran.cleansing
13/01/2017	SLS-0101170837	wnc gran.cleansing
02/01/2017	SLS-0101170126	putih langsat peel off mask
02/01/2017	SLS-0101170126	putih langsat facial foam
02/01/2017	SLS-0101170132	facial foam acne
04/03/2017	SLS-0103170302	wnc gran.cleansing
04/03/2017	SLS-0103170304	intensive acne care
04/03/2017	SLS-0103170304	facial foam acne
04/03/2017	SLS-0103170306	putih langsat facial foam
09/12/2017	SLS-0112170303	facial foam acne
09/12/2017	SLS-0112170314	intensive acne care
09/12/2017	SLS-0112170314	masker ketan hitam
09/12/2017	SLS-0112170334	putih langsat facial foam
09/12/2017	SLS-0112170334	masker ketan hitam
10/12/2017	SLS-0112170351	facial foam acne
10/12/2017	SLS-0112170351	intensive acne care
10/12/2017	SLS-0112170353	masker ketan hitam
10/12/2017	SLS-0112170364	facial foam acne
10/12/2017	SLS-0112170364	lotion jerawat
10/12/2017	SLS-0112170364	masker beras putih
11/12/2017	SLS-0112170430	masker ketan hitam
11/12/2017	SLS-0112170430	masker beras putih
12/12/2017	SLS-0112170460	masker beras putih
12/12/2017	SLS-0112170460	putih langsat facial foam
12/12/2017	SLS-0112170460	facial foam acne
12/12/2017	SLS-0112170460	intensive acne care
13/12/2017	SLS-0112170503	facial foam acne
13/12/2017	SLS-0112170503	intensive acne care

Setelah tahapan membuat *dataset*, maka selanjutnya adalah membuat *itemset* dari *dataset* tersebut. Pada penelitian ini menggunakan 10 produk terjual tertinggi sebagai atribut. Untuk membuat *itemset* adalah dengan mencari jumlah atribut yang ada di masing-masing transaksi yang telah disajikan dalam bentuk *dataset*

sebelumnya. Tabel dibawah ini adalah bentuk format *itemset* yang akan digunakan di dalam proses pengolahan data menggunakan Rapidminer.

Tabel 3. Data *sample itemset* 10 produk terjual tertinggi tahun 2017

putih langsat facial foam	*minyak zaitun*	*masker ketan hitam*	*putih langsat peel off mask*	*wnc gran.cleansing*	*pembersih kenanga*	*lotion jerawat*
1	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	1	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
1	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	0	1	0	0	0
0	1	1	0	0	0	0
0	0	1	0	0	1	0

Dari data *itemset* pada tabel diatas menjelaskan *item* produk yang dijadikan sebagai atribut dalam analisa ini, setiap produk yang terjual pada setiap transaksi akan di tandai dengan angka 1 atau bernilai *TRUE*, sedangkan untuk produk yang tidak terjual akan di tandai dengan angka 0 atau bernilai *FALSE*.

Pada data *itemset* yang sudah disediakan akan diunggah kedalam Rapidminer sebagai langkah awal, selanjutnya memilih algoritma yang akan digunakan yaitu, *FP-Growth*. Setelah itu akan dapat diketahui frekuensi atribut yang kemunculan transaksinya sering terjadi. Cara menghitung frekuensi atribut adalah dengan menjumlahkan masing-masing atribut dari seluruh transaksi yang ada. Jika dilakukan secara manual menggunakan Microsoft Excel dan Rapidminer maka data yang dihasilkan akan muncul seperti tabel di bawah ini.

Tabel 4. Jumlah frekuensi atribut menggunakan Microsoft Excel

No	*Produk*	*Frequent*
1	intensive acne care	476
2	facial foam acne	314
3	masker beras putih	247
4	lotion jerawat	219
5	masker ketan hitam	211
6	putih langsat facial foam	209
7	putih langsat peel off mask	207
8	minyak zaitun	105
9	pembersih kenanga	98
10	wnc gran.cleansing	72

Tabel 5. Hasil jumlah frekuensi atribut menggunakan Rapidminer.

No	*Name*	*Type*	*Statistics*		
			Least	*Most*	*Values*
1	intensive acne care	Binominal	0 (462)	1 (476)	1 (476), 0 (462)
2	masker beras putih	Binominal	1 (247)	0 (691)	0 (691), 1 (247)
3	facial foam acne	Binominal	1 (314)	0 (624)	0 (624), 1 (314)
4	putih langsat facial foam	Binominal	1 (209)	0 (729)	0 (729), 1 (209)
5	minyak zaitun	Binominal	1 (105)	0 (833)	0 (833), 1 (105)
6	masker ketan hitam	Binominal	1 (211)	0 (727)	0 (727), 1 (211)
7	putih langsat peel of mask	Binominal	1 (207)	0 (731)	0 (731), 1 (207)
8	wnc gran.cleansing	Binominal	1 (72)	0 (866)	0 (866), 1 (72)
9	pembersih kenanga	Binominal	1 (98)	0 (840)	0 (840), 1 (98)
10	lotion jerawat	Binominal	1 (219)	0 (719)	0 (719), 1 (219)

Dari kedua tabel diatas dapat dilihat bahwa frekuensi atribut terbesar ada pada produk "INTENSIVE ACNE CARE" dengan jumlah frekuensi sebesar 476 terjual.

Tahap berikutnya adalah menampilkan proses *support* dari *itemset* yang sudah diunggah kedalam Rapidminer. Dari hasil pengolahan *itemset* tersebut didapatkan hasil nilai *support* dari *itemset*, penelitian ini memberikan batasan nilai *support* dengan *minimum support* sebesar 0.95 atau 95%. Tabel dibawah ini adalah hasil dari tahapan tersebut.

Tabel 6. *Minimum support* 95% dari *frequent itemset*

Size	*Support*	*Item 1*	*Item 2*	*Item 3*
1	0.777	putih langsat facial foam		
1	0.665	facial foam acne		
1	0.507	intensive acne care		
1	0.263	masker beras putih		
1	0.233	lotion jerawat		
1	0.225	masker ketan hitam		
1	0.221	putih langsat peel off mask		
2	0.487	putih langsat facial foam	facial foam acne	
2	0.44	putih langsat facial foam	intensive acne care	
2	0.237	putih langsat facial foam	masker beras putih	
2	0.209	putih langsat facial foam	lotion jerawat	
2	0.197	putih langsat facial foam	masker ketan hitam	
2	0.3	facial foam acne	intensive acne care	
2	0.232	facial foam acne	masker beras putih	
2	0.197	facial foam acne	masker ketan hitam	
2	0.187	facial foam acne	putih langsat peel off mask	
3	0.249	putih langsat facial foam	facial foam acne	intensive acne care
3	0.212	putih langsat facial foam	facial foam acne	masker beras putih

Tahap selanjutnya adalah menentukan *association rules* dari *itemset* dan frekuensi yang sudah didapat. Proses menentukan *association rules* pada penelitian ini masih menggunakan *software tools* Rapidminer 8.0, dengan memilih *Operators → modeling Associations → Create Assosciation Rule* pada menu.

Pada tahapan tersebut dihasilkan aturan (*rule)* sesuai dengan nilai *confidence* yang telah ditentukan pada penelitian ini, nilai *confidence* mulai dari 0,5 (50%), 0,6 (60%), 0,7 (70%), 0,8 (80%) dan 0,9 (90%). *Confidence* yang dimaksud pada penelitian ini dengan menggunakan algoritma *FP-Growth* adalah nilai kecenderungan suatu *item* dengan *item* tertentu akan diambil oleh pelanggan secara bersamaan. Dari nilai *confidence* tersebut diharapkan dapat menghasilkan *rule* yang terbaik. Berikut ini adalah hasil *rule* dari masing-masing nilai *confidence* diatas, yaitu:

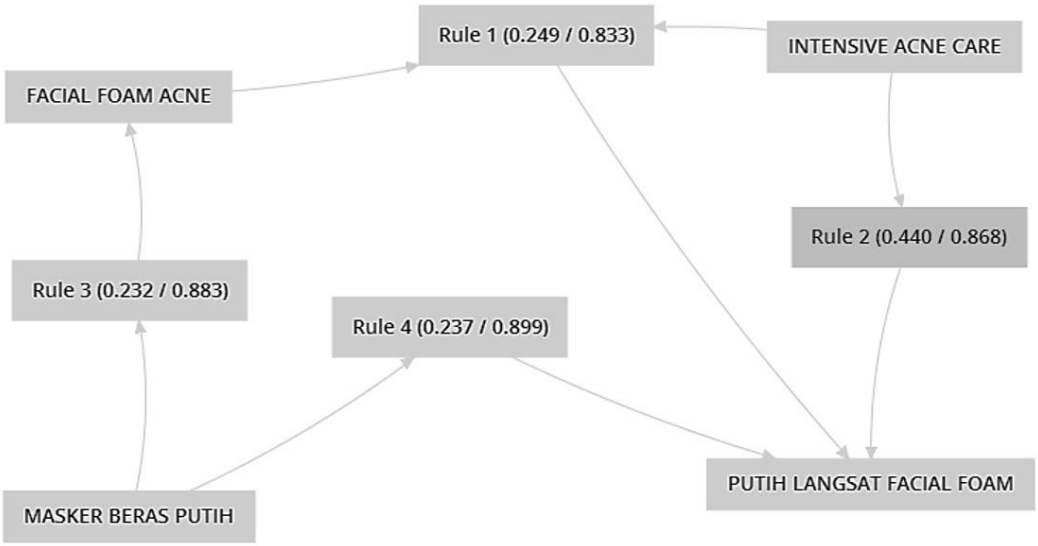
a. *Rule* dengan nilai *confidence* 0,5 (50%).
   [PUTIH LANGSAT FACIAL FOAM, FACIAL FOAM ACNE] --> [INTENSIVE ACNE CARE] (*confidence*: 0.512)
   [PUTIH LANGSAT FACIAL FOAM] --> [INTENSIVE ACNE CARE] (*confidence*: 0.567)
   [PUTIH LANGSAT FACIAL FOAM, INTENSIVE ACNE CARE] --> [FACIAL FOAM ACNE] (*confidence*: 0.567)
   [INTENSIVE ACNE CARE] --> [FACIAL FOAM ACNE] (*confidence*: 0.590)
   [PUTIH LANGSAT FACIAL FOAM] --> [FACIAL FOAM ACNE] (*confidence*: 0.627)
   [FACIAL FOAM ACNE] --> [PUTIH LANGSAT FACIAL FOAM] (*confidence*: 0.732)
   [FACIAL FOAM ACNE, INTENSIVE ACNE CARE] --> [PUTIH LANGSAT FACIAL FOAM] (*confidence*: 0.833)
   [INTENSIVE ACNE CARE] --> [PUTIH LANGSAT FACIAL FOAM] (*confidence*: 0.868)
   [MASKER BERAS PUTIH] --> [FACIAL FOAM ACNE] (*confidence*: 0.883)
   [MASKER BERAS PUTIH] --> [PUTIH LANGSAT FACIAL FOAM] (*confidence*: 0.899)

b. *Rule* dengan nilai *confidence* 0,60 (60%).
   [PUTIH LANGSAT FACIAL FOAM] --> [FACIAL FOAM ACNE] (*confidence*: 0.627)
   [FACIAL FOAM ACNE] --> [PUTIH LANGSAT FACIAL FOAM] (*confidence*: 0.732)
   [FACIAL FOAM ACNE, INTENSIVE ACNE CARE] --> [PUTIH LANGSAT FACIAL FOAM] (*confidence*: 0.833)
   [INTENSIVE ACNE CARE] --> [PUTIH LANGSAT FACIAL FOAM] (*confidence*: 0.868)
   [MASKER BERAS PUTIH] --> [FACIAL FOAM ACNE] (*confidence*: 0.883)
   [MASKER BERAS PUTIH] --> [PUTIH LANGSAT FACIAL FOAM] (*confidence*: 0.899)
c. *Rule* dengan nilai *confidence* 0,70 (70%).
   [FACIAL FOAM ACNE] --> [PUTIH LANGSAT FACIAL FOAM] (*confidence*: 0.732)

[FACIAL FOAM ACNE, INTENSIVE ACNE CARE] --> [PUTIH LANGSAT FACIAL FOAM] (*confidence*: 0.833)
[INTENSIVE ACNE CARE] --> [PUTIH LANGSAT FACIAL FOAM] (*confidence*: 0.868)
[MASKER BERAS PUTIH] --> [FACIAL FOAM ACNE] (*confidence*: 0.883)
[MASKER BERAS PUTIH] --> [PUTIH LANGSAT FACIAL FOAM] (*confidence*: 0.899)

d. *Rule* dengan nilai *confidence* 0,80 (80%).
[FACIAL FOAM ACNE, INTENSIVE ACNE CARE] --> [PUTIH LANGSAT FACIAL FOAM] (*confidence*: 0.833)
[INTENSIVE ACNE CARE] --> [PUTIH LANGSAT FACIAL FOAM] (*confidence*: 0.868)
[MASKER BERAS PUTIH] --> [FACIAL FOAM ACNE] (*confidence*: 0.883)
[MASKER BERAS PUTIH] --> [PUTIH LANGSAT FACIAL FOAM] (*confidence*: 0.899)

e. *Rule* dengan nilai *confidence* 0,90 (90%).
Tidak dihasilkan *rule* pada nilai *confidence* ini.



Gambar 2. *Graph Rule* dengan nilai *confidence* 80%

Hasil pengolahan data tersebut dapat terlihat bahwa semakin rendah penentuan nilai *confidence,* maka akan semakin banyak *rule* yang terbentuk sesuai dengan nilai *confidence* yang dimasukan. Kemudian dari data *itemset* yang diolah tersebut ternyata nilai *confidence* 89% yang paling tertinggi yang dapat dihasilkan aturannya (*rule*), seperti yang ditunjukan pada Gambar 2, sedangkan untuk nilai *confidence* 90% tidak ditemukan hasil aturannya (*rule*).

3. KESIMPULAN

Dari penelitian yang telah dilakukan pada objek MT Shop Kelapa Gading, pengolahan data transaksi penjualan selama satu tahun pada toko tersebut yang dibentuk dalam *itemset* dan diolah menggunakan Rapidminer 8.0 didapatkan nilai *confidence* tertinggi adalah 0,899 atau sekitar 89% dengan aturan (*rule*) pada setiap pembelian produk MASKER BERAS PUTIH dapat dipastikan akan membeli PUTIH LANGSAT FACIAL FOAM.

Selain itu dari kesimpulan ini diharapkan dapat menjawab permasalahan yang ada di MT Shop Kelapa Gading, seperti menentukan posisi produk-produk yang ada di rak-rak penjualan dan dapat memberikan rekomendasi bagi produk-produk yang akan dibuatkan promosi berdasarkan dari hasil pengolahan data tersebut dengan hasil *confidence* mulai dari 50% sampai dengan 89%.

Selain kesimpulan yang telah disampaikan diatas, pada penelitian ini juga berusaha memberikan saran bagi penelitian berikutnya, yaitu:

1. Pada penelitian selanjutnya dapat dilakukan dengan menggunakan data transaksi yang lebih besar lagi dan lebih beragam produk yang terjual pada setiap transaksinya.
2. Supaya dikemudian hari dapat dilakukan penelitian dengan menggunakan metode dan algoritma yang berbeda dari penelitian yang telah dilakukan saat ini. Dapat juga menggunakan algoritma *Generalized Sequential Pattern (GSP)* atau dengan algoritma prediksi *Neural Network, Decision Tree,* dan *k–Nearest Neighbor.*

3. Agar hasil penelitian ini dapat di implementasikan maka diperlukan konsultasi dan pembahasan yang mendalam dengan pihak manajemen, untuk menentukan apakah penelitian ini dapat dijadikan sebuah referensi bagi manajemen dalam menentukan strategi bisnis dan promosi dikemudian hari.

PUSTAKA

Berani, Bhavesh V, Thaker, Chirag . 2014. *FP Growth Algorithm for finding patterns in Semantic Web*. *IJEDR, Vol. 2 Issue 4*.

Erwin. 2009. *Analisis Market Basket dengan Algoritma Apriori dan FP-Growth*. *Jurnal Generic, 26-30*

Hameed, Aroosa. 2015. *FP Growth Tree Implementation In Bank Transactional Databases*. *ITEE Journal, Vol.5, Issue 3*

Han, Jiawei., Kamber, Micheline. 2006. *Data mining: Concepts and Techniques Second Edition*. San Fransisc : Morgan Kaufmann Publishers

Kavitha, M., Selvi, Tamil. 2016. *Comparative Study on Apriori Algorithm and Fp Growth Algorithm with Prod and Cons. IJCST, Vol.4, Issue 4*

Mars, Indonesia. 2017. *Studi Pemasaran Produk Kosmetik 2017,* (Online), (http://www.marsindonesia.com/products/business-reports/studi-pemasaran-produk-kosmetik-2017, diakses 03 Januari 2018)

North, Matthew. 2012. *Data mining for the Masses*. A Global Text Project Book

Samuel, David. 2008. *Penerapan Struktur FP-Tree dan Algoritma FP-Growth dalam Optimasi Penentuan Frequent Itemset*. Institut Teknologi Bandung

Sigma, Research. 2017. *Tren dan Perilaku Pasar Kosmetik Indonesia Tahun 2017.* (Online), (http://sigmaresearch.co.id/tren-dan-perilaku-pasar-kosmetik-indonesia-tahun-2017/ diakses 03 Januari 2018)

Suresh, J., Rushyanth, P., Trinath, Ch. 2013. *Generating associations rule mining using Apriori and FPGrowth Algorithms. IJCTT, Vol.4, Issue 4*

Susanto, Sani, Suryadi, Dedy. 2010. *Pengantar Data mining : Menggali Pengetahuan dan Bongkahan Data.* Yogyakarta : Andi Publisher

PERBANDINGAN ALGORITMA APRIORI DAN ALGORITMA FP-GROWTH UNTUK PEREKOMENDASI PADA TRANSAKSI PEMINJAMAN BUKU DI PERPUSTAKAAN UNIVERSITAS DIAN NUSWANTORO

Rizky Mei Anggraeni
Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro Semarang
Jl. Nakula 1 No. 5-11, Semarang, 50131, Telp: (024) 3517261, Fax : (024) 325 0165
E-mail : dira.anggraeni25@gmail.com

Abstrak

Perpustakaan merupakan salah satu tempat penyedia layanan kepustakaan yang dapat di manfaatkan untuk kegiatan proses belajar mengajar. Koleksi buku yang terdapat pada perpustakaan cukup banyak, terkadang hal tersebut membingungkan pengunjung perpustakaan yang ingin meminjam buku. Untuk memudahkan proses peminjaman buku pada perpustakaan tersebut perlu adanya sistem yaitu sistem perekomendasi untuk mahasiswa atau dosen. Pada penelitian ini dibuat sebuah sistem di mana nantinya dapat dijadikan sistem perekomendasi yaitu menggunakan dua metode data mining association rule mining dengan membandingkan algoritma apriori dan fp-growth (frequent pattern growth). Hal tersebut bertujuan untuk menentukan buku-buku yang sering dipinjam bersamaan pada transaksi peminjaman buku sebelumnya. Hasil dari sistem tersebut yaitu sebuah informasi rekomendasi buku dengan membandingkan kinerja algoritma apriori dan fp-growth.

***Kata kunci** : Data Mining, Perpustakaan, Apriori, FP-Growth*

Abstract

Right now this method of of data mining has been widely used in many fields for data processing, one of which uses of data mining is the right decision makers are educational institutions that exist in the library. The library is one of the library service providers that can be used for teaching and learning activities. Collection of books contained in the library quite a lot, sometimes confusing visitors who want to borrow a library book. To simplify the process of borrowing library books on the need for a system that can help facilitate the loan process for students or faculty. In this study perekomendasi created a system using association rule mining algorithm Apriori of data mining and Fp-Growth (Frequent Pattern Growth). It aims to look at the validity of the results obtained information books.

***kKeyword :** Data Mining, Library, Apriori, Frequent Pattern Growth*

I. PENDAHULUAN

Perpustakaan Universitas Dian Nuswantoro merupakan salah satu tempat penyedia layanan kepustakaan yang dapat dimanfaatkan untuk kegiatan proses belajar mengajar. Perpustakaan ini menyediakan berbagai macam bahan-bahan pustaka seperti buku-buku teks, literatur,refrensi, jurnal, majalah ilmiah, majalah umum, koran maupun koleksi laporan tugas akhir mahasiswa dari berbagai fakultas. Dari sekian banyak koleksi perpustakaan dapat dipinjam atau hanya boleh dibaca di tempat[1].

Untuk memudahkan proses peminjaman buku perpustakaan terutama dalam hal pencarian buku yang akan di pinjam perlu adanya sistem yang tepat dalam merekomendasikan buku apa saja yang tersedia pada perpustakaan dan yang paling sering dipinjam bersamaan oleh pengunjung perpustakaan pada transakski sebelumnya. Dikarenakan perpustakaan Universitas Dian Nuswantoro belum mempunyai sistem yang membantu merekomendasi peminjam buku perspustakaan dalam hal memilih buku yang tepat untuk digunakan, oleh karena itu perlu adanya sistem perekomendasi untuk merekomendasikan buku dengan menemukan himpunan data yang paling sering muncul dalam suatu kumpulan data. Metode yang digunakan dalam perekomendasian tersebut adalah Data Mining. Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu[3]. Perekomendasian ini dilakukan dengan membandingkan kinerja algoritma Apriori dan FP-Growth.

Algoritma Apriori ini dikembangkan pada tahun 1994 oleh Agrawal dan Srikan. Apriori adalah algoritma dalam melakukan pencarian frequent itemset dengan menggunakan pengetahuan tentang frequent itemset yang telah diketahui sebelumnya untuk memproses informasi selanjutnya. Biasanya kandidat-kandidat yang mungkin muncul ditentukan terlebih dahulu dengan memperhatikan minimum support[5].

FP-Growth adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (frequent item set) dalam sebuah kumpulan data. FP-Growth menggunakan pendekatan yang berbeda dari paradigma yang digunakan pada apriori.

II. METODE YANG DIUSULKAN

A. Data Mining

Menurut Turban dalam bukunya yang berjudul *"Decision Support System and Intelligent System", Data Mining* adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam basis data. Data Mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan machine learning untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai basis data besar[4].

B. Apriori

Algoritma Apriori digunakan untuk mencari frequent itemset yang

memenuhi minsup kemudian mendapatkan rule yang memenuhi minconf dari frequent itemset tadi[8].

Algoritma Apriori diperkenalkan oleh agrawal dan Srikant pada tahun 1994. Sampai saat ini algoritma tersebut merupakan algoritma asosiasi yang telah banyak digunakan dan dikembangkan oleh para peneliti. Algoritma Apriori dalam menangani masalah asosiasi adalah dengan mengurangi jumlah itemset yang dipertimbangkan. Pengguna menetapkan minimum support. Jadi nilai $(A \cup C) \leq$ minimum support untuk setiap assosiation rule $A \rightarrow C$ atau $(A \rightarrow C) \leq$ min- support. Apriori pertama menghasilkan semua himpunan item yang memenuhi min-dukungan. Himpunan item tersebut sering disebut himpunan item (frequent item sets)[6].

C. FP-Growth

Algoritma FP-Growth merupakan pengembangan dari algoritma Apriori. Sehingga kekurangan dari algoritma Apriori diperbaiki oleh algoritma FP-Growth[4].

Frequent Pattern Growth (FP-Growth) adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul *(frequent itemset)* dalam sebuah kumpulan data[8].

Pada algoritma Apriori diperlukan generate candidate untuk mendapatkan frequent itemsets. Akan tetapi, di algoritma FP-Growth generate candidate tidak dilakukan karena FP-Growth menggunakan konsep pembangunan tree dalam pencarian frequent itemsets. Hal tersebutlah yang menyebabkan algoritma FP-Growth lebih cepat dari algoritma Apriori.

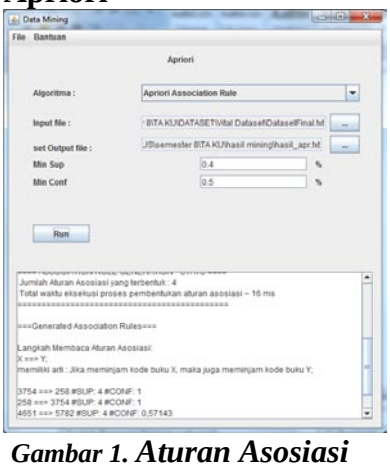
Karakteristik algoritma FP-Growth adalah struktur data yang digunakan adalah tree yang disebut dengan FP-Tree. Dengan menggunakan FP-Tree, algoritma FP-Growth dapat langsung mengekstrak frequent itemset dari FP-Tree.

Penggalian itemset yang frequent dengan menggunakan algoritma FP-Growth akan dilakukan dengan cara membangkitkan struktur data tree atau disebut dengan FP-Tree. Metode FP-Growth dapat dibagi menjadi 3 tahapan utama yaitu sebagai berikut [6]:

1. Tahap Pembangkitan conditional pattern base,
2. Tahap pebangkitan conditional FP-Tree, dan
3. Tahap pencarian frequent itemset.

III. IMPLEMENTASI

A. Apriori

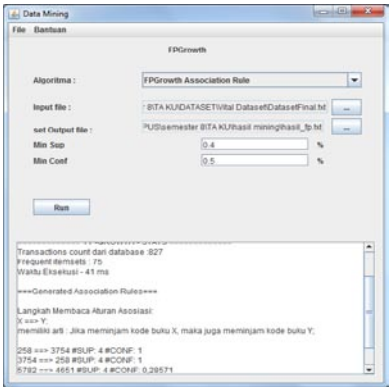


Gambar 1. Aturan Asosiasi yang terbentuk dari proses mining Apriori

Kode Judul Buku	Nilai Support	Nilai Confidence
258 ➔ 3754 Jika meminjam buku Teori dan Soal-soal Operations Research maka juga meminjam buku Dasar Teori Antrian Untuk Kehidupan Nyata	4/827 (0,4836%)	1 (100%)
3754 ➔ 258 Jika meminjam buku Dasar Teori Antrian Untuk Kehidupan Nyata maka juga meminjam buku Teori dan Soal-soal Operations Research	4/827 (0,436%)	1 (100%)
5782 ➔ 4651 Jika meminjam buku Accounting Theory : Teori Akuntansi Buku Satu maka juga meminjam buku Management Control System (Sistem Pengendalian Manajemen) Buku Edisi 11	4/827 (0,4836%)	0,28571 (28,571%)
4651 ➔ 5782 Jika meminjam buku Management Control System (Sistem Pengendalian Manajemen) Buku Edisi 11 maka juga meminjam buku Accounting Theory : Teori Akuntansi Buku Satu	3/827 (0,3627%)	0,57143 (57,143%)

Tabel 1. Aturan Asosiasi yang terbentuk dari proses mining Apriori

B. FP-Growth



Gambar 2. Aturan Asosiasi yang terbentuk dari proses mining FP-Growth

Kode Judul Buku	Nilai Support	Nilai Confidence
258 ➔ 3754 Jika meminjam buku Teori dan Soal-soal Operations Research maka juga meminjam buku Dasar Teori Antrian Untuk Kehidupan Nyata	4/827 (0,4836%)	1 (100%)
3754 ➔ 258 Jika meminjam buku Dasar Teori Antrian Untuk Kehidupan Nyata maka juga meminjam buku Teori dan Soal-soal Operations Research	4/827 (0,436%)	1 (100%)
5782 ➔ 4651 Jika meminjam buku Mnagement Control System (Sistem Pengendalian Manajemen) Buku Edisi 11 maka juga meminjam buku Accounting Theory : Teori Akuntansi Buku Satu	4/827 (0,4836%)	0,28571 (28,571%)
4651 ➔ 5782 Jika meminjam buku Pengantar Kesehatan Lingkungan maka juga meminjam buku Kesehatan Lingkungan	3/827 (0,3627%)	0,57143 (57,143%)

Tabel 2. Aturan Asosiasi yang terbentuk dari proses mining FP-Growth

C. *Hasil Perbandingan Kedua Algoritma*

Algoritma	Frequent Itemsets	Rule yang terbentuk	Waktu eksekusi program
Apriori	Dengan min support 30% terdapat 75 frequent	Dengan min confidence 50% terdapat 4 transaksi yang muncul bersamaan pada suatu buku.	Pada waktu eksekusi program waktu yang di butuhkan algoritma apriori ini yaitu 200 ms
FP-Growth	Dengan min support count 30% terdapat 75 frequent	Dengan memasukkan min conf sama dengan algoritma apriori ternyata hasilnya pun sama	Pada waktu eksekusi program untuk algoritma fp-growth waktu yang dibutuhkan yaitu 36 ms

Tabel 3. Tabel Hasil Perbandingan Apriori dan FP-Growth

Dari hasil penelitian di atas maka dapat di simpulkan bahwa kedua algoritma mempunyai kesamaan hanya saja pada kecepatan eksekusi program lebih cepat menggunakan algoritma fp-growth.

IV. HASIL DAN PEMBAHASAN

5.1 Kesimpulan

Dari permasalahan di atas mengenai data transaksi peminjaman buku, algoritma Apriori dan Frequent Pattern Growth dapat digunakan dalam perekomendasi buku perpustakaan dengan teknik data mining menggunakan aturan asosiasi untuk menentukan buku-buku yang sering dipinjam dalam waktu bersamaan. Hasil analisis kelayakan bahwa algoritma Apriori dan FP-Growth telah dijelaskan di atas bahwa aturan Asosiasi yang terbentuk dari kedua algoritma yaitu sama. Perbedaan terletak pada waktu eksekusi program dimana dengan menggunakan Apriori waktu yang digunakan untuk eksekusi program lebih banyak dibandingkan waktu eksekusi program menggunakan FP-Growth.

Dari hasil kinerja kedua algoritma semakn besar minimum support nya maka itemsets dan Association rule yang terbentuk tidak ada, maka dari itu dicari minimum support yang paling mendekati agar frequent itemsets dan Association rule yang terbentuk sempurna atau mendekati sempurna.

Maka dapat disimpulkan bahwa kedua algoritma tersebut layak diimplementasikan untuk perekomendasi peminjaman buku berdasarkan aturan asosiasi.

5.2 Saran

Agar penelitian ini terus berkembang, berikut saran-saran yang diusulkan :

1. Penelitian ini dapat dikembangkan dengan menggunakan metode aturan asociation rules data mining lainnya untuk dilakukan perbandingan.
2. Jumlah buku yang dapat dipinjam pengunjung ditingkatkan agar pola mining yang terbentuk semakin bervariasi.
3. Hasil dari penelitian ini dari rule yang terbentuk dapat dikembangkan menjadi pengetahuan untuk sistem pendukung keputusan.

V. Daftar Pustaka

[1] http://www.dinus.ac.id/about/upt/perpus

[2] Thomas Richard Stif Wijaya(2014): *"Aplikasi Data Mining Menggunakan Algoritma A-Priori Untuk Menentukan Aturan Asosiasi Pada Transaksi Peminjaman Buku Perpustakaan Universitas Dian Nuswantoro Semarang"*. Semarang : UDINUS.

[3] Goldie Gunadi dan Dana Indra Sensuse(2012): *"Penerapan Metode Data Mining Market Basket Analysis Terhadap Data Penjualan Produk Buku dengan Menggunakan Algoritma Apriori dan Frequent Pattern Growth (FP-Growth) : Studi Kasus Pada Percetakan PT. GRAMEDIA"*, Jurnal Telematika MKOM, Volume. 4 NISN:2085-725X

[4] Kusrini dan E. T. Luthfi. (2009). *"Algoritma Data Mining"*. Yogyakarta : Andi Offset.

[5] Jiawei Han dan Micheline Kamber, *"Data Mining: Concept and Techniques: Capture 6. Mining Association Rules in Large Databases"*, Simon Fraser University, 2009.

[6] Affriantari Rochmah, *"Perancangan Fitur Rekomendasi Film Di Website Solo Movie Dengan Menggunakan Metode Algoritma Apriori"* : Pustaka uns, 2010.

[7] Bapersip. Visi dan Misi [Online]. Available: http://bapersip.jatimprov.go.id/bapersip/profil.jsp?id=4 (2011).

[8] Samuel, David. 2008. *"Penerapan Struktur FP-Tree dan Algoritma FP-Growth dalam Optimasi Penentuan Frequent Itemset". Institut Teknologi Bandung.*

A Comparative Study of FP-growth Variations

Aiman Moyaid Said[A], Dr. P D D. Dominic[B], Dr. Azween B Abdullah[C]

Department of Computer and Information Sciences

Universiti Teknologi PETRONAS

Summary
Finding frequent itemsets in databases is crucial in data mining for purpose of extracting association rules. Many algorithms were developed to find those frequent itemsets. This paper presents a summarization and a comparative study of the available FP-growth algorithm variations produced for mining frequent itemsets showing their capabilities and efficiency in terms of time and memory consumption.

Keywords: Data mining, frequent itemsets, FP-growth.

1. INTRODUCTION

An association rule is defined as the relation between the itemsets, since its introduction in 1993 [1] the process of finding the association rules has received a great deal of attention. Today the extracting of association rules is still one of the main popular pattern discovery techniques in knowledge discovery and data mining (KDD).

The process of extracting the association rules can be viewed as two-phases: the first phase is to mine all frequent patterns; each of these patterns will happen at least as frequently as preset minimum support count (min_sup). The second phase is to produce strong association rules from the frequent patterns; these rules must assure minimum support and minimum confidence. The performance of discovering association rules is largely determined by the first phase, [8].

A lot of algorithms were proposed to optimize the performance of the FP-growth algorithm. In this paper we mainly restrict ourselves to study the performance of the FP-growth's Variations in term of running time and the memory usage.

2. RELATED WORKS

In fact, a broad variety of efficient algorithms for mining frequent itemsets have been developed. Agrawal et al in [1], introduced Apriori algorithm to find the frequent itemsets from market basket dataset. The Apriori algorithm adopts candidates' generations-and-testing methodology to produce the frequent itemsets. In the case of the long itemsets the Apriori approach suffer from the lack of the scalability, due to the exponential increasing of the algorithm's complexity.

FP-growth approach for mining frequent itemsets without candidate generation was proposed by Han in [2] . Its scalable frequent patterns mining method has been proposed as an alternative to the Apriori-based approach. The pattern growth approach adopts the divide-and-conquer methodology to produce the frequent itemsets.

This algorithm creates a compact tree-structure, FP-Tree, representing frequent patterns, which moderates the multi-scan problem and improves the candidate itemset generation. This algorithm is faster than others in the literature, this reported by the authors of this algorithm.

Several algorithms implicate the methodology of the FP-growth algorithm. In [3] Pei used the same approach for Mining closed frequent itemsets and max-patterns. Likewise, Pei suggested to Mining sequential patterns in [4].

Further Improvements of FP-growth Mining Methods were introduced. [5],[6],[7], adapted the similar approach of Han et al [2] for mining the frequent itemsets from the transactional database. The authors reported that these algorithms are more efficient than FP-growth.

Manuscript received May 5, 2009
Manuscript revised May 20, 2009

3. FP-GROWTH ALGORITHM REVISIT

FP-growth algorithm is an efficient method of mining all frequent itemsets without candidate's generation. The algorithm mine the frequent itemsets by using a divide-and-conquer strategy as follows: FP-growth first compresses the database representing frequent itemset into a frequent-pattern tree, or FP-tree, which retains the itemset association information as well.

The next step is to divide a compressed database into set of conditional databases (a special kind of projected database), each associated with one frequent item. Finally, mine each such database separately.

Particularly, the construction of FP-tree and the mining of FP-tree are the main steps in FP-growth algorithm.

For the explanation of the algorithm, we will use the following example. To find the frequent itemsets from transactional database DB (see Table 1).First, a scan of the database DB derivers a set of frequent 1-itemsets (L) which also include their support count. The set L is sorted in the order of descending support count, this ordering is important since each path of FP-tree will follow it.

Let the minimum support count be 3,then the set L={(f,4),(c,4),(a,3),(b,3),(m,3),(p,3)}.

Table 1: The transactional database DB

TID	Items
T1	f,a,c,d,g,i,m,p
T2	a,b,c,f,l,m,o
T3	b,f,h,j,o
T4	b,c,k,s,p
T5	a,f,c,e,I,p,m,n

Second, an FP-tree is constructed as follows: The root of the tree, labeled Null, is created. The database DB is scanned for the second time. The items in each transaction are processed in L order, and a branch is created for each transaction.

For example, the scan of the first transaction, "T1:f,a,c,d,g,I,m,p" which contains five items (f,c,a,m,p in L order).Only those items that are in the list of frequent itemsets L ,leads to constructions of the first branch of the tree with tree nodes {<f,1>,<c,1>,<a,1>,<m,1>,<p,1>} where <f,1> is linked as a child of the root. <c,1> is linked to <f,1>,<a,1> is linked to <c,1> ,<m,1> is linked to <a,1>, and <p,1> is linked to <m,1>.

The second transaction, because it shares items f, c and a, it shares the common prefix {f,c,a} with the previous branch and extends to the new branch {<f,2>,<c,2> , <a,2>,<m,1>, <p,1>} .increasing the count of the common prefix by 1.The new intermediate version of FP-tree, after adding two transactions from the database ,is given in Fig. 1.for the remaining transactions can be inserted in the same way (see Fig. 2).

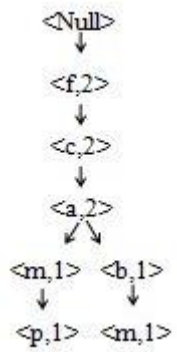


Fig.1 FP-tree for two transactions

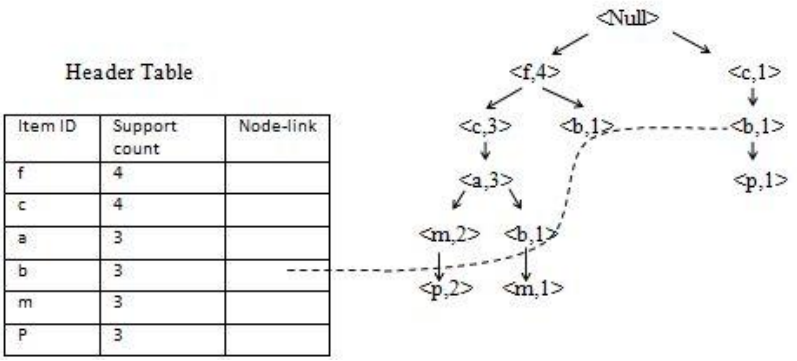


Fig. 2 Final FP-tree

To ease tree traversal, header table is built so that each item points to its occurrences in the tree via chain of node-link.

Using the compact tree structure (or FP-tree), the FP-growth algorithm mines all the frequent itemsets. The FP-tree is mined as follows. Begin from each frequent-1 pattern (as an initial suffix pattern), construct its conditional pattern base (a "subdatabase" which consists of the set of prefix paths in the FP-tree co-occurring with suffix pattern), then build its (conditional) FP-tree, and do mining recursively on such a tree. The patterns growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree.

In our example, according to L ,the complete set of frequent itemsets can be divided into subsets (6 for our example) without overlapping, first, frequent itemsets having items p (as an initial suffix pattern) ,which is the last item in L, rather than the first item. The reason for starting at the end of the list will become clear as we explain the FP-tree mining process. Second, the itemsets having item m but not p; third, the itemsets that have item b without both m and p; we continue this process to the end. Therefore, the last set will be the large itemsets only with f.

The item p occurs in two branches of the FP-tree of Fig.2.The occurrences of p can easily found by starting from the header table of p and following p's node-links. The paths formed by these branches are {<f,4>,<c,3>,<a,3>,<m,2>,<p,2>}and{<c,1>,<b,1>,<p,1} where samples with a frequent item p are {<f,2>,<c,2>,<a,2>,<m,2>,<p,2>}and{<c,1>,<b,1>,<p,1>}, which form its conditional pattern base ,these samples are the transactions that contain the branch of the tree with the existing of item p. Its conditional FP-tree contains only {<c, 3>} ,the other items are not included because its support count is less than 3.The generated frequent itemset that satisfy the minimum support count is {<c,3>,<p,3>},all the other itemsets are below the minimum support count.

The next subsets of frequent itemsets are those with m item and without p. The FP-tree recognizes the paths {<f,4>,<c,3>,<a,3>,<m,2>}and {<f,4>,<c,3>,<a,3>,<b,1> ,<m,1>},or the related accumulated samples {<f,2>,<c,2> ,<a,2>,<m,2>} and {<f,1>,<c,1>,<a,1>,<b,1>,<m,1>} . Analyzing the samples we find the frequent itemset {<f,3>,<c,3>,<a,3>,<m,3>}.

Similar to subset 3 to 6 the same process is done in our example, additional frequent itemsets can be mined.These are itemsets {f,c,a} and {f,c},but they are already subset of frequent itemsets {f,c,a,m}.Therefore ,the final set of frequent itemsets is {{c,p},{f,c,a,m}}.

4. FP-GROWTH VARIATIONS

Several optimization techniques are added to FP-growth algorithm. In this paper, we investigate the performance of three algorithms, namely AFOPT Algorithm, Nonordfp algorithm and Fpgrowth* algorithm .Our goal is not to go into many details about the algorithms but show the basic optimization ideas and the different of the performance in term of running time and memory consumption. In the following we will illustrate what are the main optimization ideas in each algorithm.

- **AFOPT ALGORITHIM**

Liu et al in [5] investigated the algorithmic performance space of the Fpgrowth algorithm. They specified the problem of conditional databases construction (particularly the number of the conditional databases constructed and the mining cost of each individual conditional database) in Fpgrowth algorithm, which have direct effect on the performance of the algorithm. They studied the problem of enhancing the Fpgrowth algorithm from four perspectives to come with the best strategy for mining the frequent itemsets. These perspectives are the item search order (in what order the search space is explored), conditional database representation, conditional database construction strategy and tree traversal strategy.

For the first part of the problem, the number of the conditional databases constructed can differ very much using different items search orders. The dynamic ascending order is able to minimize the number and /or the size of the conditional database constructed in subsequent mining, AFOPT algorithm adapt this kind of items search order which is also used by Fpgrowth.

For the second part of the problem, the mining cost of each individual conditional database is heavily depends on its representation (tree-based or array-based).AFOPT algorithm use adaptive representation ,tree-based structure in the case of dense dataset and array –based representation in the case of sparse dataset. In additions to the conditional database representation the size and the conditional database construction strategy have effect on the mining cost of each individual conditional database, two type of

the conditional database construction strategy (physical construction or pseudo-construction).
The dynamic ascending frequency search order can make the subsequent conditional databases shrink rapidly. As a result, it is useful to use the physical construction strategy with the dynamic ascending frequency order. The traversal cost of a tree us minimal using the top-down traversal strategy, AFOPT algorithm uses dynamic ascending frequency order for both the search space exploration and prefix-tree construction, it uses the top-down traversal strategy. as a summery AFOPT algorithm utilizes dynamic ascending frequency for the item search space ,adaptive representation for the conditional database format ,physical construction for the conditional database construction, and top-down traversal strategy for the tree traversal.

- **NONORDFP**

The running time and the space required for the Fpgrowth algorithm were the motivation for Nonordfp algorithm. Rácz in [9] dealt with the implementation issues, data structures, memory layout, I/O, and library functions.
A compact, memory efficient representation of an FP-tree by using Trie data structure, with memory layout that allows faster traversal was introduced, to deal with the running time and space requirement problem. This compact representation of FP-tree allows faster allocation, traversal, and optionally projection. It contains less administrative information about the items in the database (no labels for the items are stored in the node, no header lists and children are required), and allows more recursive steps to be carried out on the same data structure, with no need to rebuild it.

- **FPGROWTH* ALGORITHM**

Depending on a numerous experiments were done by Grahne et al [10], they found that 80% of the CPU time was used for traversing FP-trees. Consequently, they employed the array-based to reduce the traversal time of the FP-trees. Fpgrowth* algorithm uses FP-tree data structure in combination with the array-based and incorporates various optimization techniques.

In the case of sparse data set the array-based technique work very well, the array save traversal time for all items and the next level of FP-trees can be initialized directly. While in the case of dense data set, the FP-tree is more compact. To deal with this problem they proposed optimizing technique that help the algorithm to estimate if the data set is sparse or dense, by counting the number of the nodes in each level of the tree which done during the construction of each FP-tree. If the data set turns to be dense data set then no need to calculate the array for the next level of the FP-tree. In the case of sparse data set, the calculation of the array for the next FP-tree is required.

5. COMPARISON OF THE ALGORITHMS

To verify the efficiency of the FP-growth variation algorithms a lot of experiments were conducted. All the experiments are conducted on Core 2 Duo 2.00 GHZ CPU, 2.00 GB memory and hard disk 160 GB. The operating system is ubuntu 8.10.We test the AFOPT algorithm, Nonordfp Algorithm, Fpgrowth* algorithm [11] and the original Fpgrowth algorithm [12].To evaluate the behavior of the four algorithms different datasets and different support threshold were used, in the following subsections the type of the data sets ,the running time and the memory consumption are illustrates:

5.1 Datasets

The data is challenging due to the number of characteristics which are the number of the records, and the sparseness of the data (each records contains only small portion of items).

In our experiments we chose different dataset with different prosperities, to prove the efficiency of the algorithms, Table 2 shows the datasets and the characteristics.

Table 2: The Datasets

Data set	#Items	Avg. Length	#Trans	Type	Size
T10I4D100k	1000	10	100,000	Sparse	3.93 MB
T40I10D100K	1000	40	100,000	Sparse	14.8 MB
Mushroom	119	23	8,124	Dense	557 KB
Connect4	150	43	67557	Dense	8.89 MB

5.2 Running Time: The running time is real time, system time and user time. Figs 3, Fig 4, Fig 5, and Fig 6 depict the time needed in seconds for each one of the algorithms.

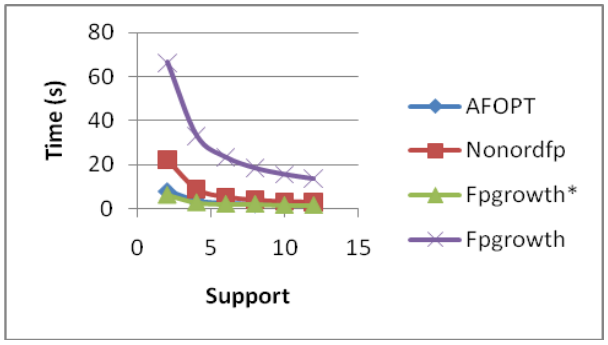


Fig 3 Execution time at various support levels on T10I4D100k

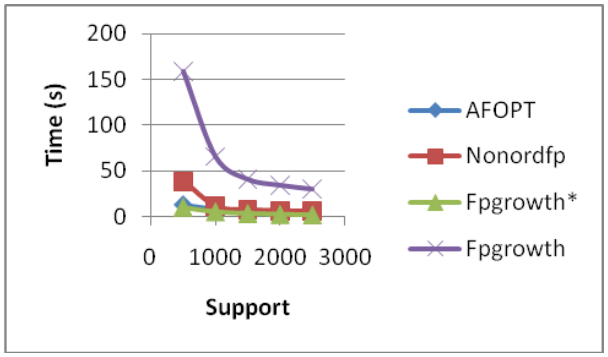


Fig 4 Execution time at various support levels on T40I10D100K

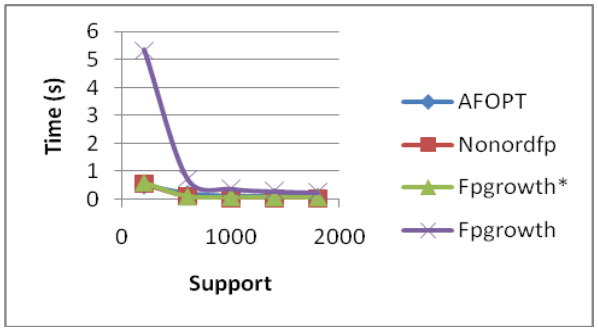


Fig 5 Execution time at various support levels on Mushroom

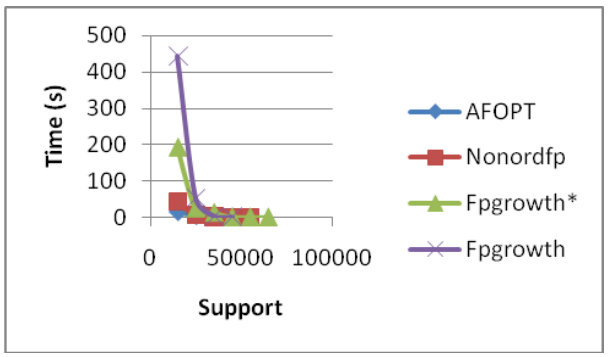


Fig 6 Execution time at various support levels on Connect4

It is clear that with the T10I4D100k data set Fpgrowth* algorithm outperforms all the other algorithms. On T40I10D100K data set there is obvious performance competition among both Fpgrowth* algorithm and AFOPT algorithm. The running times for the AFOPT algorithm, Nonordfp algorithm, and Fpgrowth* algorithm are near in the case of mushroom data set. For the connect4 data set, we should mention that some algorithms had problem, segmentation fault, with some values of support due to the huge number of the frequent itemsets satisfy those thresholds values and some took long time to find the frequent itemsets.

5.3 Memory Consumption: In this section, we calculate the total number of memory consumption for each algorithm .All the experiments are done on the same sets of data. As shown in Fig 7,Fig 8,Fig 9,and Fig 10 the support values and the amount of memory for each one. We observe that, Nonordfp algorithm remains stable over the whole range of support values on T10I4D100k.The stability in memory consumption is also observe for Fpgrowth* algorithm and AFOPT algorithm for the high values of support.

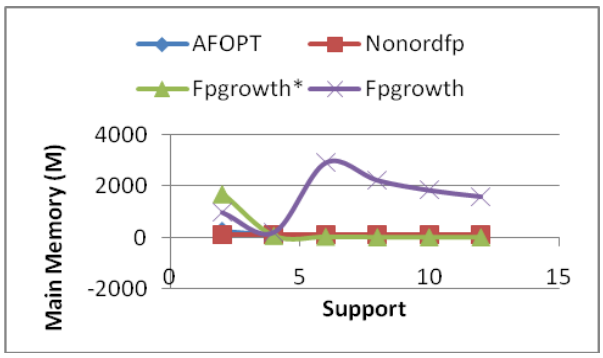


Fig 7 Memory usage on T10I4D100k

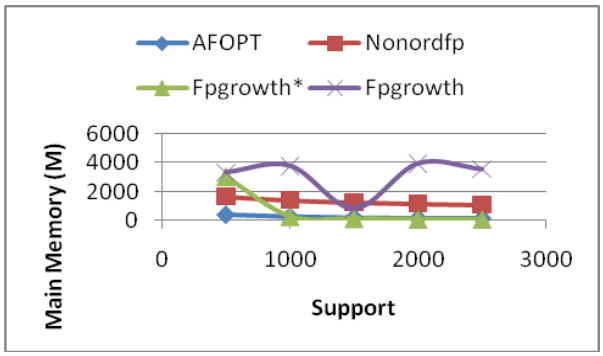


Fig 8 Memory usage on T40I10D100K

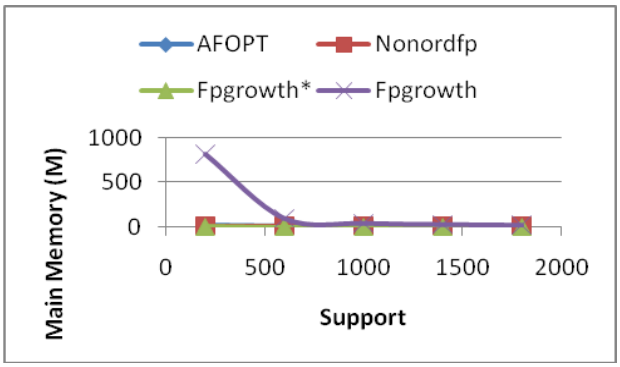


Fig 9 Memory usage on Mushroom

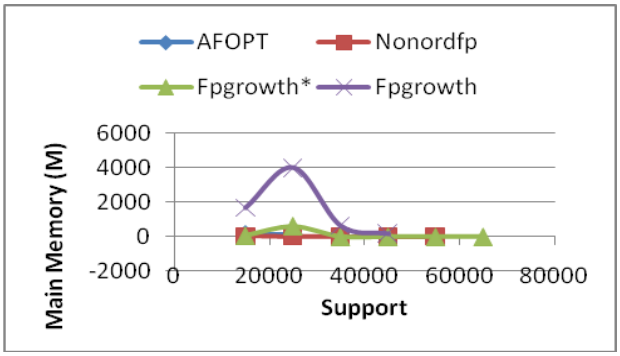


Fig 10 Memory usage on Connect4

AFOPT algorithm keeps its stable consumption of the memory on T40I10D100K. It further confirmed the fact that AFOPT algorithm is stable with sparse data sets. In Fig 9 the competition between the three algorithms is clear, the memory usage is competitive.On Connect4 data set, Fpgrowth* shows stability in the case of the high support thresholds while Nonordfp algorithm remain stable for the low values of support.

6. CONCLUSION

In this paper, we dealt with FP-growth's Variation algorithms. We restricted ourselves to the "Classic" frequent itemsets problem, which is the mining of all frequent itemsets that exist in market basket-like data with respect to support thresholds. The execution time and the memory usage were recorded to see which algorithm is the best. For the time consumption AFOPT algorithm took advantage for most of the data set even though it suffers from segmentation fault in the low support values on connect4 data set. However, for the memory consumption Nonordfp algorithm remains stable for almost all the type of the data set except for the high support values on Connect4. Unfortunately, there is no algorithm work for the all situations. The main restricted for the successful of the algorithm is statistical prosperities of the data set.

REFERENCES

[1] Agrawal , R. , Imieliński , T. , & Swami , A."Mining association rules between sets of items in large databases".In proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207-216, Washington, DC, 1993.
[2] Han, J. , Pei, J. , & Yin, Y. "Mining frequent patterns without candidate generation". In Proc. ACM-SIGMOD Int. Conf. Management of Data (SIGMOD '96), Page 205-216, 2000.
[3] J. Pei, J. Han & R. Mao. CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets", DMKD'00, 2000.
[4] J. Pei, J. Han, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. ICDE'01, 2001.
[5] Liu,G. , Lu ,H. , Yu ,J. X., Wang, W., & Xiao, X.. "AFOPT:An Efficient Implementation of Pattern Growth Approach", In Proc. IEEE ICDM'03 Workshop FIMI'03, 2003.
[6] Grahne, G. , & Zhu, J. "Fast Algorithm for frequent Itemset Mining Using FP-Trees",IEEE Transactions on Knowledge and Data Engineer,Vol.17,NO.10, 2005 .
[7] Gao, J. "Realization of new Association Rule Mining Algorithm" Int. Conf. on Computational Intelligence and Security ,IEEE, 2007.
[8] Han, J. , & Kamber, M. "Data Mining :concepts and techniques" , second edition, The Morgan Kaufmann Series in Data Management Systems, 2006.
[9] Balázes Rácz," nonordfp: An FP-Growth Variation without Rebuilding the FP-Tree**",** 2nd Int'l Workshop on Frequent Itemset Mining Implementations FIMI2004
[10] Grahne O. and Zhu J. "Efficiently Using Prefix-trees in Mining Frequent Itemsets", In Proc. of the IEEE ICDM Workshop on Frequent Itemset Mining, 2004.
[11] http://fimi.cs.helsinki.fi/
[12] http://adrem.ua.ac.be/~goethals/software/

Aiman Moyaid Said received bachelor degree in computer information system from Yarmouk University, Jordan in 2007 .And currently he is doing his master degree in computer information sciences, at Universiti Teknologi PETRONAS. His research interests include Association Rule Mining, Clustering and application of data mining to problems in retailer industry.

Dr. P.D.D.Dominic obtained his M.Sc degree in operations research in 1985, MBA from Regional Engineering College, Tiruchirappalli, India during 1991, Post Graduate Diploma in Operations Research in 2000 and completed his Ph.D during 2004 in the area of job shop scheduling at Alagappa University, Karaikudi, India. Since 1992 he has held the post of Lecturer in the Department of Management Studies, National Institute of Technology (Formally Regional Engineering College), Tiruchirappalli- 620 015, India. Presently he is working as a Senior Lecturer, in the Department of Computer and Information Science, Universiti Teknologi PETRONAS, Malaysia. His fields of interest are Operations Management, KM, E-business and Decisions Support Systems. He has published technical papers in International, National journals and conferences.

Dr Azween Abdullah obtained his bachelors degree in Computer Science in 1985, Master in Software Engineering in 1999 and his Ph.d in computer science in 2003. His work experiences includes twenty years in institutions of higher learning in both the management and academic capacities, and fifteen years in commercial companies as Software Developer and Engineer, Systems Analyst and IT/MIS and educational consultancy and training. He has spent more than a decade with leading technology firms and universities as a process analyst, senior systems analyst, project manager, and lecturer. He have participated in and managed several software development projects. These have included the development of management information systems, software process improvement initiatives design and implementation, and several business application projects.

His area of research specialization includes computational biology, system survivability and security, autonomic computing and self-healing and regenerating systems, formal specifications and network modeling. His contributions include publishing several journal and refereed conference papers and in the development of programs to enhance minority involvement in bridging the ICT digital gap. Currently he is working on two projects funded by the Ministry of Science Technology and Innovation.

ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/330341180

ANALISA POLA BELANJA MENGGUNAKAN ALGORITMA FP GROWTH, SELF ORGANIZING MAP (SOM) DAN K MEDOIDS

Article · April 2017

DOI: 10.24176/simet.v8i1.995

CITATIONS
0

READS
120

3 authors, including:

Imam Ghozali
Universitas Diponegoro
134 PUBLICATIONS **1,488** CITATIONS
SEE PROFILE

Wibowo Harry Sugiharto
Universitas Muria Kudus
16 PUBLICATIONS **3** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Accountability and Fraud Type Effects on Fraud Detection Responsibility View project

pp. 524-537 Analysis of Auditor Performance by Using Covariance Based Structural Equation Modeling: A Study of Public Accounting Firms in Indonesia View project

All content following this page was uploaded by Wibowo Harry Sugiharto on 02 February 2019.

The user has requested enhancement of the downloaded file.

ANALISA POLA BELANJA MENGGUNAKAN ALGORITMA FP GROWTH, SELF ORGANIZING MAP (SOM) DAN K MEDOIDS

Muhammad Imam Ghozali
Fakultas Teknik, Program Studi Teknik Informatika
Universitas Muria Kudus
Email: imam.ghozali@umk.ac.id

Ri'fan Zaenal Ehwan
SMA N 1 Jakenan Pati
Email : revan.ze@gmail.com

Wibowo Harry Sugiharto
Fakultas Teknik, Program Studi Teknik Informatika
Universitas Muria Kudus
Email: wibowo.harry@umk.ac.id

ABSTRAK

Dalam sebuah bisnis, diperlukan upaya memaksimalkan keuntungan. Diantaranya dengan melakukan promosi. Ketepatan promosi dapat dipelajari dari database sebuah perusahaan ritel utamanya pola belanja pada produk yang biasa dibeli bersamaan. Informasi tentang pola belanja pelanggan yang tidak akurat menyebabkan kebijakan promosi tidak tepat dan efisien.Salah satu upaya lazim untuk memperoleh dan menggali pola belanja pelanggan adalah menggunakan data mining yang dikenal sebagai *Knowledge Discovery in Database* (KDD). Pendekatan yang biasa digunakan adalah asosiasi. Permasalahannya aturan asosiasi cenderung mengabaikan dataset yang besar. Untuk mengatasi hal tersebut dilakukan klasifikasi barang yang dibeli dan tidak dibeli bersamaan. Algoritma *Self Organizing Map* (SOM) dan *K-Medoids* cocok untuk diterapkan dalam meng*cluster* dataset besar. Penelitian ini menguji kevalidan dan kecepatan algortima *Self Organizing Map* (SOM) dan *K-Medoids* jika dikombinasi dengan *Frequent Pattern-Growth* (*FP-Growth*).

Kata kunci: analisis keranjang pasar, rekomendasi produk, *self organizing map*, *k-medoids*, *fp-growth*.

ABSTRACT

In a business, it takes effort to maximize profits. Among them with promotions. The accuracy of the promotion can be learned from the database a major retail company spending patterns on products commonly purchased together. Information on customer shopping patterns inaccurate cause improper promotion policy and efisien.Salah a common effort to acquire and explore the shopping patterns of customers is using data mining known as Knowledge Discovery in Databases (KDD). The approach used is association. The problem tends to ignore the rules of association of large datasets. To overcome this problem do the classification of goods purchased and not purchased together. Algorithm Self Organizing Map (SOM) and K-Medoids mengcluster suitable to be applied in large datasets. This study tested the validity and speed algorithms Self Organizing Map (SOM) and K-Medoids when combined with Frequent Pattern-Growth (FP-Growth).

Keywords**:** *market basket analysis, recomendation product, promotion product, association rule mining*, *k-medoids*, *fp-growth*.

1. PENDAHULUAN

Sebuah bisnis diperlukan upaya untuk memaksimalkan keuntungan. Keuntungan tersebut akan dapat diperoleh jika perusahaan dapat memanfaatkan potensi dan peluang usaha yang ada. Pelanggan/ konsumen adalah sumber terbesar pendapatan sebuah usaha. Penjual dapat meningkatkan pendapatan dengan menggunakan strategi pemasaran yang tepat dan akurat. Salah satunya adalah menawarkan barang lain yang kemungkinan besar akan dibeli juga oleh pelanggan secara bersamaan dengan barang yang sudah direncanakan untuk dibeli sebelumnya. Permasalahannya adalah barang apakah yang hampir pasti dibeli oleh pelanggan dan dari manakah informasi barang rekomendasi tersebut kita peroleh? Database toko yang dimiliki toko tersebut dapat dianalisis sehingga memeroleh informasi mengenai produk yang

sering dibeli secara bersamaan. Metode yang lazim untuk memperoleh dan menggali pola pelanggan tersebut adalah dengan data mining yang biasa dikenal sebagai *Knowledge Discovery in Database* (KDD). Data mining dapat mengambil informasi yang penting dari database yang besar dan dibutuhkan manusia. Informasi atau pengetahuan tersebut menunjukkan konsep, alur, aturan maupun pola dan bentuk dari sekumpulan data yang berbeda-beda. Data mining dapat diterapkan dalam sebagian besar *field.*Misalnya untuk membangun trend, memprediksi masa depan, dan menganalisis faktor kunci yang dibutuhkan untuk sebuah permasalahan yang kompleks.

Banyak algoritma yang diusulkan untuk menemukan pola asosiasi dan algoritma pertama yang paling efisien untuk menemukan pola asosiasi adalah *apriori* [9][10]. Alasan utamanya karena *Frequent Pattern-Growth* (*FP-Growth*) merupakan algoritma yang dapat menghemat waktu dan media penyimpanan terlebih untuk database yang besar [12].

Algoritma Apriori adalah algoritma *association rule* konvensional, yang membutuhkan waktu yang sangat besar saat proses pemindaian database. Hal ini tentu saja berpengaruh pada efisiensi waktu dalam data mining. Algoritma yang diusulkan untuk mengatasi kelemahan tersebut adalah *Frequent Pattern-Growth* (*FP-Growth*) yang disebut yang tidak perlu untuk menghasilkan *generate candidate item sets* dan dianggap lebih efisien. *Frequent Pattern-Growth* (*FP-Growth*) dibangun dengan membaca kumpulan data satu transaksi pada waktu dan pemetaan setiap transaksi ke dalam bagian *Frequent Pattern-Tree* (*FP-Tree*) [2].

Permasalahan yang muncul selanjutnya dalam aturan asosiasi adalah masalah teknis yang berhubungan dengan teknik rekomendasi yang paling umum dimana aturan asosiasi cenderung mengabaikan *itemset* besar. Akibatnya rekomendasi *item* kurang tepat karena informasi tentang produk ritel tidak tersedia [8]. Untuk mengatasi masalah tersebut, atribut yang ada di*cluster* untuk membentuk kelompok atribut yang sama dan kemudian menentukan pola asosiasi pada masing-masing kelompok *Apriori*, sehingga dapat mempermudah proses mencari rekomendasi produk. Beberapa algoritma *clustering* diantara *K-Means, Fuzzy C-Means, Self Organizing Map* (SOM)*, K-Medoids* [9].

Dalam penelitian ini akan dikomparasikan kombinasi *Self Organizing Map* (SOM) dengan *Frequent Pattern-Growth* (*FP-Growth*), *K-Medoids* dengan *Frequent Pattern-Growth (FP-Growth)* dan *Frequent Pattern-Growth* (*FP-Growth*) yang berdiri sendiri tanpa di *clustering* dengan algoritma apapun sebelumnya. Algoritma *Self Organizing Map* (SOM) dipilih karena *Self Organizing Map* (SOM) merupakan algoritma *clustering* model aglomerative dan partitif selain itu juga dikenal sebagai algoritma non liner, teratur, pemetaan dari input data dimensi tinggi ke dalam *array* dimensi rendah [4][5].. Algoritma *K-Medoids* dipilih sebagai perbandingan untuk mendampingi *Self Organizing Map* (SOM) dan *Frequent Pattern-Growth* (*FP-Growth*). Pada dasarnya, algoritma *clustering* yang paling sering dipakai adalah *K-Means*. Namun, *K-Means* memiliki kekurangan yakni sensitif terhadap *outlier* dalam data dan hal ini bisa dijawab oleh *K-Medoids* yang tidak terpengaruh terhadap *outlier* data sehingga hasil *clustering* menjadi lebih akurat.

1.1 Analisis Pola Belanja Pasar

Membahas tentang pola belanja pelanggan tidak bisa dipisahkan dengan *Assosiation Rule* Mining atau analisis keranjang pasar. *Association Rule Mining* merupakan salah satu metode data mining yang berfokus pada menemukan pola pembelian dengan mengekstraksi asosiasi atau kejadian dari data transaksional sebuah toko. Analisis keranjang pasar bermula dari transaksi-transaksi yang berisi satu atau lebih barang, dan beberapa informasi sementara dari transaksi tersebut. Untuk melakukan analisis keranjang pasar, berikut langkah-langkah:

1) Tentukan nilai *Minimum Support* yang diinginkan. *Minimum Support* merupakan ambang batas minimum jumlah *itemset* yang diperbolehkan, jika jumlah *item*nya di bawah ambang batas maka *item* tersebut akan dieliminasi.
2) Menetapkan *frequent itemset* (kumpulan *item* yang muncul secara bersamaan), dengan cara mengambil *itemset* yang memiliki frekuensi *itemset* minimal sebesar *Minimum Support* sebelumnya.
3) Dari semua *frequent itemset*, hasilkan aturan asosiasi yang memenuhi nilai *Minimum Support.*

Analisis keranjang pasar didasarkan pada tiga matrik: *Support*, *Confidence* dan *Lift*. Ketiga matrik tersebut berasal dari catatan transaksi untuk bisnis [1].

Support

Matrik pertama ditetapkan untuk analisis keranjang pasar adalah *Support*, yang merupakan probabilitas dari asosiasi (probabilitas dari dua *item* yang dibeli bersama-sama). *Support* dihasilkan dari berapa kali jumlah *item* A dan B terjadi bersamaan dalam transaksi yang sama dibagi dengan jumlah total dari transaksi tersebut. Untuk menghitung nilai *Support* digunakan persamaan (1) sebagai berikut:

$$Support = P(A \cap B) = \frac{\text{Jumlah transaksi yang memuat A dan B}}{\text{total Jumlah transaksi}} \quad (1)$$

Confidence

Confidence dihasilkan dari seberapa kuat hubungan produk yang sudah dibeli.
Untuk menhitung nilai *Confidence* digunakan persamaan (2) sebagai berikut:

$$Confidence = P(B/A) = \frac{Support(A \cap B)}{P(A)} \quad (2)$$

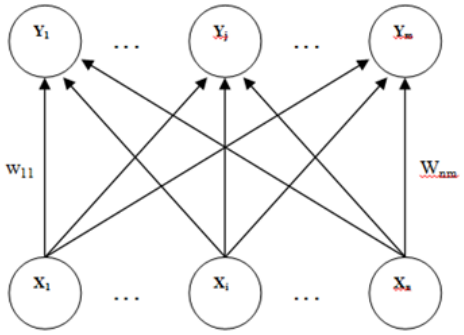
Lift

Lift Ratio mengukur seberapa penting *rule* yang teleh terbentuk berdasarkan nilai *support* dan *confidence*. *Lift Ratio* merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar produk A dibeli bersamaan dengan produk B.
Untuk menghitung nilai *Lift Ratio* digunakan persamaan (3) sebagai berikut:

$$Lift\ Ratio = \frac{Support(A \cap B)}{Support(A) * support(B)} \quad (3)$$

1.2 Algoritma Self Organizing Map (SOM)

Self Organizing Map (SOM) merupakan salah satu metode dalam Jaringan Syaraf Tiruan (*Neural Network*) yang menggunakan pembelajaran tanpa pengarahan (*unsupervised learning*). Metode pembelajaran tanpa pengarahan tidak memerlukan target, selama proses pembelajaran unit – unit input yang hampir sama dikelompokkan dalam kelompok tertentu yang berada di unit output (*cluster units*). Arsitektur *Self Organizing Map* (SOM) terdiri dari 1 lapisan input dan 1 lapisan output. Setiap unit pada lapisan input dihubungkan dengan semua unit di lapisan output dengan suatu bobot keterhubungan yang ditunjukkan pada Gambar 1.



Gambar 1. Algoritma *Self Organizing Map* (SOM)

1.3 Algoritma Frequent Pattern-Growth (FP-Growth)

Algoritma yang biasa dipakai dalam mencari *frequent item set* antara lain algoritma Apriori dan algoritma *Frequent Pattern-Growth* (*FP-Growth*). Pada penelitian ini akan dibahas bagaimana pencarian *frequentitemset* mengunakan algoritma *Frequent Pattern-Growth* (*FP-Growth*). *Frequent Pattern Tree* (*FP-Tree*) digunakan bersamaan dengan algoritma *Frequent Pattern-Growth* (*FP-Growth*) untuk menentukan *frequent item set* (data yang paling sering muncul) dari sebuah *dataset*. Algoritma *Frequent Pattern-Growth* (*FP-Growth*) adalah salah satu cara alternatif untuk menemukan himpunan data yang paling sering muncul tanpa menggunakan generasi kandidat [14]. *Frequent Pattern-Growth* (*FP-Growth*) membangun konstruksi data *Frequent Pattern Tree* (*FP-Tree*) yang sangat dikompresi, dan mengurangi data asli. Algoritma *Frequent Pattern-Growth* (*FP-Growth*) melakukan *scan* database yang sama sebanyak dua kali. *Scanning* database yang pertama, kita dapat memperoleh *frequent 1-item-set,* dan *scanning* database yang kedua, kita dapat memfilter database *non-frequent item,* selebihnya, *Frequent Pattern Tree* (*FP-Tree*) dihasilkan secara bersamaan. Akhirnya, dapat diperoleh aturan asosiasi dengan menggunakan *Frequent Pattern Tree* (*FP-Tree*) [13].

Kelemahan *Apriori* yang selalu melakukan *scanning* database secara berulang-ulang membuat *Apriori* ini kurang efektif. Berbeda dengan *Frequent Pattern Tree* (*FP-Tree*) yang digunakan bersamaan dengan algoritma *Frequent Pattern-Growth* (*FP-Growth*) yang hanya memerlukan dua kali scanning database untuk membuat *frequent item set*. Dengan menggunakan *Frequent Pattern Tree* (*FP-Tree*), algoritma *Frequent Pattern-Growth* (*FP-Growth*) dapat langsung mengekstrak *frequent Item set* dari

Frequent Pattern Tree (*FP-Tree*) yang telah terbentuk. *Frequent Pattern Tree* (*FP-Tree*) didefinisikan sebagai berikut [14] :

Sebuah *root* yang diberi label *null*, sekumpulan *sub-tree* yang beranggotakan *item-item* tertentu, dan sebuah tabel *frequent header*. Setiap simpul dalam *Frequent Pattern Tree* (*FP-Tree*) mengandung tiga *field*, yaitu:

1) *Item-name*: menginformasikan *item* yang dipresentasikan oleh simpul tersebut.
2) *Count*: mempresentasikan jumlah transaksi yang melewati simpul tersebut.
3) *Node-Link*: penghubung yang menghubungkan simpul-simpul dengan *item-name* yang sama, atau *null* jika kosong.

1.4 Algoritma K-Medoids

Untuk melakukan *clustering* dengan metode partisi dapat menggunakan *K-Means* dan *K-Medoids*. *K-Means* merupakan suatu algoritma peng*cluster*an yang cukup sederhana yang mempartisi *dataset* kedalam beberapa *cluster* k. Algoritmanya cukup mudah untuk diimplementasi dan dijalankan, relatif cepat, mudah disesuaikan dan banyak digunakan [14]. Kelemahan-kelemahan dari algoritma *K-Means* yaitu [20] :

1) Ketika jumlah data yang tidak begitu banyak, pengelompokan awal akan menentukan *cluster* secara signifikan.
2) Jumlah *cluster* K harus ditentukan terlebih dahulu.
3) *Cluster* yang asli tidak diketahui, dengan menggunakan data yang sama, jika dimasukkan dalam urutan yang berbeda dapat menghasilkan *cluster* yang berbeda jika jumlah data sedikit.
4) Kelemahan dari aritmatika mean tidak kuat untuk *outlier*, sangat jauh data dari *centroid* memungkinkan mempengaruhi *centroid* yang asli.

Algortima *K-Medoids*, juga dikenal sebagai *partitioning around Medoids*, adalah varian dari metode *K-Means*. Hal ini didasarkan pada penggunaan *Medoids* bukan dari pengamatan *mean* yang dimiliki oleh setiap *cluster,* dengan tujuan mengurangi sensitivitas dari partisi yang dihasilkan sehubungan dengan nilai-nilai ekstrim yang ada dalam *dataset* [3]. Algoritma *K-Medoids* hadir untuk mengatasi kelemahan Algoritma *K-Means* yang sensitif terhadap *outlier* karena suatu objek dengan suatu nilai yang besar mungkin secara substansial menyimpang dari distribusi data [6].

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan jenis penelitian eksperimen, dengan melakukan ujicoba terhadap dataset yang telah dikumpulkan. Tahapan-tahapan penelitian ini adalah sebagai berikut:

1) Pengumpulan Data
   Pengumpulan data merupakan langkah awal pada suatu penelitian. Data yang digunakan pada penelitian ini adalah data transaksi penjualan Dr. Tariq Mahmood.
2) Pengolahan Data Awal
   Pengolahan awal (*Pre processing*) merupakan tahap untuk mempersiapkan data yang telah diperoleh dari tahap pengumpulan data, yang akan digunakan pada tahap selanjutnya. Pada tahap ini peneliti mengubah format data tanpa mengubah isi data agar mudah untuk diolah menggunakan aplikasi rapidminer.
3) Desain Eksperimen
   Tahapan ini akan membahas alurdesain eksperimen yang digunakan pada penelitian supaya prosesnya sistematis dan menghasilkan temuan yang valid.
4) Eksperimen dan Pengujian
   Tahapan ini akan mengeksekusi tahapan penelitian dengan teknik pengujian yang akan telah dipersiapkan sebelumnya.
5) Evaluasi Penelitian
   Tahapan ini akan membahas hasil evaluasi dari eksperimen yang telah digunakan.

Dataset ini merangkum 304 atribut dimana 303 diantaranya merupakan item barang yang dibeli. Salah satu datanya menunjukkan ID transaksi penjualan yang dilakukan dalam sebuah supermarket. Terdapat 1362 record transaksi dalam dataset ditunjukkan pada Gambar 2.

	A	G	H	I	J	K	L	M	N	O	P	Q
1	Basket ID	Sugar Cookies	Onions	Deli Ham	Dishwasher Detergent	Beets	40 Watt Lightbulb	Ice Cream	Cottage Cheese	Plain English Muffins	Strawberry Soda	Vanilla Ice Cream
2	C11867	false	false	false	false	false	false	false	false	false	false	false
3	C5096	false	false	false	false	false	false	false	false	false	false	false
4	C4295	false	false	false	false	false	false	false	false	false	false	false
5	C2837	false	false	false	false	false	false	false	false	false	false	false
6	C2693	false	false	false	false	false	false	false	false	false	false	false
7	C3497	false	false	false	false	false	false	false	false	false	false	false
8	C2696	false	false	false	false	false	false	false	false	false	false	false
9	C1895	false	false	false	false	false	false	false	false	false	false	false
10	C9524	false	false	false	false	false	false	false	false	false	false	false
11	C9527	false	false	false	false	false	false	false	false	false	false	false
12	C8726	false	false	false	false	false	false	false	false	false	false	false
13	C9383	false	false	false	false	false	false	false	false	false	false	false
14	C8582	true	true	false	false	false	false	true	false	false	false	false
15	C10133	false	true	false	false	false	false	false	false	false	false	false
16	C6983	false	false	false	false	false	false	false	false	false	false	false
17	C3641	false	false	false	false	false	false	false	false	false	false	false
18	C5246	false	false	false	false	false	false	false	false	false	false	false
19	C10136	false	false	false	false	false	false	false	false	false	false	false
20	C7787	false	false	false	false	false	false	false	false	false	false	false
21	C3644	false	false	false	false	false	false	false	false	false	false	false

Gambar 2. Data Transaksi Penjualan

Keterangan:

1) Basket ID seperti C11867, C5096, C4295, C2837, C2693, C2696, C1895, C9524 merupakan ID Transaksi penjualan barang.
2) Sugar Cookies, Onions, Delli Ham, Dishwasher Detergent, Beets, 40 Watt Lightbulb, Ice Cream, Cottage Cheese, Plain English Muffins, Strawberry Soda, Vanila Ice Cream dll merupakan barang-barang yang dijual di Supermarket.
3) False menandakan bahwa barang tersebut tidak dibeli oleh pembeli.
4) True menandakan bahwa barang tersebut dibeli oleh pembeli.

2.1 Pengolahan Data Awal

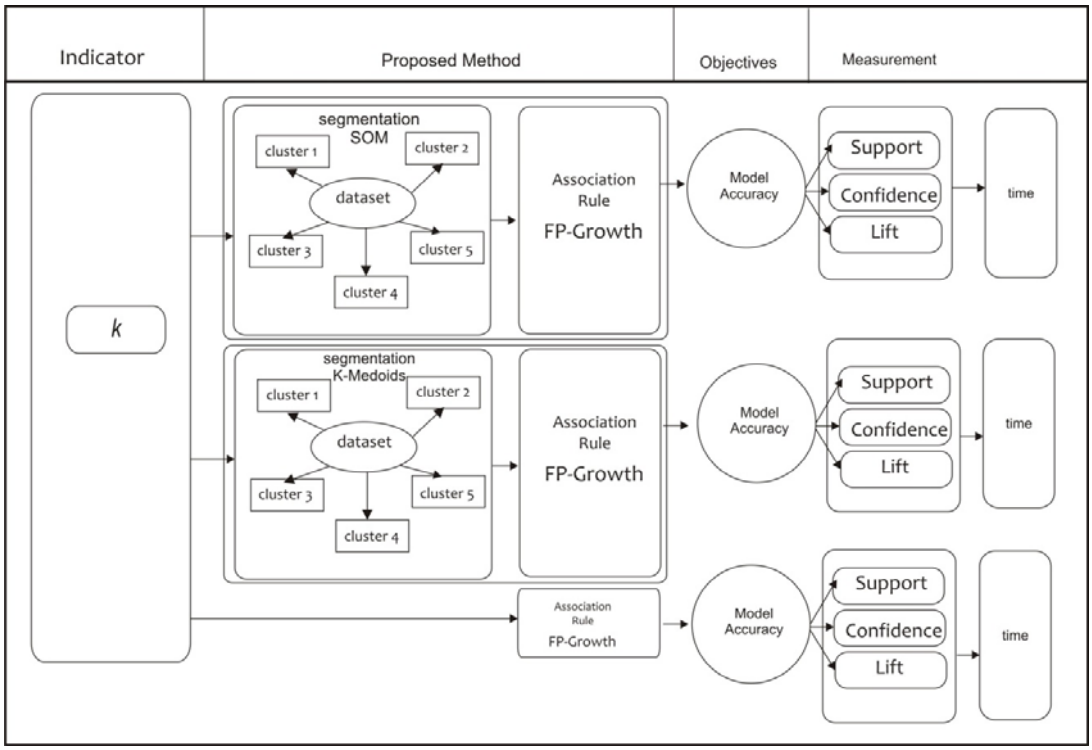
Data akan pada tahap pengolahan data awal diubah menjadi dataset yang sesuai dengan karakteristik algoritma yang akan dilakukan ujicoba. Diantaranya, menyesuaikan format *true* dan *false* dalam dataset menjadi data numerik 1 untuk *true* dan 0 untuk *false,* agar dapat digunakan dalam proses asosiasi. Tahapan selanjutnya melakukan *clustering* data tersebut tanpa menambah atribut apapun karena *Self Organizing Map* (SOM) akan meng*cluster* tersendiri. Sedangkan untuk *K-Medoids* ditentukan $k = 5$

	Basket ID	Hair Cond	Lemons	Standard	Frozen Ch	98pct. Fat	Sugar Coc	Onions	Deli Ham	Dishwash	Beets
2	C11867	0	0	0	0	0	0	0	0	0	0
3	C5096	0	0	0	0	0	0	0	0	0	0
4	C4295	0	0	0	0	0	0	0	0	0	0
5	C2837	0	0	0	0	0	0	0	0	0	0
6	C2693	0	0	0	0	0	0	0	0	0	0
7	C3497	0	0	0	0	0	0	0	0	0	0
8	C2696	0	0	0	0	0	0	0	0	0	0
9	C1895	0	0	0	0	1	0	0	0	0	0
10	C9524	0	0	0	0	0	0	0	0	0	0
11	C9527	0	0	0	0	0	0	0	0	0	0
12	C8726	0	0	0	0	0	0	0	0	0	0
13	C9383	0	0	0	0	0	0	0	0	0	0
14	C8582	0	0	0	0	0	1	1	0	0	0
15	C10133	0	0	0	0	1	0	1	0	0	0
16	C6983	0	0	0	0	0	0	0	0	0	0
17	C3641	1	0	0	0	0	0	0	0	0	0
18	C5246	0	0	0	0	0	0	0	0	0	0
19	C10136	0	0	0	0	0	0	0	0	0	0

Gambar 3. Dataset Setelah Diubah Menjadi Numerik

2.2 Desain Eksperimen

Eksperimen pada penelitian ini adalah menggunakan *Self Organizing Map* (SOM) dan *Frequent Pattern-Growth* (*FP-Growth*), kemudian dengan data yang sama menerapkan algoritma *K-Medoids* dan *Frequent Pattern-Growth* (*FP-Growth*) serta mengujinya juga dataset langsung diasosiasi dengan *Frequent Pattern-Growth* (*FP-Growth*) tanpa diawali dengan *clustering* terlebih dahulu. Desain untuk eksperimen ini tunjukkan pada gambar 4.



Gambar 4. Desain Eksperimen Penggabungan Algoritma

Eksperimen ini diawali dengan menerapkan algoritma *Self Organizing Map* (SOM) untuk *clustering* dan algoritma *Frequent Pattern-Growth* (*FP-Growth*) untuk pendekatan asosiasi. Dalam Algoritma *Self Organizing Map* (SOM) dataset di*cluster* secara alamiah tanpa ditentukan dulu jumlahnya. Proses

clustering dilakukan supaya data yang diasosiasi menjadi lebih kecil sehingga pola yang diperoleh menjadi lebih akurat. Tiap *cluster* yang terbentuk kemudian diasosiasi dengan *Frequent Pattern-Growth* (*FP-Growth*) untuk membuat pola belanja pelanggan. Dalam proses asosiasi ini akan diukur *support, confidence*, dan *lift ratio.* Setelah itu, dicatat waktu total yang dibutuhkan dari masing-masing tahapan. Eksperimen diulangi dengan mengganti algoritma *clustering Self Organizing Map* (SOM) dengan *K-Medoids* dan proses asosiasi tetap menggunakan *Self Organizing Map* (SOM). Jumlah *cluster* yang dibuat yakni 5 c*luster*. Kemudian masing-masing *cluster* di asosiasi dengan algoritma *Self Organizing Map* (SOM) untuk diukur *support, confidence*, dan *lift ratio.* Kemudian waktu proses masing-masing tahapan dicatat. Akurasi dan total waktu proses *clustering* hingga asosiasi dari tahapan 1 yakni *Self Organizing Map* (SOM) + *Frequent Pattern-Growth* (*FP-Growth*) dibandingkan dengan tahapan 2 yakni *K-Medoids* + *Frequent Pattern-Growth* (*FP-Growth*) kemudian dibandingkan dengan akurasi dan waktu yang diperlukan jika menggunakan *Frequent Pattern-Growth* (*FP-Growth*) saja. Kemudian disimpulkan diantara ketiganya terkait akurasi dan efisiensi waktunya.

2.3 Evaluasi Penelitian

Hasil akurasi untuk menentukan rekomendasi produk kepada pelanggan yang didapat dari penelitian ini dapat diukur dengan menggunakan *Lift Ratio*. *Lift Ratio* merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar produk A dibeli bersamaan dengan produk B. Sebuah transaksi dikatakan valid jika mempunyai nilai *Lift Ratio* lebih dari 1, yang berarti bahwa dalam transaksi tersebut, produk A dan B benar-benar dibeli secara bersamaan. *Lift Ratio* mengukur seberapa penting *rule* yang telah terbentuk berdasarkan nilai *support* dan *confidence*. Evaluasi rekomendasi produk untuk pelanggan ditunjukkan di gambar 5.

No.	Premises	Conclusion	Support	Confidence	Lift
1	Whole Corn	Hair Conditioner	0.970	0.976	1.006
2	Orange Flavored Fruit Bars	Hair Conditioner	0.970	0.976	1.006
3	Whole Corn	Orange Flavored Fruit Bars, Hair Conditioner	0.970	0.976	1.006
4	Orange Flavored Fruit Bars	Whole Corn, Hair Conditioner	0.970	0.976	1.006
5	Whole Corn, Orange Flavored Fruit Bars	Hair Conditioner	0.970	0.976	1.006
6	Whole Corn	Orange Flavored Fruit Bars	0.994	1	1.006
7	Orange Flavored Fruit Bars	Whole Corn	0.994	1	1.006
8	Hair Conditioner	Whole Corn	0.970	1	1.006
9	Hair Conditioner	Orange Flavored Fruit Bars	0.970	1	1.006
10	Hair Conditioner	Whole Corn, Orange Flavored Fruit Bars	0.970	1	1.006
11	Whole Corn, Hair Conditioner	Orange Flavored Fruit Bars	0.970	1	1.006
12	Orange Flavored Fruit Bars, Hair Conditioner	Whole Corn	0.970	1	1.006

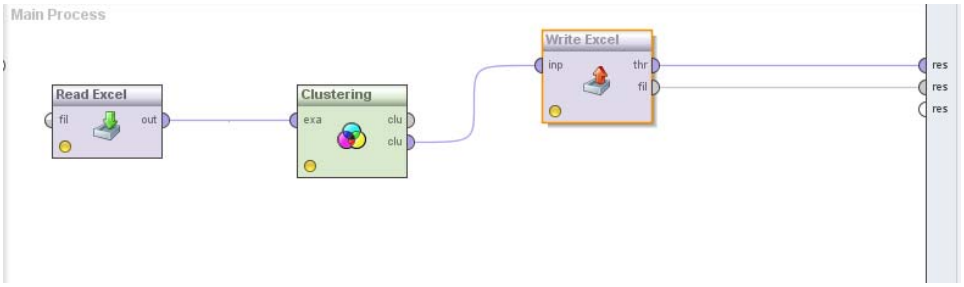
Gambar 5. Evaluasi Rekomendasi Produk Dengan *Support, Confidence, Lift*

Setelah ditemukan *Support, Confidence, Lift* kemudian dicatat pada masing-masing proses untuk diketahui seberapa efisien waktu yang diperlukan dari pengolahan dataset hingga menjadi sebuah pola belanja pelanggan. Kemudian akan disimpulkan tahapan mana yang memiliki tingkat kevalidan paling baik dan tahapan mana yang memiliki waktu proses paling efisien.

3. HASIL DAN PEMBAHASAN

3.1 Clustering Data dengan K-Medoids

Dalam *k-medoids* kita tentukan menjadi *5 cluster* (k). Penentuan $k = 5$ ini dilakukan supaya lebih mudah didapatkan hasilnya karena keterbatasan kemampuan PC. Hasil dari proses *clustering* tersebut akan di eksport dalam bentuk file terpisah untuk masing-masing hasil *cluster*. Proses *clustering* data ditunjukkan pada gambar 6, dan hasil *clustering* data ditunjukkan pada gambar 7.



Gambar 6. Proses *Clustering* Data

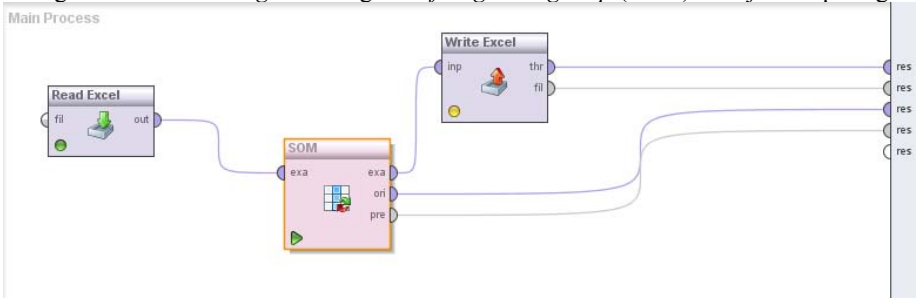
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Basket ID	Cluster	ir Conditio	Lemons	Standard	Frozen Ch	98pct. Fat	Sugar Coc	Onions	Deli Ham	Dishwash	Beets	40 Watt Li	Ice Cream	Cottage C	Plain Engl	Strawberi	Vanilla Ic
2	Basket ID	cluster_0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
3	C5252	cluster_0	1	1	0	0	1	0	1	0	1	0	0	1	1	0	0	0
4	C3659	cluster_0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
5	C4478	cluster_0	1	1	0	0	1	0	1	0	0	1	0	0	1	0	0	0
6	C887	cluster_0	0	0	0	1	0	1	1	0	1	0	0	0	1	0	0	0
7	C7995	cluster_0	0	0	1	0	0	0	0	1	1	0	1	0	0	1	1	1
8	C4694	cluster_0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
9	C8987	cluster_0	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0
10	C2177	cluster_0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0
11	C2543	cluster_0	1	1	0	1	0	1	1	0	0	0	1	1	0	0	0	0
12	C9233	cluster_0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
13	C7493	cluster_0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
14	C7646	cluster_0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
15	C1784	cluster_0	0	0	1	0	1	1	0	0	0	1	0	0	1	0	0	0
16	C5303	cluster_0	1	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0
17	C9623	cluster_0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
18	C959	cluster_0	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0	0
19	C5501	cluster_0	0	1	1	0	0	1	0	0	1	0	0	0	1	0	0	0
20	C4706	cluster_0	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0
21	C12002	cluster_0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
22	C3770	cluster_0	1	1	0	0	1	1	1	0	0	0	0	0	0	0	1	0
23	C3788	cluster_0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
24	C6347	cluster_0	1	1	0	0	1	1	1	0	0	1	0	1	0	0	0	0

Gambar 7. Hasil *Clustering* Data

Setelah hasil *cluster* didapat, yakni *cluster* 0, *cluster* 1, *cluster* 2, *cluster* 3, *cluster* 4 kemudian diurutkan/ sorting dan dipisahkan menjadi 5 file untuk kemudian di lakukan proses asosiasi dengan *Frequent Pattern-Growth* (*FP-Growth*). Hasil dari c*luster* menggunakan *K-Medoids*:
Cluster 0: 32 items
Cluster 1: 2 items
Cluster 2: 2 items
Cluster 3: 2 items
Cluster 4: 1323 items
Total number of items: 1361
Selama 28,20menit

3.2 Clustering Data dengan Self Organizing Map (SOM)

Rancangan Proses *clustering* data dengan *Self Organizing Map* (SOM) ditunjukkan pada gambar 8.



Gambar 8. Rancangan *Cluster* Data dengan *Self Organizing Map* (SOM)

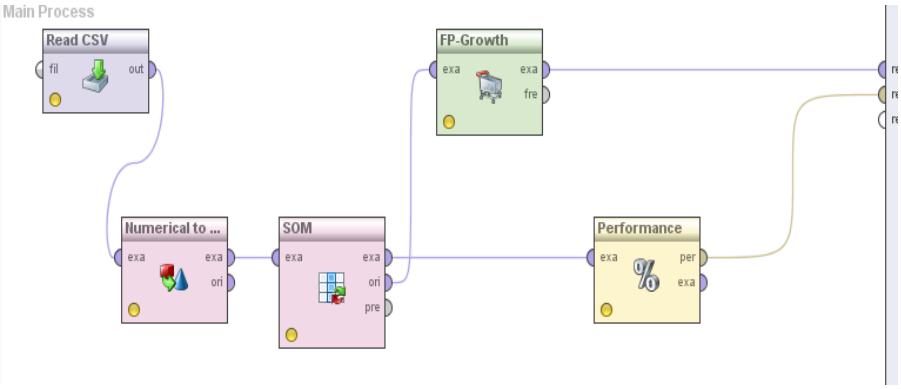
Cluster data dilakukan dengan *Self Organizing Map* (SOM) terbagi menjadi 2, dari rancangan *cluster* data tersebut didapatkan hasil yang ditunjukkan pada gambar 9.

Row No.	SOM_0	SOM_1
1	10	24
2	10	24
3	10	24
4	10	24
5	10	24
6	10	24
7	10	24
8	10	24
9	10	24
10	10	24
11	10	24
12	10	24
13	10	24
14	10	24
15	10	24

Gambar 9. Rancangan *Cluster* Data dengan *Self Organizing Map* (SOM)

3.3 Asosiasi Cluster Data Penjualan

Setiap *cluster* akan dilakukan asosiasi dengan *Frequent Pattern-Growth* (*FP-Growth*) untuk menghasilkan rekomendasi produk. Proses asosiasi *cluster* ditunjukkan pada gambar 10.



Gambar 10. Hasil Asosiasi *Cluster* Data Penjualan

Setelah proses *clustering* dilakukan, didapatkan 5 *cluster*. *Cluster* yang terbentuk dijadikan sebagai dataset baru yang lebih kecil daripada dataset sebelumnya. Kemudian setiap *cluster* akan dilakukan pendekatan asosiasi menggunakan algoritma *Frequent Pattern-Growth* (*FP-Growth*) untuk menganalisis keranjang pasar. Dengan dataset yang lebih kecil tujuan untuk meningkatkan akurasi dalam rekomendasi produk, sedangkan untuk perbandingan digunakan algoritma apriori.

4. KÉSIMPULAN

Hasil akurasi untuk menentukan rekomendasi produk kepada pelanggan yang didapat dari penelitian ini dapat diukur dengan menggunakan *Lift Ratio*. *Lift Ratio* merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar produk A dibeli bersamaan dengan produk B. Sebuah transaksi dikatakan valid jika mempunyai nilai *Lift Ratio* lebih dari 1, yang berarti bahwa dalam transaksi tersebut, produk A dan B benar-benar dibeli secara bersamaan [6]. *Lift Ratio* mengukur seberapa penting *rule* yang telah terbentuk berdasarkan nilai *support* dan *confidence*. Untuk eksperimen ditentukan nilai *Minimum Support* adalah 10% - 100% dan nilai *Minimum Confidence* 10% - 100%.

DAFTAR PUSTAKA

[1] B. Hoanca, "Using Market Basket Analysis to Estimate Potential Revenue Increases for a Small University Bookstore," *Information Systems*, vol. 1822, 2011, pp. 1-11.
[2] C. Wang, W. Lee, and C. Pang, "Applying Fuzzy *FP-Growth* to Mine Fuzzy Association Rules," *Engineering and Technology*, 2010.
[3] C. Vercellis, *Business Intelligence: Data Mining and Optimization for Decision Making*, Milan: WILEY, 2009.
[4] Endy, Charles Lim, Kho I Eng, AntoSatriyoNugroho, 2010, *Implementation of intelligent searchingUsing self-organizing map for webmining used In document containing information in Relation to cyber terrorism,Second International Conference on Advances in Computing, Control, and Telecommunication Technologies*.
[5] Juha Vesanto and Esa Alhoniemi,Clustering of the Self-Organizing Map, IEEE Transactions on *Neural Networks*, 2000.
[6] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2006.
[7] K. Christidis, D. Apostolou, and G. Mentzas, "Exploring Customer Preferences with Probabilistic Topics Models," Work, 2010, pp. 1-13.
[8] L. Gordon, "Leading Practices in Market Basket Analysis," *Strategy*, vol. 94022, 2008.
[9] M. Plasse, N. Niang, G. Saporta, A. Villeminot, and L. Leblond, "Combined use of association rules mining and clustering methods to find relevant links between binary rare attributes in a large data set," Computational Statistics & Data Analysis.
[10] M. Ykhlef, "A Quantum Swarm Evolutionary Algorithm for mining association rules in large databases," *Journal of King Saud University - Computer and Information Sciences*, vol. 23, 2011, pp. 1-6.
[11] S. Guglani, G. Gupta, and M. Dhanda, "Way to Improve K-Means Algorithm by Using Various Attributes," (IJAEST) International Journal of Advanced Engineering Sciences and Technologies, vol. 7, 2011, pp. 330 - 336.
[12] Yongmei Liu and Yong Guan, "*FP-Growth* Algoritm For Appliaction in Research of Market Basket Analysis,"2008, Capital Normal University, Beijing, China 2008.
[13] Y. Liu and Y. Guan, "*FP-Growth* Algorithm for Application in Research of Market BasketAnalysis," *Design*, 2008.
[14] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. Mclachlan, A. Ng.

View publication stats

Improvement and research of FP-growth algorithm based on distributed spark

DENG Lingling
College of Computer and Information, Hohai University
Nanjing, China
917644817@qq.com

LOU Yuansheng
College of Computer and Information, Hohai University
Nanjing, China
wise.lou@163.com

***Abstract*—FP-growth algorithm as the representatives of non-pruning algorithms is widely used in mining transaction datasets. But it is sensitive to the calculation and the scale of datasets. When building FP-tree, the search operation as the major time-consuming operation has a higher complexity. And when the horizontal or vertical dimension of data set is larger, the mining efficiency will be reduced or even failed. To solve the above problems, reducing the complexity of search time and applying distributed computing are the widely used strategies. This paper presents a distributed SPFP algorithm based on Spark framework and improved FP-growth algorithm. The results of tests show that, compared to the PFP algorithm based on MapReduce, the OPFP algorithm based on Spark and original FP-growth algorithm, SPFP has high efficiency, cluster and flexibility.**

Keywords—Frequent item sets; FP-growth; Spark; Complexity; Efficiency

I. INTRODUCTION

Association rule reflects the connections and dependence between things. The mining of association rule [1], trying to find some hidden or interesting association rules in the data set. It is one of the important research directions in the field of data mining. FP-growth algorithm [2] is a no-pruning algorithm. The structured FP-tree is decomposed into several conditional model sub-trees, which corresponds to a frequent itemsets for each and these trees can be mined recursively to get frequent itemsets of transaction databases. But the algorithm still has many defects: First, when the horizontal or vertical dimension of a transaction is very large, which leads to the failure of frequent itemsets mining; second, building conditions pattern frequently makes the space complexity too high to fail; thirdly, the time complexity of scanning head table or FP-tree node is too high to reduce the efficiency. The open source distributed computing framework Spark designed by UC Berkeley AMP lab can be applied to data mining and machine learning or other required iterative algorithms. This paper proposes a distributed SPFP algorithm based on Spark to solve these problems.

II. RELATED WORK

Many scholars proposed optimizations or improvements. Literatures [8, 9] mentioned to decompose original database to support mining. But it can't realize the parallelization, when the transaction chains are too large to be loaded in RAM. Literatures [11, 12] constructed schema matrix to mining frequent item sets. But it will be failed when the matrix can't fix in the memory. Literature [7] proposed to use MFP tree replace FP-tree to reduce scanning times. But it produced a large number of candidate items. Literatures [3, 8] proposed different ways based on MapReduce. Most were based on the thought of PFP. But PFP algorithm has a serious disadvantage that is operating transaction data between different places. The scalability of PFP algorithm is low.

In summary, in the environment of big data, these ways are still not good enough. This paper presents a SPFP algorithm. It can reduce time complexity by optimizing header table and data structure, supports fine-grained local data by Spark, improves the operational efficiency. The paper uses UCI and Frequent ItemSet Mining DataSet Repository datasets to test. Results show that, SPFP algorithm is more effective.

III. ANALYSIS AND IMPROVEMENT OF FP-GROWTH

When constructing FP-tree, insert a transaction each time should sort transactions according to the items in the head table. The time complexity is O (n^2). And the projects compare with sub-nodes in turn. For these limitations, it gives a new revised algorithm-IFP. Uses transactional database D in Table I

TABLE I TRANSACTIONAL DATABASE *D*

TID	Original Itemsets	Sorted Itemsets
T_1	f, a, c, d, g, i, m, p	*f, c, a, m, p*
T_2	a, b, c, f, l, m, o	*f, c, a, b, m*
T_3	b, f , h , j , o	*f, b*
T_4	b, c, k, s, p	*c, b, p*
T_5	a, f, c, e, l, p, m, n	*f, c, a, m, p*

The most right column of data has sorted, which is according to the frequency of items appearing in the entire database. The threshold is 3 (less than 3 will be excluded). FP-growth algorithm uses two data structures: Header Table and FP-tree. And the structure of Header Table affects the performance of FP-growth algorithm directly. This paper modifies the original data structure in Figure I (use the transaction database D in Table I as the sample data):

978-1-4673-8350-9/15 $31.00 © 2015 IEEE
DOI 10.1109/CCBD.2015.15

IEEE computer society

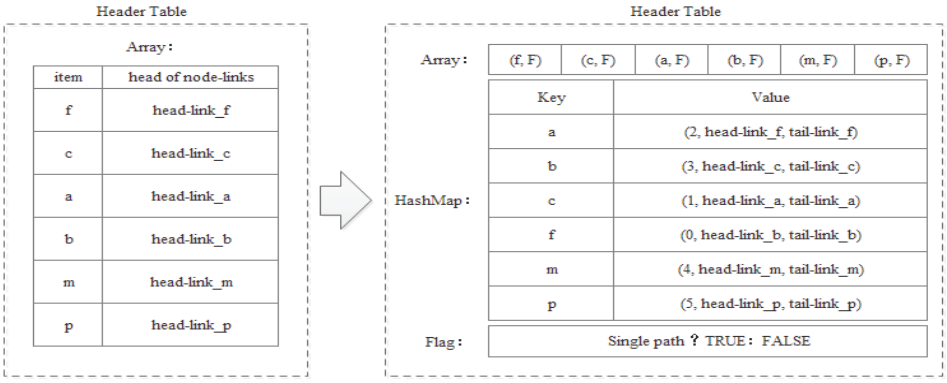


FIGURE I MODIFIED DATA STRUCTURE OF HEADER TABLE

The reflected improvements show in the following aspects:

1) Change the original array into the one Array label shows. Alter the second element of original into Boolean type.

2) Add an element of HashMap type. Key is project name. Value is a triple: first element is subscript of project name; the second is the address of head pointer of node chain, the third is of tail pointer.

3) Add an indicator variable as Flag tag shows. It is used to identify whether the corresponding FP-tree is a single path.

In the space complexity, assuming the unit storage is 1, the number of projects is m. The capacity of the left figure occupies 2m, while the right is 6m+1, three times than before. But m is small usually, so the improvement is acceptable.

In the time complexity, assume that a transaction contains k elements, according to the left figure, the sorting times are $k \cdot m$. In the worst case, the searching time is m, the comparing time is $m \cdot m$, and the total complexity time is $O((k+m) \cdot m) = O(m^2)$. In the right figure, find the index of project when sorting and make the Boolean variable True. The time complexity is $O(k)$. In worst case, the time complexity is $O(m)$ insert it into the FP-tree. Traversal Value data to determine whether the head and tag of the chain are coincident. The time complexity is $O(m)$. Totally, the time complexity is $O(k)+O(m)+O(m)=O(m)$.

Similarly, this paper modifies the structure of FP-tree in Figure II(use transaction database D in Table I as example):

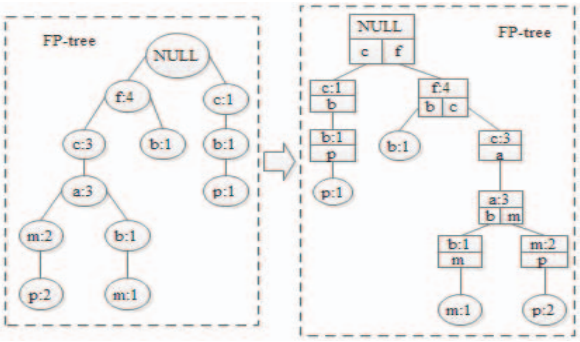


FIGURE II MODIFIED DATA STRUCTURE OF FP-TREE

Before, the time complexity of searching the child node list is $O(m)$. Now, the paper modifies every child nodes to a HashMap type. The time complexity becomes $O(1)$.

The time complexity reduced $O(m^3)$ to $O(m)$, so the improved structure can advance the efficiency.

Recursion needs to complete switching context, so reducing the recursion can improve the efficiency. In the FP-growth algorithm, if the flag "flag" is true in figure I, this FP-tree will be a single path tree and can get frequent items through permutations and combinations without recursive operations.

From the improvements of FP-growth algorithm above, the improvements of data structure reduce the time complexity and the judgments of a single path reduce the cost of space-time, which results from recursive operation. For the following description, this paper named the improved algorithm as IPF.

IV. RESEARCH OF SPFP ALGORITHM

When compressing the whole date base into FP-tree can cause memory out. So put IPF algorithm run in a distributed environment. Operate the distributed data fragmentations and combine partial frequent item sets to do the global pre-pruning. It improves the efficiency of data analysis and solves the large amounts of data efficiently that single computer can't handle.

This paper uses "Move- algorithm" to parallelize the IPF algorithm--SPFP. To improve the performance, make IPF algorithm run in the data node and divide the operation of calculating frequent items into the following two phases:

1) Produce local frequent item sets. Run IFP algorithm in m nodes in a distributed cluster, and obtain local frequent item sets for each node.

2) Build the global candidate sets. According to Spark operator to collect and union all local frequent item sets as a candidate set, then distribute it into each node. Rebuild FP-tree based on node-data blocks, statistic the occurred times of each candidate. Based on times the set name merges and the minimum support min_sup to get global frequent item sets.

The first step of SPFP is to call IPF algorithm on each data; the second is to calculate the frequency of candidates on each

fragmentation. The paper builds FP-tree on each data node, and statistics the local frequency of item sets. The next describes the implementations of the frequency statistics that the major candidate frequent item sets appear in the local fragmentation.

Take the FP-tree in transaction database *D* of table I as example, assume that the collection of candidate sets have a candidate set, if find out all paths containing a = {c, p}, the effective paths are the marked part in red in Figure III**:**

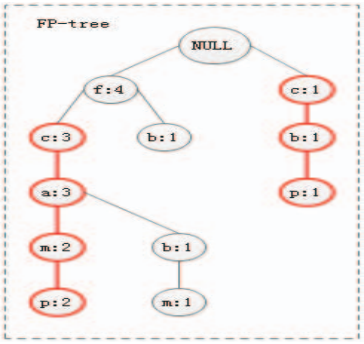


FIGURE III EFFECTIVE QUERY PATH

Usually use traversal scheme from top to down. But it needs to coverage all nodes to determine the appearing times of α on the date block. Efficiency is lower. But according to the characteristics of FP-tree, each node has the only one parent node unless the root node. If use bottom-up approach, just make sure the smallest frequency project is p in α. According to Header Table and p to locate all the nodes named p, and then do up search.

Take the left marked path in bold in Figure III as an example, the process is shown as the following in Figure IV:

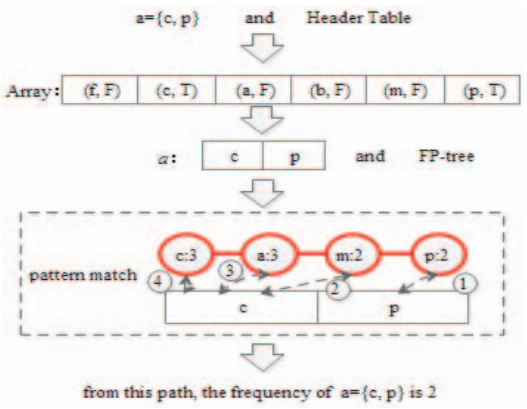


FIGURE IV JUDGE THE FREQUENT ITEM SETS

The process of pattern matching can be converted to the longest common subsequence problem. So the time complexity of matching a candidate set and a path is $O(|\alpha|+|path|)$. $|\alpha|$ is the length of candidate set, $|path|$ is the number of all the nodes, the time complexity is linear.

Use pseudo-code to describe process of SPFP as following:

Input: filePath of HDFS of D, minimum support "min_sup"
Output: global frequent itemsets "L";

(1) *val sc* = Initialize *SparkContext* **;**
(2) *val hadoopRDD* = *sc*.textFile(*filePath*) **{**
(3) *val FIRDD* = *hadoopRDD*.mapPartition(
(4) *p* => {
(5) FP-growth(*p*) }) **; /* Use FP-growth algorithm to deal with the data block p and get locally frequent item sets */**
(6) *val bro_FIRDD* = *sc*.broadcast(*FIRDD to Local*) **; } /*Localize FIRDD, eliminate duplicate elements, broadcast to each node */**
(7) *val support_item* = *hadoopRDD*.mapPartition(
(8) *p* => {
(9) get*Frequet*(*p*,*bro_FIRDD*)) **/* Construct FP-tree to get frequentness of item sets */**
(10) }) **; /* Complete the processing of fine-grained of local data block */**
(11) *val L* = getFilter(*support_item*,*min_sup*) **/* Combined the data and according to minimum support min_sup to filter the inappropriate sets to get the final data L */**
(12) return *L* **; /* Get all global frequent item sets */**

V. THE RESULTS AND EVALUATION

To verify the effectiveness, this paper uses five computers (Intel Core i5, CPU 2.4GHz, memory 2GB) consist distributed cluster of Spark environment. Parameters: Ubuntu14.04, Hadoop2.4.1, Spark1.2.0, Scala 2.10.4, JDK 7u55. One computer is the Master node, the others are Salve node. The test datasets are from the UCI a public database, or Frequent ItemSet Mining DataSet Reposito-ry [4].

Mushroom [5]. It describes the mushroom whether can be used or has toxic dataset through different attributes. The data set has 8124 data instances and 119 project values totally.

Accidents. It is the traffic accident statistics among 1991 to 2000 from the Flemish region of Belgium National Bureau of Statistics. It has 340,184 instances containing 572 attributes, per transaction contains 45 property values.

Average many values to eliminate errors to obtain the final results. These datasets use PFP, OPFP and SPFP. PFP is a parallelized FP-growth algorithm based on MapReduce while OPFP based on Spark. The results are shown in the following Figure V and Figure VI:

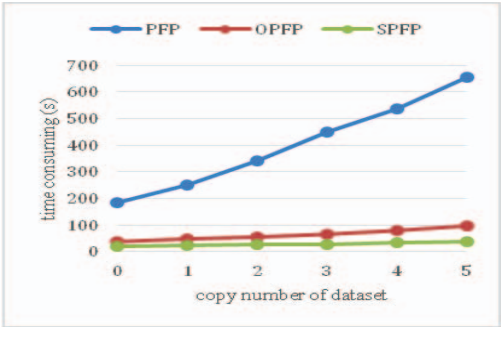


FIGURE V MUSHROOM: MIN_SUP=35%

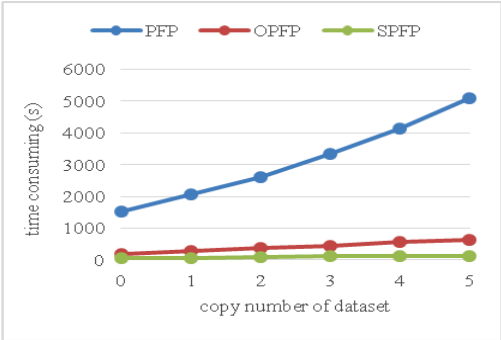


FIGURE VI ACCIDENTS: MIN_SUP=20%

Through analysis, execution time and increasing copy number rise in positive correlation. PFP is the most obvious. Because of the high latency, lack of support of the iterative nature and other shortcomings of MapReduce model. Compared to SPFP and OPFP, the execution efficiency of PFP is lowest, almost 20 to 50 times longer than SPFP. SPFP is several times higher than OPFP with improved data structure.

To perform the scalability, this paper has the following experiment: maintain the support of dataset, the cluster node scale grows from 1 to 5, one is Master node, and the others are Salve nodes. Set the variable range of copy number from 0 to 5, the results are shown in Figure VII and Figure VIII

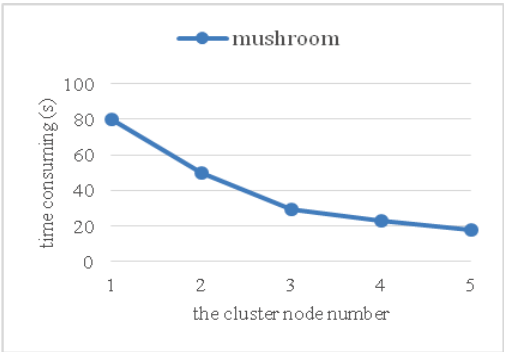


FIGURE VII MUSHROOM: MIN_SUP=35%

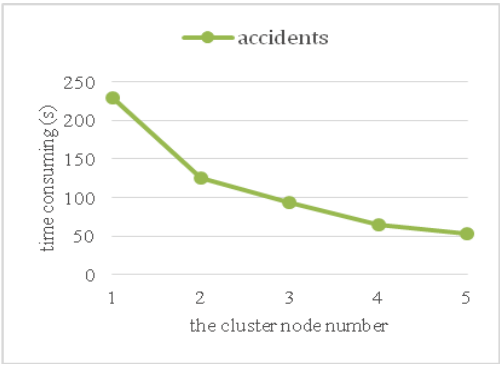


FIGURE VIII ACCIDENTS: MIN_SUP=20%

Through analysis, the running time declines while the nodes increase. It has a large decreasing amount when the node is less. With the nodes increasing, it almost decreases linearly and tends to be horizontal.

To sum up, the SPFP algorithm in this paper has higher efficiency than other algorithms and also has the very well scalability of data and cluster.

VI. SUMMARY AND FUTURE WORK

This paper analyzes the shortages of FP-growth algorithm, the traditional association rule mining. According to these shortages, the paper proposes the optimized schemes from the realization of the head table and FP-tree data structure and other angles. Submit a distributed algorithm in Spark environment—SPFP. Finally, it does experiment from date and cluster flexibility. The results prove that SPFP algorithm is an efficient frequent item set algorithm.

Based on the work, the next step will to consider how to apply the implemented algorithm to the practical problems and try to analyze the reality stock or basket data to find out the meaningful rules or relationships.

REFERENCES

[1] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases", AI magazine, vol. 17, no. 3, p. 37, 1996

[2] Jiawei Han, Jian Pei, Yiwen Yin. Mining frequent patterns without candidate generation. Proceedings of the 2000 ACM SIGMOD International conference on Management of Data SIGMOD'00. Dallas, Texas, United States: ACM Press, 200, 29(2): 1-12P

[3] Zhiyong Zeng, Chengzhi Yang, Ye Tao. Research of load balance FP-growth algorithm in parallel. Computer Engineering and Applications, 2010, 46(4): 125-126

[4] Frequent Itemset Mining Dataset Repository[EB/OL].[2012-10-21]. http://fimi.ua.ac.be/data/, 2012

[5] C.L. Blake and C.J. Merz. UCI Repository of Machine Learning Databases. Dept. of Information and Computer Science, University of California at Irvine, CA, USA. 1998. http://www.ics.uci.edu/mlearn/MLRepository.html

[6] Yong Yang, Wei Yang. A parallel FP-growth algorithm based on Mapreduce[J]. Journal of Chongqing University of Posts and Telecommunications(Natural Science Edition), 2013, 25(05): 651-657

[7] Jun Gao, Shoucai He, A Fast Association Rule Mining Algorithm [J]. Computer Engineering, 2006, 01:116-118

[8] Haiyan Yang. Improved FP-growth algorithm and its Application in mining association dishes of fast-food chain insudtry[D]. Hefei University of Technology, 2013

[9] Cuiming Yang, Xiping Liu. A New Algorithm for Mining Association Rules Based on Dataset Division[J]. Journal of Natural Science of Hunan Normal University, 2007, 02: 30-34

[10] Yun Yang, Yanxia Luo, Improved algorithm based on FP-Growth [J]. Computer Engineering and Design, 2010, 07:1506-1509

[11] Fengyi Deng, Zhenyu Liu. Improved FP-Growth algorithm based on pattern matrix[J]. Journal of Xiamen University (Nat ural Science), 2005, 05:629-633.

[12] Xingshu Chen, Shuai Zhang, Hao Tong, Xiaojing Cui. FP-growth algorithm based on boolean matrix and MapReduce[J]. Journal of South China University of Technology(Natural Science Edition) , 2014,01:135-141

[13] Jianfeng Du. Research and Implement of Distributed Mining Association Rules with Constraints[D]. Central South University, 2005

A Comparative Study of FP-growth Variations

Aiman Moyaid Said[A], Dr. P D D. Dominic[B], Dr. Azween B Abdullah[C]

Department of Computer and Information Sciences

Universiti Teknologi PETRONAS

Summary
Finding frequent itemsets in databases is crucial in data mining for purpose of extracting association rules. Many algorithms were developed to find those frequent itemsets. This paper presents a summarization and a comparative study of the available FP-growth algorithm variations produced for mining frequent itemsets showing their capabilities and efficiency in terms of time and memory consumption.

Keywords: Data mining, frequent itemsets, FP-growth.

1. INTRODUCTION

An association rule is defined as the relation between the itemsets, since its introduction in 1993 [1] the process of finding the association rules has received a great deal of attention. Today the extracting of association rules is still one of the main popular pattern discovery techniques in knowledge discovery and data mining (KDD).

The process of extracting the association rules can be viewed as two-phases: the first phase is to mine all frequent patterns; each of these patterns will happen at least as frequently as preset minimum support count (min_sup). The second phase is to produce strong association rules from the frequent patterns; these rules must assure minimum support and minimum confidence. The performance of discovering association rules is largely determined by the first phase, [8].

A lot of algorithms were proposed to optimize the performance of the FP-growth algorithm. In this paper we mainly restrict ourselves to study the performance of the FP-growth's Variations in term of running time and the memory usage.

2. RELATED WORKS

In fact, a broad variety of efficient algorithms for mining frequent itemsets have been developed. Agrawal et al in [1], introduced Apriori algorithm to find the frequent itemsets from market basket dataset. The Apriori algorithm adopts candidates' generations-and-testing methodology to produce the frequent itemsets. In the case of the long itemsets the Apriori approach suffer from the lack of the scalability, due to the exponential increasing of the algorithm's complexity.

FP-growth approach for mining frequent itemsets without candidate generation was proposed by Han in [2] . Its scalable frequent patterns mining method has been proposed as an alternative to the Apriori-based approach. The pattern growth approach adopts the divide-and-conquer methodology to produce the frequent itemsets.

This algorithm creates a compact tree-structure, FP-Tree, representing frequent patterns, which moderates the multi-scan problem and improves the candidate itemset generation. This algorithm is faster than others in the literature, this reported by the authors of this algorithm.

Several algorithms implicate the methodology of the FP-growth algorithm. In [3] Pei used the same approach for Mining closed frequent itemsets and max-patterns. Likewise, Pei suggested to Mining sequential patterns in [4].

Further Improvements of FP-growth Mining Methods were introduced. [5],[6],[7], adapted the similar approach of Han et al [2] for mining the frequent itemsets from the transactional database. The authors reported that these algorithms are more efficient than FP-growth.

Manuscript received May 5, 2009
Manuscript revised May 20, 2009

3. FP-GROWTH ALGORITHM REVISIT

FP-growth algorithm is an efficient method of mining all frequent itemsets without candidate's generation. The algorithm mine the frequent itemsets by using a divide-and-conquer strategy as follows: FP-growth first compresses the database representing frequent itemset into a frequent-pattern tree, or FP-tree, which retains the itemset association information as well.

The next step is to divide a compressed database into set of conditional databases (a special kind of projected database), each associated with one frequent item. Finally, mine each such database separately.

Particularly, the construction of FP-tree and the mining of FP-tree are the main steps in FP-growth algorithm.

For the explanation of the algorithm, we will use the following example. To find the frequent itemsets from transactional database DB (see Table 1).First, a scan of the database DB derivers a set of frequent 1-itemsets (L) which also include their support count. The set L is sorted in the order of descending support count, this ordering is important since each path of FP-tree will follow it.

Let the minimum support count be 3,then the set L={(f,4),(c,4),(a,3),(b,3),(m,3),(p,3)}.

Table 1: The transactional database DB

TID	Items
T1	f,a,c,d,g,i,m,p
T2	a,b,c,f,l,m,o
T3	b,f,h,j,o
T4	b,c,k,s,p
T5	a,f,c,e,I,p,m,n

Second, an FP-tree is constructed as follows: The root of the tree, labeled Null, is created. The database DB is scanned for the second time. The items in each transaction are processed in L order, and a branch is created for each transaction.

For example, the scan of the first transaction, "T1:f,a,c,d,g,I,m,p" which contains five items (f,c,a,m,p in L order).Only those items that are in the list of frequent itemsets L ,leads to constructions of the first branch of the tree with tree nodes {<f,1>,<c,1>,<a,1>,<m,1>,<p,1>} where <f,1> is linked as a child of the root. <c,1> is linked to <f,1>,<a,1> is linked to <c,1> ,<m,1> is linked to <a,1>, and <p,1> is linked to <m,1>.

The second transaction, because it shares items f, c and a, it shares the common prefix {f,c,a} with the previous branch and extends to the new branch {<f,2>,<c,2> , <a,2>,<m,1>, <p,1>} .increasing the count of the common prefix by 1.The new intermediate version of FP-tree, after adding two transactions from the database ,is given in Fig. 1.for the remaining transactions can be inserted in the same way (see Fig. 2).

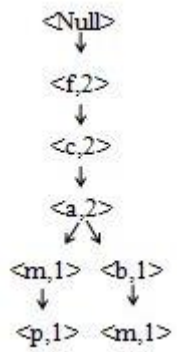


Fig.1 FP-tree for two transactions

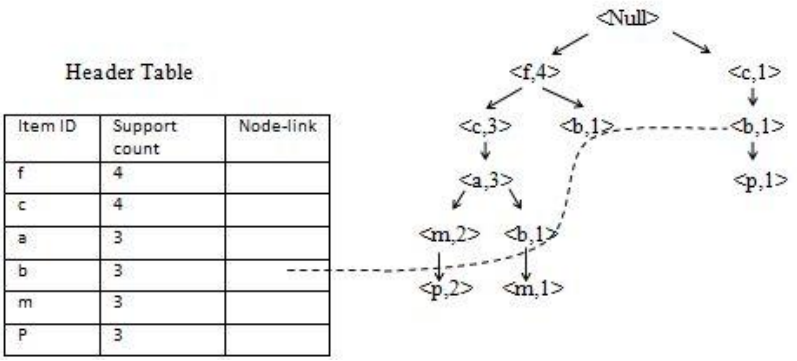


Fig. 2 Final FP-tree

To ease tree traversal, header table is built so that each item points to its occurrences in the tree via chain of node-link.

Using the compact tree structure (or FP-tree), the FP-growth algorithm mines all the frequent itemsets. The FP-tree is mined as follows. Begin from each frequent-1 pattern (as an initial suffix pattern), construct its conditional pattern base (a "subdatabase" which consists of the set of prefix paths in the FP-tree co-occurring with suffix pattern), then build its (conditional) FP-tree, and do mining recursively on such a tree. The patterns growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree.

In our example, according to L ,the complete set of frequent itemsets can be divided into subsets (6 for our example) without overlapping, first, frequent itemsets having items p (as an initial suffix pattern) ,which is the last item in L, rather than the first item. The reason for starting at the end of the list will become clear as we explain the FP-tree mining process. Second, the itemsets having item m but not p; third, the itemsets that have item b without both m and p; we continue this process to the end. Therefore, the last set will be the large itemsets only with f.

The item p occurs in two branches of the FP-tree of Fig.2.The occurrences of p can easily found by starting from the header table of p and following p's node-links. The paths formed by these branches are {<f,4>,<c,3>,<a,3>,<m,2>,<p,2>}and{<c,1>,<b,1>,<p,1} where samples with a frequent item p are {<f,2>,<c,2>,<a,2>,<m,2>,<p,2>}and{<c,1>,<b,1>,<p,1>}, which form its conditional pattern base ,these samples are the transactions that contain the branch of the tree with the existing of item p. Its conditional FP-tree contains only {<c, 3>} ,the other items are not included because its support count is less than 3.The generated frequent itemset that satisfy the minimum support count is {<c,3>,<p,3>},all the other itemsets are below the minimum support count.

The next subsets of frequent itemsets are those with m item and without p. The FP-tree recognizes the paths {<f,4>,<c,3>,<a,3>,<m,2>}and {<f,4>,<c,3>,<a,3>,<b,1> ,<m,1>},or the related accumulated samples {<f,2>,<c,2> ,<a,2>,<m,2>} and {<f,1>,<c,1>,<a,1>,<b,1>,<m,1>} . Analyzing the samples we find the frequent itemset {<f,3>,<c,3>,<a,3>,<m,3>}.

Similar to subset 3 to 6 the same process is done in our example, additional frequent itemsets can be mined.These are itemsets {f,c,a} and {f,c},but they are already subset of frequent itemsets {f,c,a,m}.Therefore ,the final set of frequent itemsets is {{c,p},{f,c,a,m}}.

4. FP-GROWTH VARIATIONS

Several optimization techniques are added to FP-growth algorithm. In this paper, we investigate the performance of three algorithms, namely AFOPT Algorithm, Nonordfp algorithm and Fpgrowth* algorithm .Our goal is not to go into many details about the algorithms but show the basic optimization ideas and the different of the performance in term of running time and memory consumption. In the following we will illustrate what are the main optimization ideas in each algorithm.

- **AFOPT ALGORITHIM**

Liu et al in [5] investigated the algorithmic performance space of the Fpgrowth algorithm. They specified the problem of conditional databases construction (particularly the number of the conditional databases constructed and the mining cost of each individual conditional database) in Fpgrowth algorithm, which have direct effect on the performance of the algorithm. They studied the problem of enhancing the Fpgrowth algorithm from four perspectives to come with the best strategy for mining the frequent itemsets. These perspectives are the item search order (in what order the search space is explored), conditional database representation, conditional database construction strategy and tree traversal strategy.

For the first part of the problem, the number of the conditional databases constructed can differ very much using different items search orders. The dynamic ascending order is able to minimize the number and /or the size of the conditional database constructed in subsequent mining, AFOPT algorithm adapt this kind of items search order which is also used by Fpgrowth.

For the second part of the problem, the mining cost of each individual conditional database is heavily depends on its representation (tree-based or array-based).AFOPT algorithm use adaptive representation ,tree-based structure in the case of dense dataset and array –based representation in the case of sparse dataset. In additions to the conditional database representation the size and the conditional database construction strategy have effect on the mining cost of each individual conditional database, two type of

the conditional database construction strategy (physical construction or pseudo-construction).
The dynamic ascending frequency search order can make the subsequent conditional databases shrink rapidly. As a result, it is useful to use the physical construction strategy with the dynamic ascending frequency order. The traversal cost of a tree us minimal using the top-down traversal strategy, AFOPT algorithm uses dynamic ascending frequency order for both the search space exploration and prefix-tree construction, it uses the top-down traversal strategy. as a summery AFOPT algorithm utilizes dynamic ascending frequency for the item search space ,adaptive representation for the conditional database format ,physical construction for the conditional database construction, and top-down traversal strategy for the tree traversal.

- **NONORDFP**

The running time and the space required for the Fpgrowth algorithm were the motivation for Nonordfp algorithm. Rácz in [9] dealt with the implementation issues, data structures, memory layout, I/O, and library functions.
A compact, memory efficient representation of an FP-tree by using Trie data structure, with memory layout that allows faster traversal was introduced, to deal with the running time and space requirement problem. This compact representation of FP-tree allows faster allocation, traversal, and optionally projection. It contains less administrative information about the items in the database (no labels for the items are stored in the node, no header lists and children are required), and allows more recursive steps to be carried out on the same data structure, with no need to rebuild it.

- **FPGROWTH* ALGORITHM**

Depending on a numerous experiments were done by Grahne et al [10], they found that 80% of the CPU time was used for traversing FP-trees. Consequently, they employed the array-based to reduce the traversal time of the FP-trees. Fpgrowth* algorithm uses FP-tree data structure in combination with the array-based and incorporates various optimization techniques.

In the case of sparse data set the array-based technique work very well, the array save traversal time for all items and the next level of FP-trees can be initialized directly. While in the case of dense data set, the FP-tree is more compact. To deal with this problem they proposed optimizing technique that help the algorithm to estimate if the data set is sparse or dense, by counting the number of the nodes in each level of the tree which done during the construction of each FP-tree. If the data set turns to be dense data set then no need to calculate the array for the next level of the FP-tree. In the case of sparse data set, the calculation of the array for the next FP-tree is required.

5. COMPARISON OF THE ALGORITHMS

To verify the efficiency of the FP-growth variation algorithms a lot of experiments were conducted. All the experiments are conducted on Core 2 Duo 2.00 GHZ CPU, 2.00 GB memory and hard disk 160 GB. The operating system is ubuntu 8.10.We test the AFOPT algorithm, Nonordfp Algorithm, Fpgrowth* algorithm [11] and the original Fpgrowth algorithm [12].To evaluate the behavior of the four algorithms different datasets and different support threshold were used, in the following subsections the type of the data sets ,the running time and the memory consumption are illustrates:

5.1 Datasets

The data is challenging due to the number of characteristics which are the number of the records, and the sparseness of the data (each records contains only small portion of items).

In our experiments we chose different dataset with different prosperities, to prove the efficiency of the algorithms, Table 2 shows the datasets and the characteristics.

Table 2: The Datasets

Data set	#Items	Avg. Length	#Trans	Type	Size
T10I4D100k	1000	10	100,000	Sparse	3.93 MB
T40I10D100K	1000	40	100,000	Sparse	14.8 MB
Mushroom	119	23	8,124	Dense	557 KB
Connect4	150	43	67557	Dense	8.89 MB

5.2 Running Time: The running time is real time, system time and user time. Figs 3, Fig 4, Fig 5, and Fig 6 depict the time needed in seconds for each one of the algorithms.

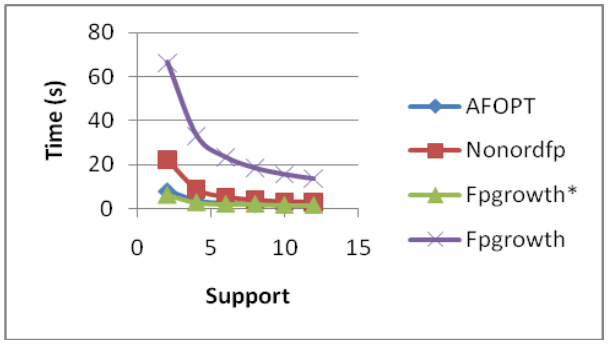


Fig 3 Execution time at various support levels on T10I4D100k

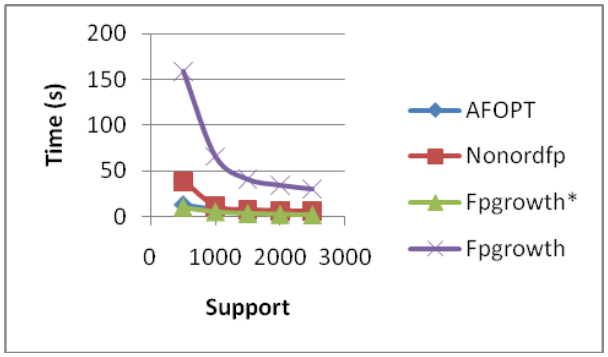


Fig 4 Execution time at various support levels on T40I10D100K

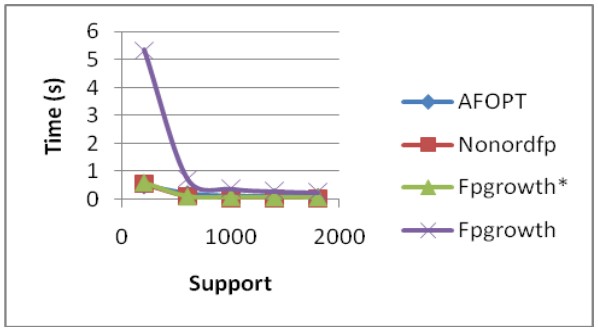


Fig 5 Execution time at various support levels on Mushroom

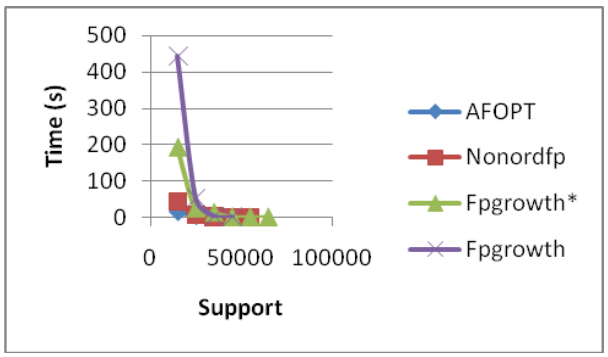


Fig 6 Execution time at various support levels on Connect4

It is clear that with the T10I4D100k data set Fpgrowth* algorithm outperforms all the other algorithms. On T40I10D100K data set there is obvious performance competition among both Fpgrowth* algorithm and AFOPT algorithm. The running times for the AFOPT algorithm, Nonordfp algorithm, and Fpgrowth* algorithm are near in the case of mushroom data set. For the connect4 data set, we should mention that some algorithms had problem, segmentation fault, with some values of support due to the huge number of the frequent itemsets satisfy those thresholds values and some took long time to find the frequent itemsets.

5.3 Memory Consumption: In this section, we calculate the total number of memory consumption for each algorithm .All the experiments are done on the same sets of data. As shown in Fig 7,Fig 8,Fig 9,and Fig 10 the support values and the amount of memory for each one. We observe that, Nonordfp algorithm remains stable over the whole range of support values on T10I4D100k.The stability in memory consumption is also observe for Fpgrowth* algorithm and AFOPT algorithm for the high values of support.

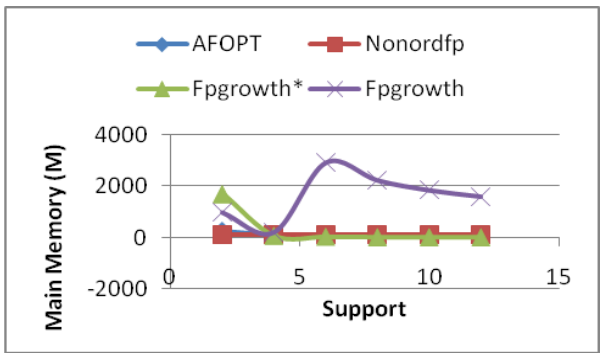


Fig 7 Memory usage on T10I4D100k

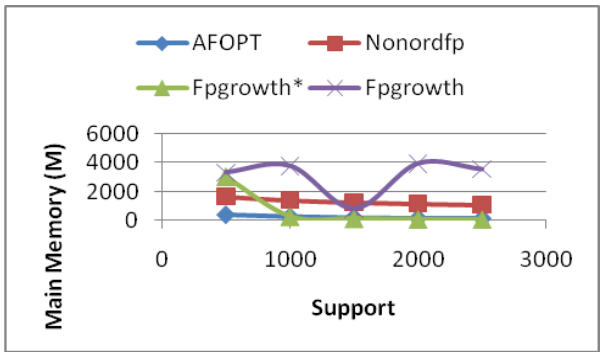


Fig 8 Memory usage on T40I10D100K

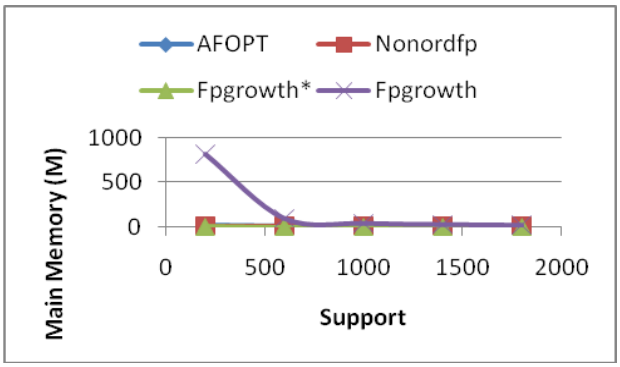


Fig 9 Memory usage on Mushroom

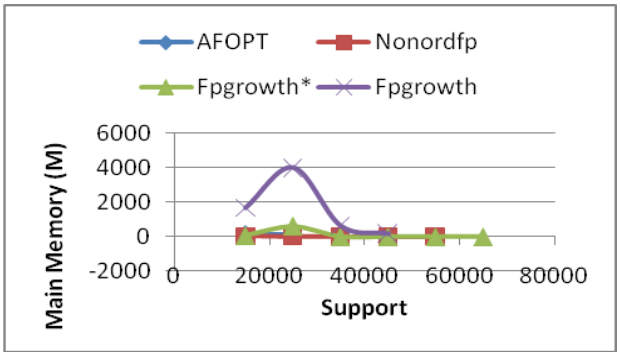


Fig 10 Memory usage on Connect4

AFOPT algorithm keeps its stable consumption of the memory on T40I10D100K. It further confirmed the fact that AFOPT algorithm is stable with sparse data sets. In Fig 9 the competition between the three algorithms is clear, the memory usage is competitive.On Connect4 data set, Fpgrowth* shows stability in the case of the high support thresholds while Nonordfp algorithm remain stable for the low values of support.

6. CONCLUSION

In this paper, we dealt with FP-growth's Variation algorithms. We restricted ourselves to the "Classic" frequent itemsets problem, which is the mining of all frequent itemsets that exist in market basket-like data with respect to support thresholds. The execution time and the memory usage were recorded to see which algorithm is the best. For the time consumption AFOPT algorithm took advantage for most of the data set even though it suffers from segmentation fault in the low support values on connect4 data set. However, for the memory consumption Nonordfp algorithm remains stable for almost all the type of the data set except for the high support values on Connect4. Unfortunately, there is no algorithm work for the all situations. The main restricted for the successful of the algorithm is statistical prosperities of the data set.

REFERENCES

[1] Agrawal , R. , Imieliński , T. , & Swami , A."Mining association rules between sets of items in large databases".In proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207-216, Washington, DC, 1993.
[2] Han, J. , Pei, J. , & Yin, Y. "Mining frequent patterns without candidate generation". In Proc. ACM-SIGMOD Int. Conf. Management of Data (SIGMOD '96), Page 205-216, 2000.
[3] J. Pei, J. Han & R. Mao. CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets", DMKD'00, 2000.
[4] J. Pei, J. Han, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. ICDE'01, 2001.
[5] Liu,G. , Lu ,H. , Yu ,J. X., Wang, W., & Xiao, X.. "AFOPT:An Efficient Implementation of Pattern Growth Approach", In Proc. IEEE ICDM'03 Workshop FIMI'03, 2003.
[6] Grahne, G. , & Zhu, J. "Fast Algorithm for frequent Itemset Mining Using FP-Trees",IEEE Transactions on Knowledge and Data Engineer,Vol.17,NO.10, 2005 .
[7] Gao, J. "Realization of new Association Rule Mining Algorithm" Int. Conf. on Computational Intelligence and Security ,IEEE, 2007.
[8] Han, J. , & Kamber, M. "Data Mining :concepts and techniques" , second edition, The Morgan Kaufmann Series in Data Management Systems, 2006.
[9] Balázes Rácz," nonordfp: An FP-Growth Variation without Rebuilding the FP-Tree**",** 2nd Int'l Workshop on Frequent Itemset Mining Implementations FIMI2004
[10] Grahne O. and Zhu J. "Efficiently Using Prefix-trees in Mining Frequent Itemsets", In Proc. of the IEEE ICDM Workshop on Frequent Itemset Mining, 2004.
[11] http://fimi.cs.helsinki.fi/
[12] http://adrem.ua.ac.be/~goethals/software/

Aiman Moyaid Said received bachelor degree in computer information system from Yarmouk University, Jordan in 2007 .And currently he is doing his master degree in computer information sciences, at Universiti Teknologi PETRONAS. His research interests include Association Rule Mining, Clustering and application of data mining to problems in retailer industry.

Dr. P.D.D.Dominic obtained his M.Sc degree in operations research in 1985, MBA from Regional Engineering College, Tiruchirappalli, India during 1991, Post Graduate Diploma in Operations Research in 2000 and completed his Ph.D during 2004 in the area of job shop scheduling at Alagappa University, Karaikudi, India. Since 1992 he has held the post of Lecturer in the Department of Management Studies, National Institute of Technology (Formally Regional Engineering College), Tiruchirappalli- 620 015, India. Presently he is working as a Senior Lecturer, in the Department of Computer and Information Science, Universiti Teknologi PETRONAS, Malaysia. His fields of interest are Operations Management, KM, E-business and Decisions Support Systems. He has published technical papers in International, National journals and conferences.

Dr Azween Abdullah obtained his bachelors degree in Computer Science in 1985, Master in Software Engineering in 1999 and his Ph.d in computer science in 2003. His work experiences includes twenty years in institutions of higher learning in both the management and academic capacities, and fifteen years in commercial companies as Software Developer and Engineer, Systems Analyst and IT/MIS and educational consultancy and training. He has spent more than a decade with leading technology firms and universities as a process analyst, senior systems analyst, project manager, and lecturer. He have participated in and managed several software development projects. These have included the development of management information systems, software process improvement initiatives design and implementation, and several business application projects.

His area of research specialization includes computational biology, system survivability and security, autonomic computing and self-healing and regenerating systems, formal specifications and network modeling. His contributions include publishing several journal and refereed conference papers and in the development of programs to enhance minority involvement in bridging the ICT digital gap. Currently he is working on two projects funded by the Ministry of Science Technology and Innovation.

ISSN : 1978-6603

JURNAL Ilmiah SAINTIKOM Sains dan Komputer

Penerapan Data Mining dengan Algoritma Fp-Growth untuk Mendukung Strategi Promosi Pendidikan (Studi Kasus Kampus STMIK Triguna Dharma)

Ali Ikhwan[*1], Dicky Nofriansyah[#2], Sriani[#3]

[*1]Magister Ilmu Komputer Universitas Putra Indonesia"YPTK" Padang

[#2,3]Program Studi Sistem Komputer, STMIK TrigunaDharma

Jl. Raya Lubuk Begalung, Sumbar – Indonesia

Jl. A.H. Nasution No. 73 F-Medan

Email :Aliikhwan053@gmal.com

ABSTRAK

Saat ini semakin banyaknya jumlah kampus yang semakin berkembang yang membuat para pengelola ingin mendapatkan strategi promosi yang lebih baik. Adapun cara dalam menemukan strategi promosi yang tepat akan dapat mengurangi biaya promosi dan mencapai sasaran promosi yang tepat. Salah satu cara yang dapat dilakukan untuk penentuan strategi promosi adalah dengan menggunakan teknik data mining. Adapun teknik yang digunakan dalam hal ini adalah Algoritma FP-Growth. FP-Growth adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (frequent item set) dalam sekumpulan data. Algoritma FP-Growth merupakan pengembangan dari algoritma Apriori. Sedangkan di dalam algoritma FP-Growth tidak dilakukan generate candidate karena FP-Growth menggunakan konsep pembangunan tree dalam pencarian frequent itemset. Penelitian dilakukan dengan mengamati beberapa variabel penelitian yang sering dipertimbangkan oleh perguruan tinggi khusunya bagian marketing dalam menentukan sasaran promosinya, Yaitu Pendidikan Terakhir,Alamat Rumah,Jurusan,Pilihan Prodi. Hasil penelitian ini adalah berupa suatu perangkat lunak dengan mengimplementasikan algoritma FP-Growth yang menggunakan konsep pembangunan FP-Tree dalam mencari Frequent Itemset.

Kata Kunci : Data Mining,Association Rules,Frequent Itemset,FP-Growth.

ABSTRAC

Nowadays more and more growing amount of campus that make the managers want to get better promotion strategy. So how to find the right promotional strategy will be able to reduce costs and achieve targeted promotional campaign right. One of proces that can be done to determine promotion strategy is to use data mining techniques . The technique used in this case is the FP-Growth Algorithm. FP-Growth algorithm is one of the alternatives that can be used to determine the set of data that appears most frequently (frequent item sets) in a set of data. FP-Growth algorithm is an extension of Apriori algorithm. While in the FP-Growth algorithm do not generate candidate because the FP-Growth using the concept of development in the search for frequent itemset tree. The research doing by observing and some research variables are often considered by colleges especially the marketing in targeting the promotion, Namely Latest Education, Home Address, Subject, Options Prodi. The results of this research are in the form of a software to implement the FP-Growth algorithm that uses the concept of development of FP-Tree in finding Frequent Itemset.

Keywords: Data Mining,Association Rules,Frequent Itemset,FP-Growth.

1. PENDAAHULUAN

1.1 Latar Belakang Masalah

Persaingan di dalam dunia bisnis ini sangat ketat , khususnya dalam mempromosikan perguruan tinggi. Setiap kampus berusaha memberikan fasilitas yang terbaik. Oleh karena itu setiap kampus berusaha mencari strategi yang tepat untuk mempromosikan kampusnya. Menurut para pengembang untuk menemukan suatu strategi yang dapat menentukan strategi pemasaran dalam mempromosikan perguruan tinggi dengan memaksimalkan pelayanan kepada masyarakat. Salah satu teknik yang dipakai dalam penerapan *Data Mining* adalah di dalam bidang promosi. Bila sasaran promosi tidak ditentukan secara baik, dalam arti tidak diupayakan mencari sasaran promosi yang potensial, maka hanya akan menghabiskan banyak waktu dan biaya yang seharusnya bisa diminimalisir melalui pemilihan target promosi yang baik.

Salah satu cara yang dapat diterapkan adalah dengan menerapkan penggunaan *Data Mining*. Karena di dalam data mining sendiri terdapat cara dan teknik dalam pemenuhan kebutuhan salah satunya adalah kebutuhan informasi yang luas, dan dari informasi yang kita dapat bia kita gunakan sebagai suatu keputusan atau menentukan sebuah kualitas dalam menentukan sauatu keputusan. Adapun kumpulan dari sebuah data atau informasi memeiliki banyak potensi untuk dijadikan suatu kesimpulan dalam mengambil sebuah keputusan dengan melakukan analisis dan menggali suatu unformasi yang terdapat di dalam sebuah data. Maka dimungkinkan untuk dibuat strategi dalam mendukung promosi pendidikan. Dari definisi *Data Mining* yang luas terdapat banyak jenis teknik analisa yang dapat digolongkan dalam *Data Mining*. Teknik *Data Mining* yang akan digunakan dalam penelitian ini adalah *Association Rule Mining*. *Association rule mining* adalah teknik *mining* untuk menemukan aturan asosiatif antara suatu kombinasi *item*.

Adapun dalam penelitian ini akan dibahas bagaimana cara mengimplementasikan salah satu algoritma dalam data mining, yaitu algoritma *frequent Pattern-Growth (FP-Growth)*. Algoritma ini adalah bagian dari teknik asosiasi pada data mining. Adapun *FP-Growth* sendiri adalah salah satu altenatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering mucul *(frequent itemset)* dalam sekumpulan data. Karakteristik algoritma *FP-Growth* adalah struktur data yang digunakan dalam *tree* yang disebut *FP-Tree*. Dengan menggunakan *FP-Tree*, algoritma *FP-Growth* dapat langsung mengekstrak *frequent itemset* dari *FP-Tree.*

1.2 Perumusan Masalah

1. Bagaimana sistem memberikan informasi kepada user dalam mempromosikan pendidikan di Perguruan Tinggi Swasta secara efektif dan efisien?
2. Bagaimana sistem ini menentukan target dalam mempromosikan pendidikan?
3. Bagaimana menerapkan algoritma *FP-Growth* di dalam promosi pendidikan sehingga dalam proses tersebut dapat menentukan strategi yang tepat ?
4. Bagaimana melakukan pengujian Metode *FP-Growth* dengan *Rapidminer* sehingga menghasilkan penyelesain masalah dalam mencari strategi promosi yang tepat?

1.3 Batasan Masalah

penulis mencoba memberikan batasan masalah antara lain:

1. Algoritma *FP- Growth* yaitu algoritma yang digunakan di dalam penyeleksian untuk mempercepat proses penetuan *frequent itemset* sebelum meng-*generate rull* sebagai rekomendasi keputusan.
2. Penelitian ini khusus membahas tentang strategi promosi pendidikan pada Perguruan Tinggi khususnya di STMIK TRIGUNA DHARMA .
3. Data yang digunakan sebagai sumber merupakan data yang diambil di STMIK TRIGUNA DHARMA 1 tahun terakhir, yang meliputi asal sekolah, daerah asal, jurusan dan pilihan prodi.
4. *Tools* yang digunakan di dalam pengolahan data adalah *Rapidminer 5.2*

1.4 Tujuan Penelitian

1. Untuk menganalisa permasalahan promosi pendidikan di STMIK TRIGUNA DHARMA.
2. Untuk menerapkan metode *FP- Growth* di dalam strategi promosi pendidikan di STMIK TRIGUNA DHARMA.
3. Untuk melakukan pengujian metode *FP-Growth* dengan menggunakan tools *Rapidminer* sehingga menghasilkan strategi promosi pendidikan yang efektif dan efisien.
4. Menghasilkan sebuah hubungan antara variable atau item berdasarkan implementasi dari algoritma *FP-Growth.*
5. Menguji hubungan variable atau item tersebut untuk mendapatkan pengetahuan baru dari proses ekstraksi menggunakan *Tools* yang ada.

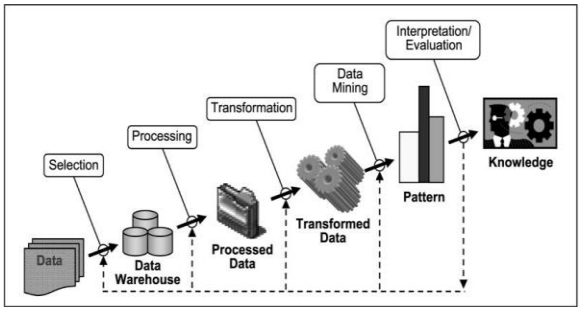
1.5 Manfaat Penelitian

1. Mengetahui algoritma *FP-Growth* dapat memberikan alternatif pengetahuan tentang menemukan strategi dalam promosi pendidikan.
2. Menemukan strategi atau sasaran yang lebih tepat dalam mempromosikan pendidikan.
3. Menjadikan hasil software ini dapat digunakan pihak kampus untuk menentukan strategi dalam promosi pendidikan.

2. LANDASAN TEORI

2.1 *Knowledge Discovery In Database* (KKD)

Knowledge Discovery in Database (KDD) adalah proses menentukan informasi yang berguna serta pola-pola yang ada dalam data. Informasi ini terkandung dalam basis data yang berukuran besar yang sebelumnya tidak diketahui dan potensial bermanfaat. *Data Mining* merupakan salah satu langkah dari serangkaian proses *iterative* KDD. Tahapan proses KDD dapat dilihat pada gambar 2.1.



Gambar 2.1 Tahapan dalam KDD

Tahapan proses KDD terdiri dari:

a. Data *Selection*

Pada proses ini dilakukan pemilihan himpunan data, menciptakan himpunan data target, atau memfokuskan pada subset

variable (sampel data) dimana penemuan (*discovery*) akan dilakukan. Hasil seleksi disimpan dalam suatu berkas yang terpisah dari basis data operasional.

b. Pre-*Processing* dan *Cleaning* Data

Pre-*Processing* dan *Cleaning* Data dilakukan membuang data yang tidak konsisten dan *noise*, duplikasi data, memperbaiki kesalahan data, dan bisa diperkaya dengan data *eksternal* yang relevan.

c. *Transformation*

Proses ini *mentransformasikan* atau menggabungkan data ke dalam yang lebih tepat untuk melakukan proses *mining* dengan cara melakukan peringkasan (*agregasi*).

d. *Data Mining*

Proses *Data Mining* yaitu proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik, metode atau algoritma tertentu sesuai dengan tujuan dari proses KDD secara keseluruhan.

e. *Interpretation* / Evaluasi

Proses untuk menerjemahkan pola-pola yang dihasilkan dari *Data Mining*. Mengevaluasi (menguji) apakah pola atau informasi yang ditemukan bersesuaian atau bertentangan dengan fakta atau hipotesa sebelumnya. Pengetahuan yang diperoleh dari pola-pola yang terbentuk dipresentasikan dalam bentuk visualisasi.

2.2 Data *Mining*

Data *Mining* adalah proses untuk mendapatkan informasi dengan melakukan pencarian pola dan relasi-relasi yang tersembunyi di dalam timbunan data yang banyak. (Fadlina, 2014) Data *Mining* atau sering disebut sebagai *knowledge discovery in database* (KDD) adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam data berukuran besar. Keluaran *Data Mining* ini bisa dipakai untuk membantu pengambilan keputusan di masa depan. Pengembangan KDD ini menyebabkan penggunaan *pattern recognition* semakin berkurang karena telah menjadi bagian *Data Mining*.

2.3 Tahapan dalam Data Mining

Menurut (Buulolo, 2013) ada empat tahap yang dilalui dalam data mining, antara lain:

1. Tahap pertama : *Pricise statement of the problem* (pernyataan tepat terhadap permasalahan)

Sebelum mengakses perangkat lunak data mining, seorang analis harus memiliki kejelasan perihal ‘pertanyaan apa yang akan ingin dijawabnya’. Jika tidak ada *formula* yang tepat untuk problematika yang ada maka anda hanya akan membuang-buang dan uang dalam membuat solusinya.

2. Tahap kedua : *Initial exploration*

Tahap ini dimulai dengan mempersiapkan data yang juga juga termasuk kedalam data mining “*cleaning*” (misalnya : mengedentifikasi dan menyikirkan data yang dikodekkan salah), transformasi data, memilih *subset record, data set,* langkah awal seleksi. Mendeskripsikan dan memvisualisasikan data adalah kunci dari tahap ini.

3. Tahap tiga : *Model building and validation*

Tahap ini melibatkan pertimbangan terhadap ragam permodelan dan memilih yang terbaik bagi performasi prediktif.

4. Tahap ke-empat : *Deployment*

Memilih aplikasi yang tepat berikut permodelan untuk membuat (*generate*) prediksi. Selanjutnya kita kan melihat rincian perihal tahapan-tahapan data mining.

2.4 Pengelompokan *Data Mining*

Data Mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu (Kusrini dan Emha Taufiq Luthfi, 2009):

1. Deskripsi
   Terkadang peneliti dan analis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menentukan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden.

2. Estimasi
   Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih kearah numerik daripada kearah kategori. Sebagai contoh akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya.

3. Prediksi
   Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang.
   Contoh prediksi bisnis dan penelitian adalah:
   a. Prediksi harga beras dalam tiga bulan yang akan datang.
   b. Prediksi persentasi kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikkan.
   Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

4. Klasifikasi
   Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang dan pendapatan rendah.
   Contoh lain klasifikasi dalam bisnis dan penelitian adalah:
   a. Menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang atau tidak.
   b. Memperkirakan apakah suatu pengajuan hipotek oleh nasabah merupakan suatu kredit yang baik atau buruk.
   c. Mendiagnosis penyakit seorang pasien untuk mendapatkan termasuk kategori penyakit apa.

5. Pengklusteran
   Pengklusteran merupakan pengelom-pokan *record*, pengamatan atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan.
   Contoh pengklusteran dalam bisnis dan penelitian adalah:

a. Mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari satu suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar.
b. Untuk tujuan audit akuntansi, yaitu melakukan pemisahan terhadap perilaku *financial* dalam baik dan mencurigakan.
c. Melakukan pengklusteran terhadap ekspresi dari *gen*, untuk mendapatkan kemiripan perilaku dari *gen* dalam jumlah besar.

6. Asosiasi
   Tugas asosiasi dalam *Data Mining* adalah menemukan *attribut* yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja. Contoh asosiasi dalam bisnis dan penelitian adalah:
   a. Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respon positif terhadap penawaran *upgrade* layanan yang diberikan.
   b. Menentukan barang dalam supermarket yang dibeli secara bersamaan dan yang tidak pernah dibeli secara bersamaan.

2.5 Tahapan Asosiasi

Analisis asosiasi dikenal juga sebagai salah satu teknik *Data Mining* yang menjadi dasar dari berbagai teknik *Data Mining* lainnya. Khususnya salah satu tahap dari analisis asosiasi yang disebut analisis pola frequensi tinggi *(frequent pattern mining)* menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien. Penting tidaknya suatu aturan assosiatif dapat diketahui dengan dua parameter, *support* (nilai penunjang) yaitu persentase kombinasi item tersebut dalam *database* dan *confidence* (Fadlina, 2014).

A. Data Preparation.

Pada banyak bidang keilmuan, terutama komputer sains, diperlukan data yang berkualitas melalui proses *preparation* dari data mentah. Dalam prakteknya, ditemukan bahwa data *cleaning* dan *preparation* membutuhkan total usaha 80% dari usaha untuk merekayasa data, sehingga menjadikan data *preparation* sebagai proses yang krusial. Pentingnya proses ini dapat dilihat dari tiga aspek, yakni:

1. Data *real world* merupakan data kotor. Data *real world* bisa mengandung data yang tidak, terdapat *noise*, tidak konsisten, yang dikarenakan:
   a. Tidak lengkap (*incomplete*), yaitu kekurangan nilai atribut atau hanya mengandung agregat data (contoh : address = " ")
   b. *Noise*, yaitu masih mengandung *error* dan *outliers*.
   c. Tidak konsisten (*inconsistent*), yaitu data yang mengandung discrepansi dalam code dan nama atau singkatnya datanya tidak konsisten.
2. Sistem *mining* dengan performa tinggi membutuhkan data yang berkualitas. Data *preparation* atau *preparation* menghasilkan dataset yang lebih sedikit daripada dataset yang asli, ini bias meningkatkan efisiensi dari *Data Mining*. Langkah ini mengandung:
   a. Memilih data yang relevan
   b. Mengurangi data.
3. Data yang berkualitas menghasilkan pola yang berkualitas. Dengan data *preparation*, maka data yang dihasilkan adalah data yang berkualitas, yang mengarah pada pola yang berkualitas pula, dengan:

a. Mengembalikan data yang tidak lengkap
b. Membenarkan *eror*, atau menghilangkan *outliers*
c. Membenahi data yang bertentangan.

B. Association Rules

Association rule merupakan salah satu metode yang bertujuan mencari pola yang sering muncul di antara banyak transaksi, dimana setiap transaksi terdiri dari beberapa *item* sehingga metode ini akan mendukung sistem rekomendasi melalui penemuan pola antar *item* dalam transaksi-transaksi yang terjadi
Metodologi dasar analisis asosiasi terbagi menjadi dua tahap:

1. Analisa pola frekuensi tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai support dalam *database*. Nilai *support* sebuah *item* diperoleh dengan rumus berikut:

$$Support\ (A) = \frac{\text{Jumlah Transaksi Mengandung A}}{\text{Total Transaksi}} \quad (1)$$

$$Support\ (A\ C\ B) = \frac{\text{Jumlah Transaksi Mengandung A dan B}}{\text{Total Transaksi}} \quad (2)$$

2. Pembentukan aturan assosiatif

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan assosiatif yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan assosiatif A _B Nilai *confidence* dari aturan A _B diperoleh dari rumus berikut:

C. Frequent Itemset

Langkah pertama pada *association rule* adalah menghasilkan semua *itemset* yang memungkinkan dengan kemungkinan *itemset* yang muncul dengan *m-item* adalah 2*m*. Karena besarnya komputasi untuk menghitung *frequent itemset*, yang membandingkan setiap kandidat *itemset* dengan setiap transaksi, maka ada beberapa pendekatan untuk mengurangi komputasi tersebut, salah satunya dengan algoritma apriori.

D. Apriori

Algoritma apriori digunakan untuk mencari *frequent itemset* yang memenuhi *minsup* kemudian mendapatkan *rule* yang memenuhi *minconf* dari *frequent itemset* tadi. Algoritma ini mengontrol berkembangnya kandidat *itemset* dari hasil *frequent itemset* dengan *support-based pruning* untuk menghilangkan *itemset* yang tidak menarik dengan menetapkan *minsup*. Prinsip dari apriori ini adalah bila *itemset* digolongkan sebagai *frequent itemset*, yang memiliki *support* lebih dari yang ditetapkan sebelumnya, maka semua subsetnya juga termasuk golongan *frequent itemset*, dan sebaliknya.

E. Rule Generation

Setelah mendapatkan *frequent itemset* menggunakan algoritma apriori, langkah selanjutnya adalah mendapatkan *rule* yang memenuhi *confidence*. Karena *rule* yang dihasilkan berasal dari *frequent itemset*, dengan kata lain, dalam menghitung *rule* menggunakan *confidence*, tidak perlu lagi menghitung *support*-nya karena semua calon *rules* yang dihasilkan telah memenuhi *minsup* sesuai yang ditentukan. Penghitungan ini juga tidak perlu melakukan perulangan *scanning* pada *database* untuk menghitung *confidence*, cukup dengan mengambil *itemset* dari hasil *support.*

2. 6 Algoritma *FP-Growth*

(Sensuse, 2012)*FP-growth* adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. *FP-growth* menggunakan pendekatan yang berbeda dari paradigma yang digunakan pada algoritma Apriori.

(Ririanti, 2014). *FP-Growth* adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent item set*) dalam sekumpulan data. Algoritma *FP-Growth* merupakan pengembangan dari algoritma *Apriori*.

FP-growth adalah metode yang sering *itemset* pertambangan tanpa Generasi calon. Ini membangun sebuah struktur data yang sangat padat *(FP-tree)* untuk kompres *database* transaksi asli.

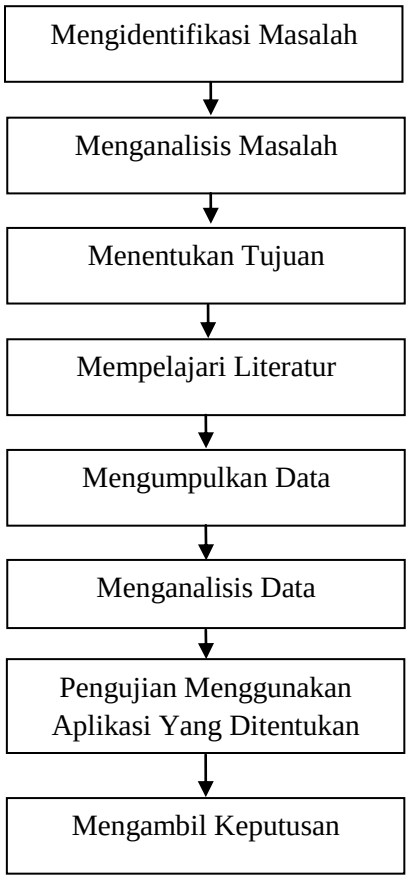
3. METODOLOGI PENELITIAN

3.1 Pendahuluan

Dalam melakukan penelitian agar hasilnya bisa maksimal, penulis mengikuti kaidah-kaidah (metode) yang telah ditetapkan. Metode penelitian yang akan diterapkan adalah metode penelitian ilmiah yang telah diajarkan di bangku perkuliahan sarjana. Metodologi penelitian ini memuat tentang kerangka kerja penelitian yang akan dibahas di bawah ini. Pada bab ini akan dibahas metodologi penelitian untuk mengidentifikasi permasalahan, analisa permasalahan dan pada akhirnya mencari penyelesaian masalah di dalam menganalisa strategi dalam mempromosikan pendidikan pada STMIK TRIGUNA DHARMA MEDAN.

3.2 Kerangka Kerja

Dalam metodologi penelitian ada urutan kerangka kerja yang harus diikuti, urutan kerangka kerja ini merupakan gambaran dari langkah–langkah yang harus dilalui agar penelitian ini bisa berjalan dengan baik. Kerangka kerja yang harus diikuti bisa dilihat pada gambar 3.1.



Gambar 3.1 Kerangka Kerja

Berdasarkan gambar 3.1 dijabarkan urutan–urutan langkah kerja sebagai berikut:

3.2.1. Mengidentifikasi Masalah

Masalah yang diidentifikasi dalam penelitian ini untuk mendapatkan strategi promosi yang efektif dan efisien dalam mendapatkan calon mahasiswa baru

ditengah ketatnya persaingan antar Perguruan Tinggi .

3.2.2. Menganalisis Masalah

Analisis masalah pada penelitian ini dilakukan dengan dua metode yaitu metode deskriptif dan metode komperatif.

a. Metode Deskriptif
   Pada metode ini data yang ada dikumpulkan, disusun, dikelompokkan dan dianalisis sehingga diperoleh beberapa gambaran yang jelas pada masalah yang akan dibahas.

b. Metode Komperatif
   Pada metode ini analisis dilakukan dengan cara membandingkan teori dan praktek sehingga diperoleh gambaran yang jelas tentang persamaan dan perbedaan di antara keduanya.

3.2.3. Menentukan Tujuan

Berdasarkan uraian di atas, tujuan dari penelitian ini adalah:

a. Untuk mengklasifikasikan setiap wilayah di Sumatera Utara dalam menentukan target promosi pendidikan di STMIK TRIGUNA DHARMA MEDAN.
b. Untuk mendapatkan atau menghasilkan suatu knowledge (pengetahuan) tentang wilayah yang akan dipilih dalam mempromosikan pendidikan.
c. Untuk memberikan masukan kepada pihak akademik sebagai bahan pertimbangan pihak akademik di dalam menentukan wilayah yang akan dijadikan target dalam promosikan pendidikan.

Mempelajari Literatur

Literatur-literatur yang dipakai sebagai bahan referensi dalam penelitian ini adalah dari jurnal-jurnal ilmiah, modul pembelajaran dan buku tentang *Data Mining*. literatur-literatur ini akan menjadi pedoman untuk melakukan penelitian agar memudahkan proses penelitian.

3.2.4. Mengumpulkan Data

Metode pengumpulan data dilakukan dengan cara melakukan pengamatan langsung ke kampus STMIK TRIGUNA DHARMA. Selain pengamatan, juga dilakukan wawancara kepada pihak-pihak yang terkait dengan penelitian ini. Di samping itu juga, melakukan pengambilan sampel *database* nasabah untuk menunjang penelitian ini.

3.2.5. Menganalisis Data

Pada tahap ini data yang telah dikumpulkan akan dianalisis. Analisis menggunakan *Association Rule* yang dimaksud dilakukan melalui mekanisme penghitungan *support* dan *confidence* dari suatu hubungan item. Sebuah rule asosiasi dikatakan *interesting* jika nilai *support* adalah lebih besar dari *minimum support* dan juga nilai *confidence* adalah lebih besar dari *minimum confidence* .Dengan menggunakan teknik *FP-Growth* yang menghasilkan *Frequent Itemset* tanpa melakukan *candidates generation*. dengan tujuan untuk mendapatkan pengetahuan yang baru (*knowledge*) berupa informasi tentang strategi promosi yang efektif dan efisien dalam mendapatkan calon mahasiswa baru ditengah ketatnya persaingan antar Perguruan Tinggi.

3.2.6. Pengujian Menggunakan Aplikasi Yang Ditentukan

Pada tahap ini, *rule* dites kembali atau diuji lagi menggunakan sistem *data*

mining yang sudah ada. *Tools* yang digunakan sebagai pengujian sistem adalah *Ramidminer 5. 2*.

Mula-mula data manual ”Data Mahasiswa Baru” yang sudah ada sebelumnya, di transformasikan ke dalam *Microsoft Excel*.

Sebagai tahap awal di dalam proses pengujian yaitu *mengimport* data mahasiswa baru pada kampus STMIK TRIGUNA DHARMA yang akan dijadikan sebagai data tabel transaksi *frequensi item* di dalam *tools Ramidminer 5 2.* tersebut. Kemudian tahap selanjutnya yaitu, melakukan proses *Data Input* dan *Affinity analysis* atau *market basket analysis*. Analisis asosiasi atau *association rule* adalah teknik *Data Mining* untuk menemukan aturan asosiatif antara suatu kombninasi *item*. Analasis *FP-Growth* didefenisikan suatu proses untuk menemukan semua aturan *FP-Growth* yang memenuhi syarat *minimum support* dan syarat minimum untuk *confidence*. Dari proses *Data Input* dan *Affinity* tersebut akan terdapat pola atau kriteria tertentu dalam mendapatkan strategi promosi. Hasil dari Proses *Data Input* dan *Affinity* tersebut akan diolah kembali untuk menghasilkan suatu *association rule*. Dari *association rule* tersebut akan terlihat hasilnya. Hasilnya tersebut merupakan pengetahuan baru *(knowledge)* dan juga merupakan sebagai acuan untuk menganalisa strategi dalam mempromosikan pendidikan.

3.2.7. Pengambilan Keputusan

Setelah diuji, hasil analisis antara *manual ways* (cara manual) dan pengujian memanfaatkan *tools* akan terlihat perbandingannya. Langkah berikutnya adalah penentuan atau pengambilan keputusan terhadap *knowledge* yang baru didapat yaitu berupa penentuan dalam menentukan strategi dalam mempromosikan pendidikan.

4. ANALISA

4.1 Analisa Data

Karakteristik algoritma *FP-Growth* adalah struktur data yang digunakan adalah *tree* yang disebut dengan *FP-Tree*. Dengan menggunakan *FP-Tree*, algoritma *FP-growth* dapat langsung mengekstrak *frequent Itemset* dari *FP-Tree.* Penggalian *itemset* yang *frequent* dengan menggunakan algoritma *FP-Growth* akan dilakukan dengan cara membangkitkan struktur data *tree* atau disebut dengan *FP-Tree*. Metode *FP-Growth* dapat dibagi menjadi 3 tahapan utama yaitu sebagai:

1. Tahap pembangkitan *conditional pattern base*,
2. Tahap pembangkitan *conditional FP-Tree*, dan
3. Tahap pencarian *frequent itemset*.

   Ketiga tahap tersebut merupakan langkah yang akan dilakukan untuk mendapat *frequent itemset*.

Input : *FP-Tree Tree*
Output : *Rt* Sekumpulan lengkap pola *frequent*
Methode : *FP-Growth (Tree, null)*
Procedure : *FP-Growth (Tree, α)*
{
01*: if Tree* mengandung *single path* P;
02*: then* untuk tiap kombinasi (dinotasikan β) dari *node-node* dalam *path do*
03: bangkitkan pola β α dengan *support* dari *node-node* dalam *path do* β;
04: *else u*ntuk tiap a1 dalam *header* dari *tree do*
}
05: bangkitkan pola

06: bangun β = a1 α dengan *support* = a1 *support*
07*: if Tree* β =

4.2 Pra Proses

Tabel 4.1 Untuk Memberi Tanda Pada Setiap *Itemset*

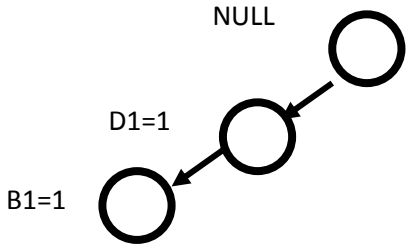
Item
SMK Negeri (A1)
SMK Swasta(A2)
SMA Negeri (A3)
SMA Swasta (A4)
Madrasah Aliyah (A5)
Medan (B1)
Aceh (B2)
Lubuk Pakam (B3)
Deli Serdang (B4)
Perbaungan (B5)
Tebing Tinggi (B6)
Riau (B7)
IPA (C1)
IPS (C2)
TKJ (C3)
MULTIMEDIA (C4)
Sistem Informasi (D1)
Sistem Komputer (D2)
Teknik Komputer (D3)
Manajemen Informatika (D4)

Tabel 4.2 Frekuensi dari Setiap Transaksi *Item*

TID	ITEM
1	D1,B1
2	C1,A3,B1
3	A3,C2
4	D1,A4,B1,C2
5	D1,C1,A4
6	C1,A3
7	D1,C1,A3
8	D1
9	D1,C1,A3
10	D1,C1,A4
11	D1,C1,A4
12	A3,B1.C2
13	D1,C1
14	C1
15	D1,C1,A3
16	C1,A3
17	D1,C1,A3
18	-
19	D1,A4,C2
20	D1,A4,B1,C2
21	A3,B1,C2
22	A3,B1,C2
23	D1,B1
24	D1
25	D1,C1,A4,B1
26	D1,C1,A4,B1
27	D1,C1,A4
28	D1,C1,A3
29	C2
30	D1,A4,C2

Maka langkah selanjutnya adalah membentuk pohon *FP-Tree* dengan melihat tabel 4.2 sebagai berikut:

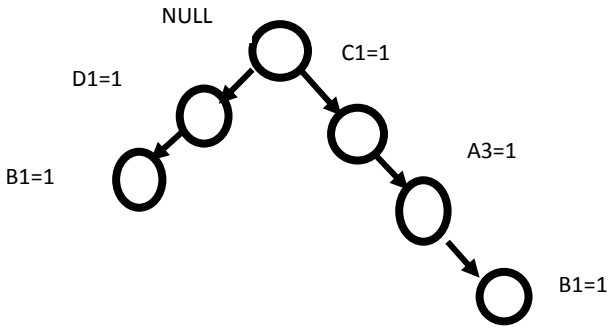
Gambar di bawah ini memberikan ilustrasi mengenai pembentukan *FP-tree* setelah pembacaan TID pada tabel 4.2



Gambar 4.1: Hasil Pembentukan *FP-tree* Setelah Pembacaan TID 1

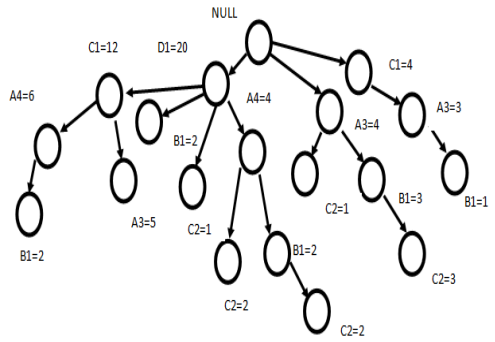
Gambar 4.1 adalah penjelasan tentang pembentukan *FP-Tree* setelah

pembacaan didapat setelah melakukan TID 1, Yaitu berisi : NULL- D1 (Sistem Informasi) =1 –(B1) Medan= 1.



Gambar 4.2 : Hasil Pembentukan *FP-tree* Setelah Pembacaan TID 2

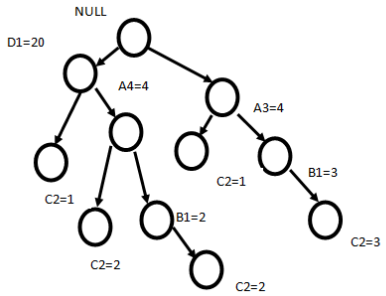
Gambar 4.2 didapat setelah melakukan TID 2 ,Yaitu: NULL-C1 (IPA) =1-A3 (SMAN) =1-B1(Medan) =1.



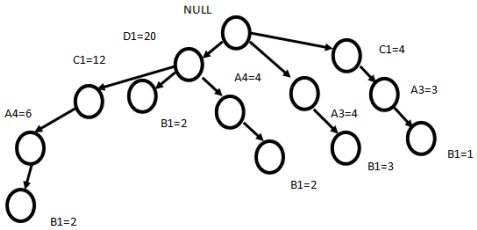
Gambar 4.3 : Hasil Pembentukan *FP-tree* Setelah Pembacaan TID 30

Gambar didapat setelah melakukan TID 30 yang dijumlahkan, yaitu berisi Null-Sistem Informasi (D1) = 20 -IPA (C1) =16 -SMA Negeri (A3) =12 -SMA Swasta (A4) =10-Medan (B1) = 10-IPS (C2) = 9.

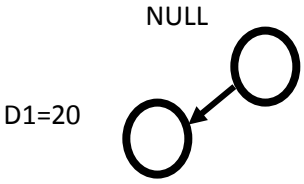
Untuk menemukan *Frequent itemset* dari tabel 4.2, maka perlu ditentukan terlebih dahulu lintasan yang berakhir dengan *support count* terkecil, yaitu C2 yang diikuti dengan A4,A3, C1,D1 dan di akhiri D1. Proses pembentukan masing-masing *node* dapat dilihat pada gambar 4.4.



Gambar 4.4 Lintasan yang Mengandung Simpul C2



Gambar 4.5 Lintasan yang Mengandung Simpul B1



Gambar 4.6 Lintasan yang Mengandung Simpul D1

Setelah mencari *frequent itemset* untuk beberapa akhiran *suffix* maka didapat hasil yang dirangkum dalam tabel berikut:

Tabel 4.3 Daftar *Frequent Itemset* Diurutkan Berdasarkan Hubungan Akhiran

Suffix	*Frequent Itemset*
C2	{C2},{C2,A4}{C2,A3}{C2,D1}{C2,B1,A3},{C2,B1,A4}
B1	{B1},{B1,D1},{B1,A3},{B1,A3,C1},(B1,A4,D1},{B1,A4,C1,D1}
A4	{A4},{A4,C1,D1}

A3	{A3},{A3,C1},{A3,C1,D1}
C1	{C1},{ C1,D1}
D1	{D1}

Dari *frequent itemset* yang didapat dari pembentikan *Fp-Tree* dan *FP-Growth* maka dapat dihitung nilai *Support* dan *Confidence* sebagai berikut:

Support = (Sistem Informasi,IPA,SMA Swasta,Medan) = *Count* (Sistem Informasi,IPA,SMA Swasta,Medan)/Jumlah Transaksi = 1/30.

Support = (Sistem Informasi,IPA,SMA Negeri) = *Count* (Sistem Informasi,IPA,SMA Negeri)/Jumlah Transaksi = 1/30.

Support = (Sistem Informasi, Medan) *Count* (Sistem Informasi, Medan)/Jumlah Transaksi =1/30.

Support = (Sistem Informasi,SMA Swasta, IPS) *Count*(Sistem Informasi,SMA Swasta, IPS)/Jumlah Transaksi = 1/30.

Support = (Sistem Informasi,SMA Swasta,Medan, IPS) *Count* (Sistem Informasi,SMA Swasta, Medan,IPS)/Jumlah Transaksi =1/30.

Support = (SMA Negeri ,IPS) *Count* (SMA Negeri ,IPS) / Jumlah Transaksi = 1/30.

Support = (SMA Negeri ,Medan,IPS) *Count* (SMA Negeri ,Medan,IPS) / Jumlah Transaksi = 1/30.

Support = (IPA,SMA Negeri ,Medan) *Count* (IPA,SMA Negeri ,Medan) / Jumlah Transaksi = 1/30.

Sedangkan untuk *confidence* atau nilai kepercayaannya adalah sebagai berikut :

Confidence = (SMA Negeri,IPS) = Count (SMA Negeri,IPS)/*Count* Medan= 1/9.

Confidence = (Sistem Informasi,SMA Swasta,Medan,IPS) = *Count* (Sistem Informasi,SMA Swasta,Medan,IPS /*Count* Medan = 1/9.

Confidence = (SMA Negeri,Medan,IPS) *Count* (SMA Negeri,Medan,IPS /*Count* Medan =1/9.

Confidence = (Sistem Informasi,SMA Swasta,IPS) *Count*(Sistem Informasi,SMA Swasta,IPS)/*Count* Medan= 1/19.

Confidence = (Sistem Informasi,SMA Swasta,Medan,IPS) *Count* (Sistem Informasi,SMA Swasta,Medan,IPS)/*Count* Medan =1/9.

Confidence = (SMA Negeri, Medan,IPS) *Count* (SMA Negeri, Medan,IPS) / *Count* Medan =1/9.

Confidence = (Sistem Informasi,IPS) *Count* (Sistem Informasi,IPS) / *Count* Medan =1/9.

Confidence = (Sistem Informasi,SMA Swasta,IPS) *Count* (Sistem Informasi,SMA Swasta,IPS) / *Count* Medan =1/9.

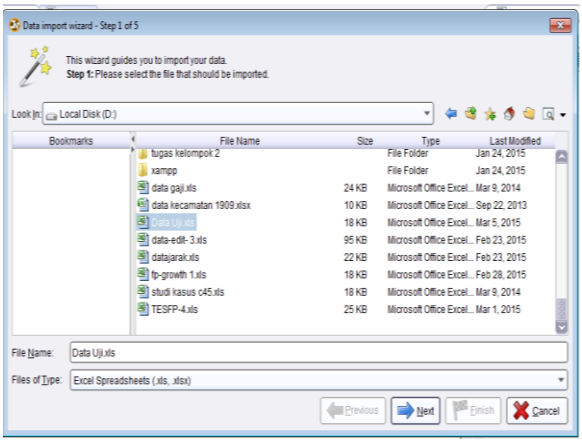
Setelah didapat nilai *support* dan *confidence* dari keseluruhan kombinasi pada data dengan perhitungan *FP-Tree* dan *FP-Growth* maka didapat niai *support* dan *confidence* yang paling tinggi dan akurat yaitu kombinasi. (D1,C1,A3) { Sistem Informasi, IPA, SMA Negeri } yang mempunyai nilai *support* : 5/30 = 0,16 dan nilai *confidence* : 5/20 = 0.25.

5. IMPLEMENTASI SISTEM

Untuk membuktikan kebenaran hasil analisa diperlukan sebuah proses pengujian untuk menguji kebenaran dari hasil pengolahan data yang dikerjakan secara manual pada BAB IV yang telah dikerjakan sebelumnya, untuk proses pengujian tersebut kita dapat menggunakan salah satu *software* aplikasi seperti *Rapidminer,* dengan langkah-langkah sebagai berikut:

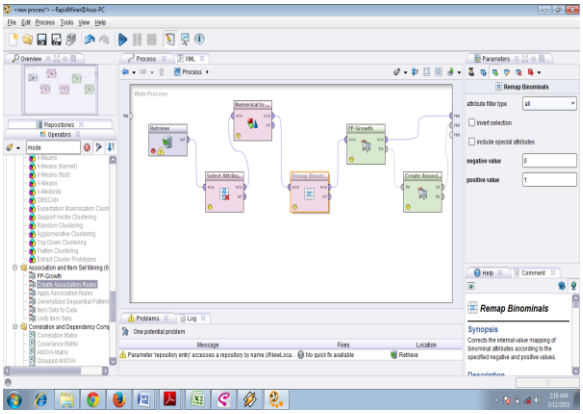
1. Dataset yang terdiri 30 record data dan berisi variabel-variabel atribut Sistem Informasi,IPA,SMA Negeri,SMA

Swasta,Medan, IPS disimpan dalam aplikasi *microsoft excel* dengan nama *file* datauji.xls dan yang akan dicoba menggunakan *software rapidminer* 5.2 untuk melihat hasil sama tidaknya dengan pencarian frequent itemsetnya melalu :



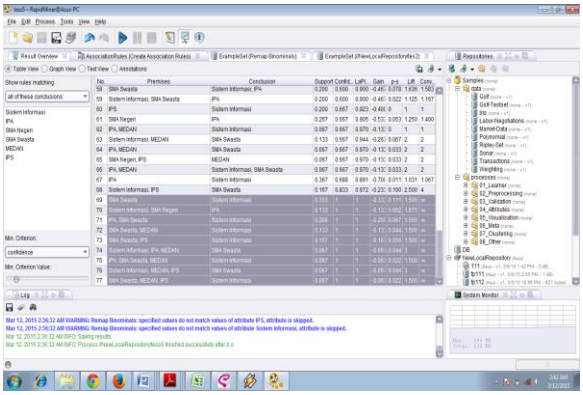
Gambar 5.1 Tampilan *Import Excel Sheet*

Gambar 5.1 adalah proses dari import data uji Xls ke dalam rapidminer



Gambar 5.2 Proses menghubungkan *FP-Growth* ke ras pertama dan *Create Association Rules* ke res kedua

Pada Gamabar 5.2 dijelaskan tentang proses menghubungkan *FP-Growth* ke ras Pertama dan *Create Association Rules* ke ras kedua dan diatur pada minimum *support* dan *minimum confidence*nya.



Gambar 5.20 *Rule valid* jika nilai *Lift Ratio* > 1

Lift Ratio adalah parameter penting selain *support* dan *confidence* dalam *association rule*. *Lift Ratio* mengukur seberapa penting *rule* yang teleh terbentuk berdasarkan nilai *support* dan *confidence*. *Lift Ratio* merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar produk A dibeli bersamaan dengan produk B.

Improvement Ratio dapat dihitung dengan rumus

$$\frac{Support(A \cap B)}{SupportA * SupportB} =$$

Untuk mencari nilai *rule* yang *valid* adalah jika mempunyai nilai *Lift Ratio* > 1 dengan cara *Support Lift Ratio* = *Support* yang mengandung nilai A dan nilai B dibagi *support* A* *Support* B nilai *support* yang mengandung nilai A dan nilai B Adalah hasil dari minimum *support* dibagi dengan *item* (D1,C1,A3) { Sistem Informasi, IPA, SMA Negeri }Pada setiap kemunculan ditransaksi.

=

$$\frac{5/30}{20/30*16/30*12/30} = \frac{0{,}16}{0{,}66*0{,}53*0{,}4} =$$

$$\frac{0{,}16}{0{,}1392} = 1{,}1494$$

Jadi dari hasil pencarian nilai *Lift Ratio* > 1 maka kita bisa menentukan sebuah *rule* yang *valid* dari sekian banyaknya rule yang dijalankan dengan *software Rapidminer*.

Adapun hasil dari rule yang paling mempangaruhi adalah:

Jika dia Asal sekolahnya SMA Swasta, Alamatnya Medan dan Asal Jurusanya IPS maka dia memilih Prodi Sistem Informasi dengan tingkat kepercayaan 100% dan didukung 6% dari data keseluruhan.

Hasil dari rule tersebut yang akan dijadikan target dalam mempromosikan pendidikan.

6. SIMPULAN DAN SARAN

6.1 Simpulan

1. Metode dalam pencarian *Frequent Itemsset* pohon keputusan menggunakan algoritma *FP-Ggrowth* bekerja sangat baik dalam melakukan *Frequent Itemsset* dengan proses pembentukan *FP-Tree* dengan menghasilkan *rule* dari data sampel mahasiswa baru.
2. Penentuan data variabel sangat menentukan tingkat akurasi *FP-Growth* yang dibuat dan besarnya presentase dalam menentukan minum *support* dan minimum *confidence* dipengaruhi oleh data variabel yang digunakan untuk mencari *frequent itemset* yang saling berhubungan untuk menemukan data variabel yang akan dijadikan stretegi dalam promosi pendidikan.
3. Dari penelitian yang dilakukan ada beberapa atribut yang tidak digunakan dalam *rule* yang dihasilkan, sehingga pemilihan atribut di dalam *dataset* sangat penting.
4. Algoritma *FP-Growth* dapat diterapkan untuk mendukung strategi promosi pendidikan pada Perguruan Tinggi. Adapun Informasi yang berkaitan dengan pelaksanaan promosi dapat tersedia dengan cepat, contohnya dalam pemilihan tempat yang akan dijadikan promosi pendidikan, sehingga pihak manajemen dari kampus sendiri dapat melakukan pengambilan keputusan dengan cepat.
5. Pelaksanaan promosi pendidikan pada perguruan tinggi sangat dibantu dengan adanya penerapan algoritma *FP-Growth* ini sehingga diharapkan efektifitas pelaksanaan promosi pendidikan akan dapat semakin ditingkatkan.
6. Untuk menemukan *rule* atau *knowledge* yang sangat berarti dalam menganalisa strategi dalam menentukan promosi pendidikan, dibutuhkan banyak data sampel yang digunakan untuk proses ekstraksi *rule* atau *knowledge* tersebut.

6.2 Saran

Salah satu tujuan penelitian ini adalah agar teknik dalam menentukan strategi promosi pendidikan di Kampus STMIK TRIGUNA DHARMA MEDAN dapat *diimplementasikan* atau diterapkan dengan baik, untuk itu saran yang dapat diperhatikan untuk *pengimplementasian* dari teknik ini adalah:

1. Pada penelitian ini, penulis hanya mencoba salah satu teknik dalam menemukan hubungan beberapa *item* yang digunakan untuk membentuk FP-*Tree* dalam menentukan strategi

promosi pendidikan dengan menggunakan Algorima *FP-Growth.* Untuk menentukan stategi dalam promosi pendidikan yang baik dapat digunakan beberapa atau penggabungan beberapa teknik yang akan digabungkan dalam sistem pengambilan keputusan untuk menentukan stategi yang tepat dalam menentukan stategi promosi pendidikan atau bisa juga menggunakan algoritma *CT-Pro.* Oleh sebab itu perlunya dilakukan penelitian lebih lanjut untuk mendapatkan strategi promosi dan menentukan teori mana yang menghasilkan strategi yang paling baik.

2. Untuk mendapatkan hasil promosi yang baik sangat diperlukan sumber data yang yang lebih banyak dan lebih lengkap.
3. Untuk mengimplementasikan *rule* yang dihasilkan dari teknik dari algoritma *FP-Growth* dapat berkonsultasi dengan tim marketing yang ada di kampus tersebut. Sehingga dengan adanya hubungan algoritma *FP-Growth* dengan pihak manajemen dapat mengembangkan strategi dalam promosi pendidikan sehingga pihak kampus akan lebih maju lagi kedepannya.

7. DAFTAR PUSTAKA

BUULOLO, E. 2013. Implementasi Algoritma Apriori Pada Sistem Persediaan Obat (Studi Kasus : Apotik

Rumah Sakit Estomihi Medan). *Pelita Informatika Budi Darma,* Vol : IV No: 1**,** 71-83.

FADLINA. 2014. Data Mining Untuk Analisa Tingkat Kejahatan Jalanan Dengan Algoritma Association Rule Metode Apriori (Studi Kasus Di Polsekta Medan Sunggal). *Informasi dan Teknologi Ilmiah (INTI),* Vol : III No : 1**,** 144-154.

RIRIANTI. 2014. Implementasi Algoritma FP-GROWTH Pada Aplikasi Prediksi Persedian Sepeda Motor (Studi Kasus PT. Pilar Deli Labumas). *Pelita Informatika Budi Darma,* Vol : VI No: 1.

SENSUSE, G. G. D. D. I. 2012. Penerapan Metode Data Mining Market Basket Analysis Terhadap Data Penjualan Produk Buku Dengan Menggunakan Algoritma Apriori dan Frequent Pattern Growth (FP-GROWTH) : Studi Kasus Percetakan PT. GRAMEDIA. *Jurnal TELEMATIKA MKOM,* Vol.4 No.1**,** 118-132.

KUSRINI dan EMHA TAUFIQ LUTHFI. 2009. “Algoritma Data Minning.” Jakarta : Andi. 10

ANALISA POLA BELANJA MENGGUNAKAN ALGORITMA FP GROWTH, SELF ORGANIZING MAP (SOM) DAN K MEDOIDS

Muhammad Imam Ghozali
Fakultas Teknik, Program Studi Teknik Informatika
Universitas Muria Kudus
Email: imam.ghozali@umk.ac.id

Ri'fan Zaenal Ehwan
SMA N 1 Jakenan Pati
Email : revan.ze@gmail.com

Wibowo Harry Sugiharto
Fakultas Teknik, Program Studi Teknik Informatika
Universitas Muria Kudus
Email: wibowo.harry@umk.ac.id

ABSTRAK

Dalam sebuah bisnis, diperlukan upaya memaksimalkan keuntungan. Diantaranya dengan melakukan promosi. Ketepatan promosi dapat dipelajari dari database sebuah perusahaan ritel utamanya pola belanja pada produk yang biasa dibeli bersamaan. Informasi tentang pola belanja pelanggan yang tidak akurat menyebabkan kebijakan promosi tidak tepat dan efisien.Salah satu upaya lazim untuk memperoleh dan menggali pola belanja pelanggan adalah menggunakan data mining yang dikenal sebagai *Knowledge Discovery in Database* (KDD). Pendekatan yang biasa digunakan adalah asosiasi. Permasalahannya aturan asosiasi cenderung mengabaikan dataset yang besar. Untuk mengatasi hal tersebut dilakukan klasifikasi barang yang dibeli dan tidak dibeli bersamaan. Algoritma *Self Organizing Map* (SOM) dan *K-Medoids* cocok untuk diterapkan dalam meng*cluster* dataset besar. Penelitian ini menguji kevalidan dan kecepatan algortima *Self Organizing Map* (SOM) dan *K-Medoids* jika dikombinasi dengan *Frequent Pattern-Growth* (*FP-Growth*).

Kata kunci: analisis keranjang pasar, rekomendasi produk, *self organizing map*, *k-medoids*, *fp-growth*.

ABSTRACT

In a business, it takes effort to maximize profits. Among them with promotions. The accuracy of the promotion can be learned from the database a major retail company spending patterns on products commonly purchased together. Information on customer shopping patterns inaccurate cause improper promotion policy and efisien.Salah a common effort to acquire and explore the shopping patterns of customers is using data mining known as Knowledge Discovery in Databases (KDD). The approach used is association. The problem tends to ignore the rules of association of large datasets. To overcome this problem do the classification of goods purchased and not purchased together. Algorithm Self Organizing Map (SOM) and K-Medoids mengcluster suitable to be applied in large datasets. This study tested the validity and speed algorithms Self Organizing Map (SOM) and K-Medoids when combined with Frequent Pattern-Growth (FP-Growth).

Keywords**:** *market basket analysis, recomendation product, promotion product, association rule mining*, *k-medoids*, *fp-growth*.

1. PENDAHULUAN

Sebuah bisnis diperlukan upaya untuk memaksimalkan keuntungan. Keuntungan tersebut akan dapat diperoleh jika perusahaan dapat memanfaatkan potensi dan peluang usaha yang ada. Pelanggan/ konsumen adalah sumber terbesar pendapatan sebuah usaha. Penjual dapat meningkatkan pendapatan dengan menggunakan strategi pemasaran yang tepat dan akurat. Salah satunya adalah menawarkan barang lain yang kemungkinan besar akan dibeli juga oleh pelanggan secara bersamaan dengan barang yang sudah direncanakan untuk dibeli sebelumnya. Permasalahannya adalah barang apakah yang hampir pasti dibeli oleh pelanggan dan dari manakah informasi barang rekomendasi tersebut kita peroleh? Database toko yang dimiliki toko tersebut dapat dianalisis sehingga memeroleh informasi mengenai produk yang

sering dibeli secara bersamaan. Metode yang lazim untuk memperoleh dan menggali pola pelanggan tersebut adalah dengan data mining yang biasa dikenal sebagai *Knowledge Discovery in Database* (KDD). Data mining dapat mengambil informasi yang penting dari database yang besar dan dibutuhkan manusia. Informasi atau pengetahuan tersebut menunjukkan konsep, alur, aturan maupun pola dan bentuk dari sekumpulan data yang berbeda-beda. Data mining dapat diterapkan dalam sebagian besar *field.*Misalnya untuk membangun trend, memprediksi masa depan, dan menganalisis faktor kunci yang dibutuhkan untuk sebuah permasalahan yang kompleks.

Banyak algoritma yang diusulkan untuk menemukan pola asosiasi dan algoritma pertama yang paling efisien untuk menemukan pola asosiasi adalah *apriori* [9][10]. Alasan utamanya karena *Frequent Pattern-Growth* (*FP-Growth*) merupakan algoritma yang dapat menghemat waktu dan media penyimpanan terlebih untuk database yang besar [12].

Algoritma Apriori adalah algoritma *association rule* konvensional, yang membutuhkan waktu yang sangat besar saat proses pemindaian database. Hal ini tentu saja berpengaruh pada efisiensi waktu dalam data mining. Algoritma yang diusulkan untuk mengatasi kelemahan tersebut adalah *Frequent Pattern-Growth* (*FP-Growth*) yang disebut yang tidak perlu untuk menghasilkan *generate candidate item sets* dan dianggap lebih efisien. *Frequent Pattern-Growth* (*FP-Growth*) dibangun dengan membaca kumpulan data satu transaksi pada waktu dan pemetaan setiap transaksi ke dalam bagian *Frequent Pattern-Tree* (*FP-Tree*) [2].

Permasalahan yang muncul selanjutnya dalam aturan asosiasi adalah masalah teknis yang berhubungan dengan teknik rekomendasi yang paling umum dimana aturan asosiasi cenderung mengabaikan *itemset* besar. Akibatnya rekomendasi *item* kurang tepat karena informasi tentang produk ritel tidak tersedia [8]. Untuk mengatasi masalah tersebut, atribut yang ada di*cluster* untuk membentuk kelompok atribut yang sama dan kemudian menentukan pola asosiasi pada masing-masing kelompok *Apriori*, sehingga dapat mempermudah proses mencari rekomendasi produk. Beberapa algoritma *clustering* diantara *K-Means, Fuzzy C-Means, Self Organizing Map* (SOM)*, K-Medoids* [9].

Dalam penelitian ini akan dikomparasikan kombinasi *Self Organizing Map* (SOM) dengan *Frequent Pattern-Growth* (*FP-Growth*), *K-Medoids* dengan *Frequent Pattern-Growth (FP-Growth)* dan *Frequent Pattern-Growth* (*FP-Growth*) yang berdiri sendiri tanpa di *clustering* dengan algoritma apapun sebelumnya. Algoritma *Self Organizing Map* (SOM) dipilih karena *Self Organizing Map* (SOM) merupakan algoritma *clustering* model aglomerative dan partitif selain itu juga dikenal sebagai algoritma non liner, teratur, pemetaan dari input data dimensi tinggi ke dalam *array* dimensi rendah [4][5].. Algoritma *K-Medoids* dipilih sebagai perbandingan untuk mendampingi *Self Organizing Map* (SOM) dan *Frequent Pattern-Growth* (*FP-Growth*). Pada dasarnya, algoritma *clustering* yang paling sering dipakai adalah *K-Means*. Namun, *K-Means* memiliki kekurangan yakni sensitif terhadap *outlier* dalam data dan hal ini bisa dijawab oleh *K-Medoids* yang tidak terpengaruh terhadap *outlier* data sehingga hasil *clustering* menjadi lebih akurat.

1.1 Analisis Pola Belanja Pasar

Membahas tentang pola belanja pelanggan tidak bisa dipisahkan dengan *Assosiation Rule* Mining atau analisis keranjang pasar. *Association Rule Mining* merupakan salah satu metode data mining yang berfokus pada menemukan pola pembelian dengan mengekstraksi asosiasi atau kejadian dari data transaksional sebuah toko. Analisis keranjang pasar bermula dari transaksi-transaksi yang berisi satu atau lebih barang, dan beberapa informasi sementara dari transaksi tersebut. Untuk melakukan analisis keranjang pasar, berikut langkah-langkah:

1) Tentukan nilai *Minimum Support* yang diinginkan. *Minimum Support* merupakan ambang batas minimum jumlah *itemset* yang diperbolehkan, jika jumlah *item*nya di bawah ambang batas maka *item* tersebut akan dieliminasi.
2) Menetapkan *frequent itemset* (kumpulan *item* yang muncul secara bersamaan), dengan cara mengambil *itemset* yang memiliki frekuensi *itemset* minimal sebesar *Minimum Support* sebelumnya.
3) Dari semua *frequent itemset*, hasilkan aturan asosiasi yang memenuhi nilai *Minimum Support.*

Analisis keranjang pasar didasarkan pada tiga matrik: *Support*, *Confidence* dan *Lift*. Ketiga matrik tersebut berasal dari catatan transaksi untuk bisnis [1].

Support

Matrik pertama ditetapkan untuk analisis keranjang pasar adalah *Support*, yang merupakan probabilitas dari asosiasi (probabilitas dari dua *item* yang dibeli bersama-sama). *Support* dihasilkan dari berapa kali jumlah *item* A dan B terjadi bersamaan dalam transaksi yang sama dibagi dengan jumlah total dari transaksi tersebut. Untuk menghitung nilai *Support* digunakan persamaan (1) sebagai berikut:

$$Support = P(A \cap B) = \frac{\text{Jumlah transaksi yang memuat A dan B}}{\text{total Jumlah transaksi}} \quad (1)$$

Confidence

Confidence dihasilkan dari seberapa kuat hubungan produk yang sudah dibeli.
Untuk menhitung nilai *Confidence* digunakan persamaan (2) sebagai berikut:

$$Confidence = P(B/A) = \frac{Support(A \cap B)}{P(A)} \quad (2)$$

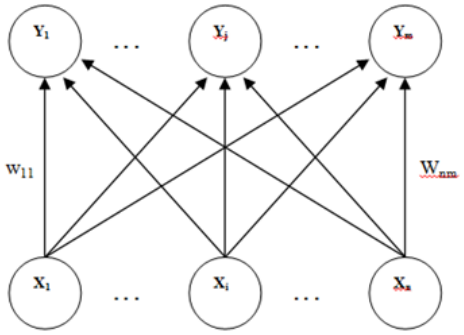
Lift

Lift Ratio mengukur seberapa penting *rule* yang teleh terbentuk berdasarkan nilai *support* dan *confidence*. *Lift Ratio* merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar produk A dibeli bersamaan dengan produk B.
Untuk menghitung nilai *Lift Ratio* digunakan persamaan (3) sebagai berikut:

$$Lift\ Ratio = \frac{Support(A \cap B)}{Support(A) * support(B)} \quad (3)$$

1.2 Algoritma Self Organizing Map (SOM)

Self Organizing Map (SOM) merupakan salah satu metode dalam Jaringan Syaraf Tiruan (*Neural Network*) yang menggunakan pembelajaran tanpa pengarahan (*unsupervised learning*). Metode pembelajaran tanpa pengarahan tidak memerlukan target, selama proses pembelajaran unit – unit input yang hampir sama dikelompokkan dalam kelompok tertentu yang berada di unit output (*cluster units*). Arsitektur *Self Organizing Map* (SOM) terdiri dari 1 lapisan input dan 1 lapisan output. Setiap unit pada lapisan input dihubungkan dengan semua unit di lapisan output dengan suatu bobot keterhubungan yang ditunjukkan pada Gambar 1.



Gambar 1. Algoritma *Self Organizing Map* (SOM)

1.3 Algoritma Frequent Pattern-Growth (FP-Growth)

Algoritma yang biasa dipakai dalam mencari *frequent item set* antara lain algoritma Apriori dan algoritma *Frequent Pattern-Growth* (*FP-Growth*). Pada penelitian ini akan dibahas bagaimana pencarian *frequentitemset* mengunakan algoritma *Frequent Pattern-Growth* (*FP-Growth*). *Frequent Pattern Tree* (*FP-Tree*) digunakan bersamaan dengan algoritma *Frequent Pattern-Growth* (*FP-Growth*) untuk menentukan *frequent item set* (data yang paling sering muncul) dari sebuah *dataset*. Algoritma *Frequent Pattern-Growth* (*FP-Growth*) adalah salah satu cara alternatif untuk menemukan himpunan data yang paling sering muncul tanpa menggunakan generasi kandidat [14]. *Frequent Pattern-Growth* (*FP-Growth*) membangun konstruksi data *Frequent Pattern Tree* (*FP-Tree*) yang sangat dikompresi, dan mengurangi data asli. Algoritma *Frequent Pattern-Growth* (*FP-Growth*) melakukan *scan* database yang sama sebanyak dua kali. *Scanning* database yang pertama, kita dapat memperoleh *frequent 1-item-set,* dan *scanning* database yang kedua, kita dapat memfilter database *non-frequent item,* selebihnya, *Frequent Pattern Tree* (*FP-Tree*) dihasilkan secara bersamaan. Akhirnya, dapat diperoleh aturan asosiasi dengan menggunakan *Frequent Pattern Tree* (*FP-Tree*) [13].

Kelemahan *Apriori* yang selalu melakukan *scanning* database secara berulang-ulang membuat *Apriori* ini kurang efektif. Berbeda dengan *Frequent Pattern Tree* (*FP-Tree*) yang digunakan bersamaan dengan algoritma *Frequent Pattern-Growth* (*FP-Growth*) yang hanya memerlukan dua kali scanning database untuk membuat *frequent item set*. Dengan menggunakan *Frequent Pattern Tree* (*FP-Tree*), algoritma *Frequent Pattern-Growth* (*FP-Growth*) dapat langsung mengekstrak *frequent Item set* dari

Frequent Pattern Tree (*FP-Tree*) yang telah terbentuk. *Frequent Pattern Tree* (*FP-Tree*) didefinisikan sebagai berikut [14] :

Sebuah *root* yang diberi label *null*, sekumpulan *sub-tree* yang beranggotakan *item-item* tertentu, dan sebuah tabel *frequent header*. Setiap simpul dalam *Frequent Pattern Tree* (*FP-Tree*) mengandung tiga *field*, yaitu:

1) *Item-name*: menginformasikan *item* yang dipresentasikan oleh simpul tersebut.
2) *Count*: mempresentasikan jumlah transaksi yang melewati simpul tersebut.
3) *Node-Link*: penghubung yang menghubungkan simpul-simpul dengan *item-name* yang sama, atau *null* jika kosong.

1.4 Algoritma K-Medoids

Untuk melakukan *clustering* dengan metode partisi dapat menggunakan *K-Means* dan *K-Medoids*. *K-Means* merupakan suatu algoritma peng*cluster*an yang cukup sederhana yang mempartisi *dataset* kedalam beberapa *cluster* k. Algoritmanya cukup mudah untuk diimplementasi dan dijalankan, relatif cepat, mudah disesuaikan dan banyak digunakan [14]. Kelemahan-kelemahan dari algoritma *K-Means* yaitu [20] :

1) Ketika jumlah data yang tidak begitu banyak, pengelompokan awal akan menentukan *cluster* secara signifikan.
2) Jumlah *cluster* K harus ditentukan terlebih dahulu.
3) *Cluster* yang asli tidak diketahui, dengan menggunakan data yang sama, jika dimasukkan dalam urutan yang berbeda dapat menghasilkan *cluster* yang berbeda jika jumlah data sedikit.
4) Kelemahan dari aritmatika mean tidak kuat untuk *outlier*, sangat jauh data dari *centroid* memungkinkan mempengaruhi *centroid* yang asli.

Algortima *K-Medoids*, juga dikenal sebagai *partitioning around Medoids*, adalah varian dari metode *K-Means*. Hal ini didasarkan pada penggunaan *Medoids* bukan dari pengamatan *mean* yang dimiliki oleh setiap *cluster,* dengan tujuan mengurangi sensitivitas dari partisi yang dihasilkan sehubungan dengan nilai-nilai ekstrim yang ada dalam *dataset* [3]. Algoritma *K-Medoids* hadir untuk mengatasi kelemahan Algoritma *K-Means* yang sensitif terhadap *outlier* karena suatu objek dengan suatu nilai yang besar mungkin secara substansial menyimpang dari distribusi data [6].

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan jenis penelitian eksperimen, dengan melakukan ujicoba terhadap dataset yang telah dikumpulkan. Tahapan-tahapan penelitian ini adalah sebagai berikut:

1) Pengumpulan Data
   Pengumpulan data merupakan langkah awal pada suatu penelitian. Data yang digunakan pada penelitian ini adalah data transaksi penjualan Dr. Tariq Mahmood.
2) Pengolahan Data Awal
   Pengolahan awal (*Pre processing*) merupakan tahap untuk mempersiapkan data yang telah diperoleh dari tahap pengumpulan data, yang akan digunakan pada tahap selanjutnya. Pada tahap ini peneliti mengubah format data tanpa mengubah isi data agar mudah untuk diolah menggunakan aplikasi rapidminer.
3) Desain Eksperimen
   Tahapan ini akan membahas alurdesain eksperimen yang digunakan pada penelitian supaya prosesnya sistematis dan menghasilkan temuan yang valid.
4) Eksperimen dan Pengujian
   Tahapan ini akan mengeksekusi tahapan penelitian dengan teknik pengujian yang akan telah dipersiapkan sebelumnya.
5) Evaluasi Penelitian
   Tahapan ini akan membahas hasil evaluasi dari eksperimen yang telah digunakan.

Dataset ini merangkum 304 atribut dimana 303 diantaranya merupakan item barang yang dibeli. Salah satu datanya menunjukkan ID transaksi penjualan yang dilakukan dalam sebuah supermarket. Terdapat 1362 record transaksi dalam dataset ditunjukkan pada Gambar 2.

	A	G	H	I	J	K	L	M	N	O	P	Q
1	Basket ID	Sugar Cookies	Onions	Deli Ham	Dishwasher Detergent	Beets	40 Watt Lightbulb	Ice Cream	Cottage Cheese	Plain English Muffins	Strawberry Soda	Vanilla Ice Cream
2	C11867	false	false	false	false	false	false	false	false	false	false	false
3	C5096	false	false	false	false	false	false	false	false	false	false	false
4	C4295	false	false	false	false	false	false	false	false	false	false	false
5	C2837	false	false	false	false	false	false	false	false	false	false	false
6	C2693	false	false	false	false	false	false	false	false	false	false	false
7	C3497	false	false	false	false	false	false	false	false	false	false	false
8	C2696	false	false	false	false	false	false	false	false	false	false	false
9	C1895	false	false	false	false	false	false	false	false	false	false	false
10	C9524	false	false	false	false	false	false	false	false	false	false	false
11	C9527	false	false	false	false	false	false	false	false	false	false	false
12	C8726	false	false	false	false	false	false	false	false	false	false	false
13	C9383	false	false	false	false	false	false	false	false	false	false	false
14	C8582	true	true	false	false	false	false	true	false	false	false	false
15	C10133	false	true	false	false	false	false	false	false	false	false	false
16	C6983	false	false	false	false	false	false	false	false	false	false	false
17	C3641	false	false	false	false	false	false	false	false	false	false	false
18	C5246	false	false	false	false	false	false	false	false	false	false	false
19	C10136	false	false	false	false	false	false	false	false	false	false	false
20	C7787	false	false	false	false	false	false	false	false	false	false	false
21	C3644	false	false	false	false	false	false	false	false	false	false	false

Gambar 2. Data Transaksi Penjualan

Keterangan:

1) Basket ID seperti C11867, C5096, C4295, C2837, C2693, C2696, C1895, C9524 merupakan ID Transaksi penjualan barang.
2) Sugar Cookies, Onions, Delli Ham, Dishwasher Detergent, Beets, 40 Watt Lightbulb, Ice Cream, Cottage Cheese, Plain English Muffins, Strawberry Soda, Vanila Ice Cream dll merupakan barang-barang yang dijual di Supermarket.
3) False menandakan bahwa barang tersebut tidak dibeli oleh pembeli.
4) True menandakan bahwa barang tersebut dibeli oleh pembeli.

2.1 Pengolahan Data Awal

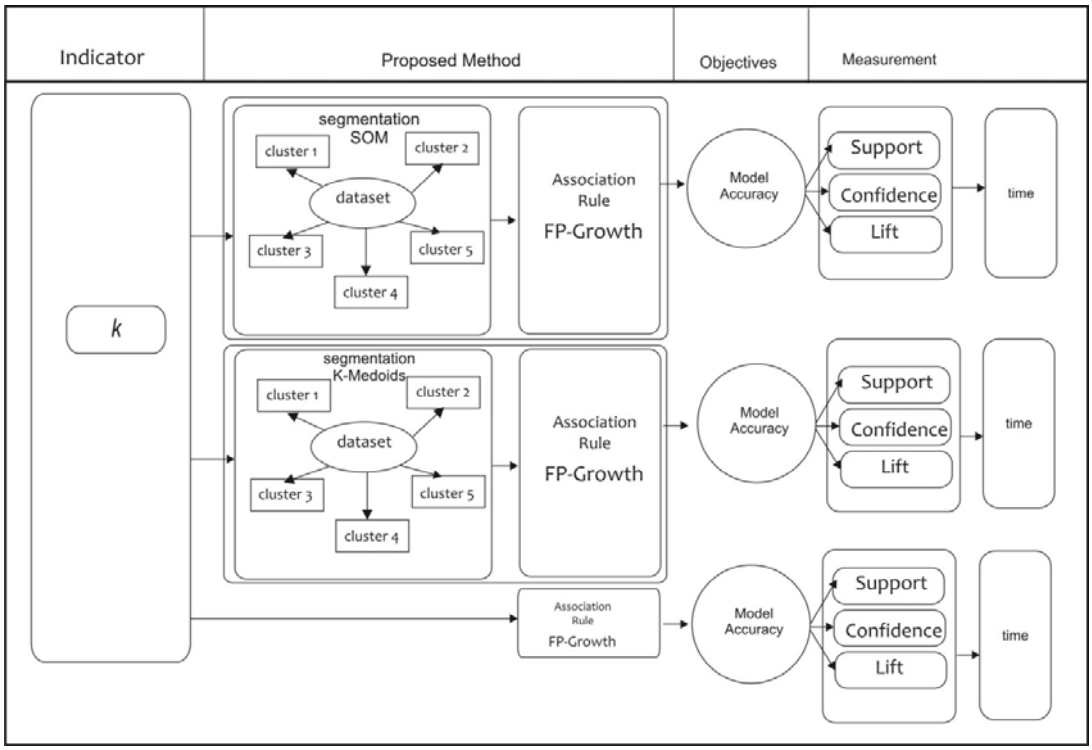
Data akan pada tahap pengolahan data awal diubah menjadi dataset yang sesuai dengan karakteristik algoritma yang akan dilakukan ujicoba. Diantaranya, menyesuaikan format *true* dan *false* dalam dataset menjadi data numerik 1 untuk *true* dan 0 untuk *false,* agar dapat digunakan dalam proses asosiasi. Tahapan selanjutnya melakukan *clustering* data tersebut tanpa menambah atribut apapun karena *Self Organizing Map* (SOM) akan meng*cluster* tersendiri. Sedangkan untuk *K-Medoids* ditentukan $k = 5$

	Basket ID	Hair Cond	Lemons	Standard	Frozen Ch	98pct. Fat	Sugar Coc	Onions	Deli Ham	Dishwash	Beets
2	C11867	0	0	0	0	0	0	0	0	0	0
3	C5096	0	0	0	0	0	0	0	0	0	0
4	C4295	0	0	0	0	0	0	0	0	0	0
5	C2837	0	0	0	0	0	0	0	0	0	0
6	C2693	0	0	0	0	0	0	0	0	0	0
7	C3497	0	0	0	0	0	0	0	0	0	0
8	C2696	0	0	0	0	0	0	0	0	0	0
9	C1895	0	0	0	0	1	0	0	0	0	0
10	C9524	0	0	0	0	0	0	0	0	0	0
11	C9527	0	0	0	0	0	0	0	0	0	0
12	C8726	0	0	0	0	0	0	0	0	0	0
13	C9383	0	0	0	0	0	0	0	0	0	0
14	C8582	0	0	0	0	0	1	1	0	0	0
15	C10133	0	0	0	0	1	0	1	0	0	0
16	C6983	0	0	0	0	0	0	0	0	0	0
17	C3641	1	0	0	0	0	0	0	0	0	0
18	C5246	0	0	0	0	0	0	0	0	0	0
19	C10136	0	0	0	0	0	0	0	0	0	0

Gambar 3. Dataset Setelah Diubah Menjadi Numerik

2.2 Desain Eksperimen

Eksperimen pada penelitian ini adalah menggunakan *Self Organizing Map* (SOM) dan *Frequent Pattern-Growth* (*FP-Growth*), kemudian dengan data yang sama menerapkan algoritma *K-Medoids* dan *Frequent Pattern-Growth* (*FP-Growth*) serta mengujinya juga dataset langsung diasosiasi dengan *Frequent Pattern-Growth* (*FP-Growth*) tanpa diawali dengan *clustering* terlebih dahulu. Desain untuk eksperimen ini tunjukkan pada gambar 4.



Gambar 4. Desain Eksperimen Penggabungan Algoritma

Eksperimen ini diawali dengan menerapkan algoritma *Self Organizing Map* (SOM) untuk *clustering* dan algoritma *Frequent Pattern-Growth* (*FP-Growth*) untuk pendekatan asosiasi. Dalam Algoritma *Self Organizing Map* (SOM) dataset di*cluster* secara alamiah tanpa ditentukan dulu jumlahnya. Proses

clustering dilakukan supaya data yang diasosiasi menjadi lebih kecil sehingga pola yang diperoleh menjadi lebih akurat. Tiap *cluster* yang terbentuk kemudian diasosiasi dengan *Frequent Pattern-Growth* (*FP-Growth*) untuk membuat pola belanja pelanggan. Dalam proses asosiasi ini akan diukur *support, confidence*, dan *lift ratio.* Setelah itu, dicatat waktu total yang dibutuhkan dari masing-masing tahapan. Eksperimen diulangi dengan mengganti algoritma *clustering Self Organizing Map* (SOM) dengan *K-Medoids* dan proses asosiasi tetap menggunakan *Self Organizing Map* (SOM). Jumlah *cluster* yang dibuat yakni 5 c*luster*. Kemudian masing-masing *cluster* di asosiasi dengan algoritma *Self Organizing Map* (SOM) untuk diukur *support, confidence*, dan *lift ratio.* Kemudian waktu proses masing-masing tahapan dicatat. Akurasi dan total waktu proses *clustering* hingga asosiasi dari tahapan 1 yakni *Self Organizing Map* (SOM) + *Frequent Pattern-Growth* (*FP-Growth*) dibandingkan dengan tahapan 2 yakni *K-Medoids* + *Frequent Pattern-Growth* (*FP-Growth*) kemudian dibandingkan dengan akurasi dan waktu yang diperlukan jika menggunakan *Frequent Pattern-Growth* (*FP-Growth*) saja. Kemudian disimpulkan diantara ketiganya terkait akurasi dan efisiensi waktunya.

2.3 Evaluasi Penelitian

Hasil akurasi untuk menentukan rekomendasi produk kepada pelanggan yang didapat dari penelitian ini dapat diukur dengan menggunakan *Lift Ratio*. *Lift Ratio* merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar produk A dibeli bersamaan dengan produk B. Sebuah transaksi dikatakan valid jika mempunyai nilai *Lift Ratio* lebih dari 1, yang berarti bahwa dalam transaksi tersebut, produk A dan B benar-benar dibeli secara bersamaan. *Lift Ratio* mengukur seberapa penting *rule* yang telah terbentuk berdasarkan nilai *support* dan *confidence*. Evaluasi rekomendasi produk untuk pelanggan ditunjukkan di gambar 5.

No.	Premises	Conclusion	Support	Confidence	Lift
1	Whole Corn	Hair Conditioner	0.970	0.976	1.006
2	Orange Flavored Fruit Bars	Hair Conditioner	0.970	0.976	1.006
3	Whole Corn	Orange Flavored Fruit Bars, Hair Conditioner	0.970	0.976	1.006
4	Orange Flavored Fruit Bars	Whole Corn, Hair Conditioner	0.970	0.976	1.006
5	Whole Corn, Orange Flavored Fruit Bars	Hair Conditioner	0.970	0.976	1.006
6	Whole Corn	Orange Flavored Fruit Bars	0.994	1	1.006
7	Orange Flavored Fruit Bars	Whole Corn	0.994	1	1.006
8	Hair Conditioner	Whole Corn	0.970	1	1.006
9	Hair Conditioner	Orange Flavored Fruit Bars	0.970	1	1.006
10	Hair Conditioner	Whole Corn, Orange Flavored Fruit Bars	0.970	1	1.006
11	Whole Corn, Hair Conditioner	Orange Flavored Fruit Bars	0.970	1	1.006
12	Orange Flavored Fruit Bars, Hair Conditioner	Whole Corn	0.970	1	1.006

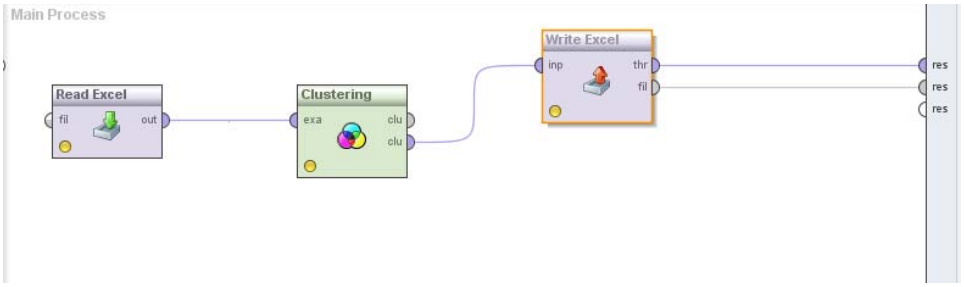
Gambar 5. Evaluasi Rekomendasi Produk Dengan *Support, Confidence, Lift*

Setelah ditemukan *Support, Confidence, Lift* kemudian dicatat pada masing-masing proses untuk diketahui seberapa efisien waktu yang diperlukan dari pengolahan dataset hingga menjadi sebuah pola belanja pelanggan. Kemudian akan disimpulkan tahapan mana yang memiliki tingkat kevalidan paling baik dan tahapan mana yang memiliki waktu proses paling efisien.

3. HASIL DAN PEMBAHASAN

3.1 Clustering Data dengan K-Medoids

Dalam *k-medoids* kita tentukan menjadi *5 cluster* (k). Penentuan $k = 5$ ini dilakukan supaya lebih mudah didapatkan hasilnya karena keterbatasan kemampuan PC. Hasil dari proses *clustering* tersebut akan di eksport dalam bentuk file terpisah untuk masing-masing hasil *cluster*. Proses *clustering* data ditunjukkan pada gambar 6, dan hasil *clustering* data ditunjukkan pada gambar 7.



Gambar 6. Proses *Clustering* Data

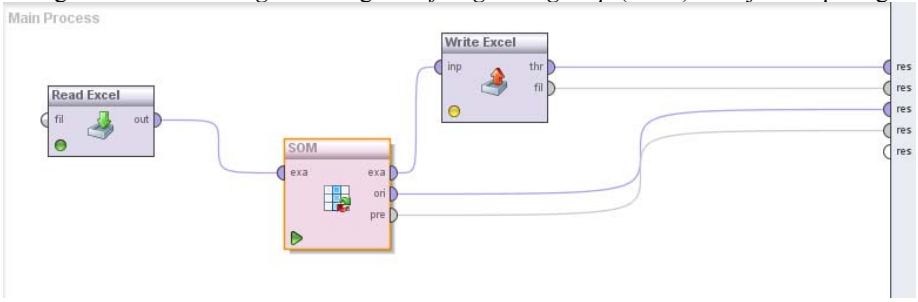
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Basket ID	Cluster	ir Conditio	Lemons	Standard	Frozen Ch	98pct. Fat	Sugar Coc	Onions	Deli Ham	Dishwash	Beets	40 Watt Li	Ice Cream	Cottage C	Plain Engl	Strawberi	Vanilla Ic
2	Basket ID	cluster_0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
3	C5252	cluster_0	1	1	0	0	1	0	1	0	1	0	0	1	1	0	0	0
4	C3659	cluster_0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
5	C4478	cluster_0	1	1	0	0	1	0	1	0	0	1	0	0	1	0	0	0
6	C887	cluster_0	0	0	0	1	0	1	1	0	1	0	0	0	1	0	0	0
7	C7995	cluster_0	0	0	1	0	0	0	0	1	1	0	1	0	0	1	1	1
8	C4694	cluster_0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
9	C8987	cluster_0	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0
10	C2177	cluster_0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0
11	C2543	cluster_0	1	1	0	1	0	1	1	0	0	0	1	1	0	0	0	0
12	C9233	cluster_0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
13	C7493	cluster_0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
14	C7646	cluster_0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
15	C1784	cluster_0	0	0	1	0	1	1	0	0	0	1	0	0	1	0	0	0
16	C5303	cluster_0	1	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0
17	C9623	cluster_0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
18	C959	cluster_0	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0	0
19	C5501	cluster_0	0	1	1	0	0	1	0	0	1	0	0	0	1	0	0	0
20	C4706	cluster_0	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0
21	C12002	cluster_0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
22	C3770	cluster_0	1	1	0	0	1	1	1	0	0	0	0	0	0	0	1	0
23	C3788	cluster_0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
24	C6347	cluster_0	1	1	0	0	1	1	1	0	0	1	0	1	0	0	0	0

Gambar 7. Hasil *Clustering* Data

Setelah hasil *cluster* didapat, yakni *cluster* 0, *cluster* 1, *cluster* 2, *cluster* 3, *cluster* 4 kemudian diurutkan/ sorting dan dipisahkan menjadi 5 file untuk kemudian di lakukan proses asosiasi dengan *Frequent Pattern-Growth* (*FP-Growth*). Hasil dari c*luster* menggunakan *K-Medoids*:
Cluster 0: 32 items
Cluster 1: 2 items
Cluster 2: 2 items
Cluster 3: 2 items
Cluster 4: 1323 items
Total number of items: 1361
Selama 28,20menit

3.2 Clustering Data dengan Self Organizing Map (SOM)

Rancangan Proses *clustering* data dengan *Self Organizing Map* (SOM) ditunjukkan pada gambar 8.



Gambar 8. Rancangan *Cluster* Data dengan *Self Organizing Map* (SOM)

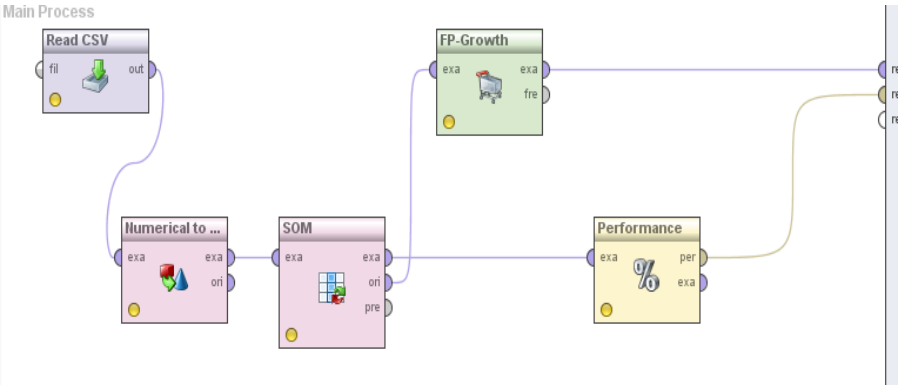
Cluster data dilakukan dengan *Self Organizing Map* (SOM) terbagi menjadi 2, dari rancangan *cluster* data tersebut didapatkan hasil yang ditunjukkan pada gambar 9.

Row No.	SOM_0	SOM_1
1	10	24
2	10	24
3	10	24
4	10	24
5	10	24
6	10	24
7	10	24
8	10	24
9	10	24
10	10	24
11	10	24
12	10	24
13	10	24
14	10	24
15	10	24

Gambar 9. Rancangan *Cluster* Data dengan *Self Organizing Map* (SOM)

3.3 Asosiasi Cluster Data Penjualan

Setiap *cluster* akan dilakukan asosiasi dengan *Frequent Pattern-Growth* (*FP-Growth*) untuk menghasilkan rekomendasi produk. Proses asosiasi *cluster* ditunjukkan pada gambar 10.



Gambar 10. Hasil Asosiasi *Cluster* Data Penjualan

Setelah proses *clustering* dilakukan, didapatkan 5 *cluster*. *Cluster* yang terbentuk dijadikan sebagai dataset baru yang lebih kecil daripada dataset sebelumnya. Kemudian setiap *cluster* akan dilakukan pendekatan asosiasi menggunakan algoritma *Frequent Pattern-Growth* (*FP-Growth*) untuk menganalisis keranjang pasar. Dengan dataset yang lebih kecil tujuan untuk meningkatkan akurasi dalam rekomendasi produk, sedangkan untuk perbandingan digunakan algoritma apriori.

4. KÉSIMPULAN

Hasil akurasi untuk menentukan rekomendasi produk kepada pelanggan yang didapat dari penelitian ini dapat diukur dengan menggunakan *Lift Ratio*. *Lift Ratio* merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar produk A dibeli bersamaan dengan produk B. Sebuah transaksi dikatakan valid jika mempunyai nilai *Lift Ratio* lebih dari 1, yang berarti bahwa dalam transaksi tersebut, produk A dan B benar-benar dibeli secara bersamaan [6]. *Lift Ratio* mengukur seberapa penting *rule* yang telah terbentuk berdasarkan nilai *support* dan *confidence*. Untuk eksperimen ditentukan nilai *Minimum Support* adalah 10% - 100% dan nilai *Minimum Confidence* 10% - 100%.

DAFTAR PUSTAKA

[1] B. Hoanca, "Using Market Basket Analysis to Estimate Potential Revenue Increases for a Small University Bookstore," *Information Systems*, vol. 1822, 2011, pp. 1-11.
[2] C. Wang, W. Lee, and C. Pang, "Applying Fuzzy *FP-Growth* to Mine Fuzzy Association Rules," *Engineering and Technology*, 2010.
[3] C. Vercellis, *Business Intelligence: Data Mining and Optimization for Decision Making*, Milan: WILEY, 2009.
[4] Endy, Charles Lim, Kho I Eng, AntoSatriyoNugroho, 2010, *Implementation of intelligent searchingUsing self-organizing map for webmining used In document containing information in Relation to cyber terrorism,Second International Conference on Advances in Computing, Control, and Telecommunication Technologies*.
[5] Juha Vesanto and Esa Alhoniemi,Clustering of the Self-Organizing Map, IEEE Transactions on *Neural Networks*, 2000.
[6] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2006.
[7] K. Christidis, D. Apostolou, and G. Mentzas, "Exploring Customer Preferences with Probabilistic Topics Models," Work, 2010, pp. 1-13.
[8] L. Gordon, "Leading Practices in Market Basket Analysis," *Strategy*, vol. 94022, 2008.
[9] M. Plasse, N. Niang, G. Saporta, A. Villeminot, and L. Leblond, "Combined use of association rules mining and clustering methods to find relevant links between binary rare attributes in a large data set," Computational Statistics & Data Analysis.
[10] M. Ykhlef, "A Quantum Swarm Evolutionary Algorithm for mining association rules in large databases," *Journal of King Saud University - Computer and Information Sciences*, vol. 23, 2011, pp. 1-6.
[11] S. Guglani, G. Gupta, and M. Dhanda, "Way to Improve K-Means Algorithm by Using Various Attributes," (IJAEST) International Journal of Advanced Engineering Sciences and Technologies, vol. 7, 2011, pp. 330 - 336.
[12] Yongmei Liu and Yong Guan, "*FP-Growth* Algoritm For Appliaction in Research of Market Basket Analysis,"2008, Capital Normal University, Beijing, China 2008.
[13] Y. Liu and Y. Guan, "*FP-Growth* Algorithm for Application in Research of Market BasketAnalysis," *Design*, 2008.
[14] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. Mclachlan, A. Ng.

Jurnal Sistem Informasi (Journal of Information Systems). 2/13 (2017), 118-124
DOI: http://dx.doi.org/10.21609/jsi.v13i2.551

IMPLEMENTASI ALGORITMA *FP-GROWTH* DALAM PENENTUAN POLA HUBUNGAN KECELAKAAN LALU LINTAS

Rizky Fitria[1], Warnia Nengsih[2], dan Dini Hidayatul Qudsi[3]

[1,2,3] Program Studi Sistem Informasi, Politeknik Caltex Riau, Jl. Umban Sari (Patin) No. 1 Rumbai, Pekanbaru, Riau, 28265, Indonesia

Email: [1]rizky13si@mahasiswa.pcr.ac.id, [2]warnia@pcr.ac.id, [3]dinihq@pcr.ac.id

Abstrak

Tingginya jumlah kecelakaan yang terjadi menjadi alasan penting bagi pihak Satuan Lalu Lintas, salah satunya Polresta Pekanbaru untuk mengetahui hubungan antara factor-faktor penyebab kecelakaan. Hubungan tersebut dipelajari dengan menggunakan teknik *data mining*. Teknik ini bertujuan untuk menemukan informasi berupa pola yang dapat menjadi acuan dalam pengambilan kebijakan di Satuan Lalu Lintas Polresta Pekanbaru. Teknik *data mining* yang digunakan yaitu teknik *association rule* dengan Algoritma *Fp-growth*. Algoritma ini menerapkan struktur *data tree* untuk mengetahui pola kecelakaan lalu lintas. Pola tersebut ditentukan oleh dua parameter, yaitu *support* (nilai penunjang) dan *confidence* (nilai kepastian). Sistem dibuat berbasis *desktop* dengan bahasa pemograman Visual Basic.Net. Sistem ini menghasilkan pola kecelakaan yang sering terjadi. Berdasarkan pengujian *lift ratio,* didapatkan pola kecelakaan yang paling sering terjadi, yaitu faktor kecelakaan dengan jenis_luka adalah luka ringan, jenis_jalan adalah jalan arteri, waktu adalah padat kendaraan, dan jenis_kelamin adalah perempuan dengan nilai *lift ratio* 1.20%. Sedangkan hasil pengujian skala Likert didapatkan bahwa 88.09% pengguna merasa dimudahkan dengan sistem ini dan dapat membantu mereka dalam mengetahui pola kecelakaan yang sering terjadi.

Kata kunci: *kecelakaan lalu lintas, associattion rule, Fp-growth, visual basic.net*

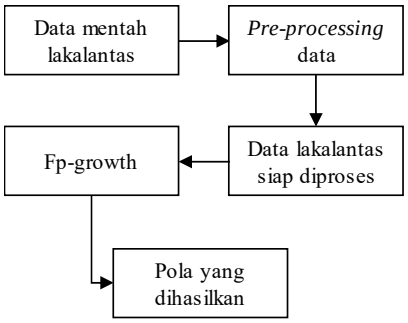
Abstract

The high number of accidents that happened is an important reason for the Traffic Unit, such as for Polresta Pekanbaru to determine the relationship between the factors causing the accident. The relationship is studied using data *mining* technique. This technique aims to find information in the form of patterns that can be a reference for Pekanbaru Police Traffic Unit in making policy. Data *mining* technique used in this research is association *rule* technique with *Fp-growth* Algorithm. This algorithm applies data *tree* structure to learn the traffic accident pattern. The pattern is determined by two parameters, namely support (the value of support) and confidence (the value of certainty). The system is developed as a desktop-based system using Visual Basic.Net programming language. This system generates *frequent* crash patterns. Based on the lift ratio test, the most *frequent* accident pattern, ie accident factor with *jenis_luka* is minor injury, *jenis_jalan* is arterial road, waktu is high density of vehicles, and *jenis_kelamin* is women with lift ratio value 1.20%. While Likert scale test results obtained that 88.09% of users feel facilitated with this system that it can help them in knowing the pattern of accidents that often occur.

Keywords: *traffic accident, association rule, Fp-growth, visual basic.net*

1. Pendahuluan

Kecelakaan lalu lintas (lakalantas) adalah suatu peristiwa yang tidak diduga dan tidak disengaja yang melibatkan kendaraan atau pengguna jalan lain yang mengakibatkan korban manusia dan kerugian harta benda [1]. Hal ini dapat kita lihat dalam kehidupan sehari-hari, terjadinya lakalantas mengakibatkan kerugian yang sangat besar dan bahkan dapat menghilangkan nyawa seseorang. Kejadian ini menjadi hal yang mengerikan yang terjadi di beberapa negara yang umumnya adalah negara-negara berkembang, dimana urusan transportasi masih dalam tahap pembangunan dan perbaikan. Pada saat ini, India merupakan negara dengan jumlah kematian terbanyak akibat kecelakaan lalu lintas, sedangkan Indonesia menempati urutan kelima (berdasarkan data WHO tahun 2011 [2]). Sementara, menurut Badan Intelijen Negara, kecelakaan lalu lintas di Indonesia menjadi pembunuh terbesar ketiga, dibawah penyakit jantung koroner dan TBC [3]. Bahkan, kecelakaan lalu lintas menjadi penyebab



Gambar 1. Diagram blok sistem secara garis besar

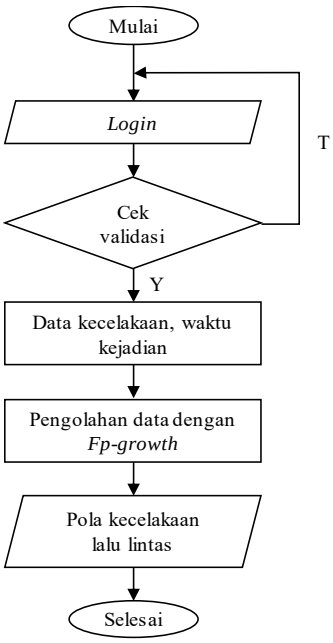
utama kematian anak-anak di dunia, dengan rentang usia 10-24 tahun. Terdapat sekitar 400.000 korban dengan usia dibawah 25 tahun yang meninggal di jalan raya, dengan rata rata angka kematian 1000 orang setiap harinya.

Untuk mendapatkan pola kecelakaan lalu lintas, diperlukan suatu teknik *data mining* yaitu teknik *association rule* untuk menemukan pola hubungan ketekaitan suatu *item*. Algoritma yang digunakan dari teknik *association rule* ini adalah adalah algoritma *Fp-growth. Fp-growth* adalah sebuah metode dalam *data mining* untuk mencari *frequent itemset* tanpa menggunakan *candidate generation. Fp-growth* ini merupakan pengembangan dari algoritma apriori. Kekurangan dari algoritma apriori diperbaiki dengan menghilangkan *candidate generation,* karena *Fp-growth* menggunakan konsep pembangunan *tree* dalam pencarian *frequent itemset.* Hal tersebut membuat algoritma *Fp-growth* lebih cepat dibandingkan algoritma apriori.

Pada penelitian ini, peneliti telah menganalisis hubungan kecelakaan lalu lintas di Kota Pekanbaru dengan variabel-variabel tertentu. Variabel yang akan diteliti antara lain yaitu waktu kejadian, jenis pekerjaan, usia, jenis kelamin, jenis kecelakaan dan jenis luka, jenis kendaraan dan jenis jalan. Penelitian ini menggunakan teknik *Association Rule* yaitu *Fp-growth* untuk mengetahui pola yang terbentuk dari faktor-faktor penyebab terjadinya kecelakaan. Penelitian ini diharapkan dapat membantu sistem yang ada untuk menghasilkan pengetahuan baru yang lebih bermanfaat bagi pihak Satuan Lalu Lintas Polresta Pekanbaru.

Penelitian Terdahulu

Penelitian yang dilakukan oleh [4] membahas tentang implementasi algoritma *Fp-growth* untuk menganalisis data transaksi yang terjadi disebuah toko kue yaitu L'Cheese pekanbaru. Pada pengimplementasiannya, peneliti menerapkan algoritma *Fp-growth* untuk menganalisis pola pembelian pelanggan dengan menganalisis penjualannya. Hasilnya



Gambar 2. Alur kerja sistem secara umum

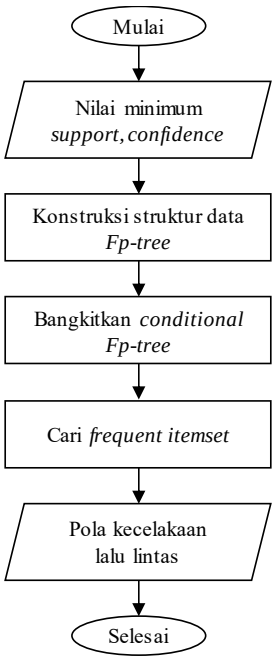
berupa *item* yang paling banyak terjual dan keterkaitan antara dua *item* atau lebih yang nantinya akan berdampak pada penjualan kue tersebut.

Penelitian yang dilakukan oleh [5] menganalisis terhadap penjualan sepeda motor untuk mengetahui sepeda motor mana yang lebih banyak dijual, dan berapa banyak persediaan yang diperlukan perusahaan untuk menyediakan sepeda motor. Untuk mengimplementasikan algoritma *Fp-growth* pada penelitian ini, dilihat dari banyaknya *item* yang terjual, lalu membuat rangkaian *tree* dengan *fp-tree* untuk mengetahui banyaknya *frequent itemset* yang terjadi.

Peneliti [6] melakukan penelitian untuk mengetahui pola hubungan yang sering terbentuk dari faktor-faktor penyebab terjadinya kecelakaan di jalan raya khususnya pada Kabupaten Sleman. Penelitian ini menggunakan metode *association rule* algoritma *apriori.* Faktor yang diambil dalam penelitian ini adalah waktu, profesi, usia, jenis kelamin, dan Surat Izin Mengemudi (SIM).

2. Method

Penelitian yang dibahas pada artikel ini merupakan penelitian untuk membangun sebuah aplikasi *desktop* yang akan menganalisis pola hubungan kecelakaan lalu lintas menggunakan metode *association rule* yaitu algoritma *Fp-growth.* Faktor yang diambil dalam penelitian ini adalah waktu kejadian, jenis pekerjaan, usia, jenis kelamin, jenis kecelakaan, jenis luka, jenis kendaraan dan jenis jalan.



Gambar 3. Cara kerja dari proses algoritma *fp-growth*

Teknik *Association Rule*

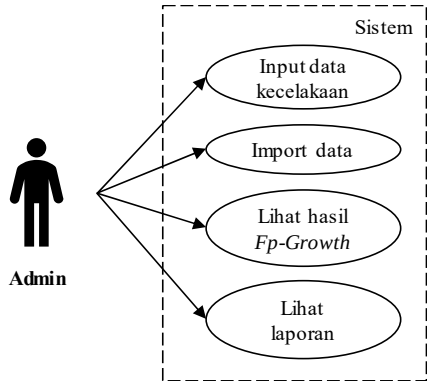
Analisis asosiasi adalah teknik *data mining* untuk menemukan aturan asosiatif antara kombinasi suatu *item*. Contoh aturan asosiatif dari analisis pembelian di suatu pasar swalayan adalah seperti berapa besar kemungkinan seorang pelanggan membeli roti bersamaan dengan susu [4]. *Association Rule* ditentukan oleh dua parameter, yaitu: 1) *Support* adalah suatu ukuran yang menunjukkan seberapa besar tingkat dominasi/persentase suatu kombinasi *item* dari keseluruhan transaksi, 2) *Confidence* adalah suatu ukuran yang menunjukkan kuatnya hubungan antar *item* dalam aturan asosiatif. Metodologi yang mendasari metode ini adalah analisis pola frekuensi tinggi dan pembentukan aturan asosiatif.

Analisis Pola Frekuensi Tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam *database*. Nilai *support* sebuah *item* diperoleh dengan persamaan(1). Sedangkan nilai *support* dari 2 *item* diperoleh dari persamaan(2).

$$Support(A) = \frac{\sum Transaksi\ Mengandung\ A}{\sum Transaksi} x100\% \quad (1)$$

$$Support(A,B) = P(A \cap B) = \frac{\sum Transaksi\ Mengandung\ A\ dan\ B}{\sum Transaksi} x100\% \quad (2)$$



Gambar 4. *Use case diagram* untuk pengguna "Admin"

Pembentukan Aturan Asosiatif

Setelah semua pola frekuensi tertinggi di*item*ukan, baru dicari aturan *asosiatif* yang memenuhi sarat minimum untuk *confidence* dengan menghitung *confidence* aturan *asosiatif* A > B nilai *confidence* dari aturan A > B diperoleh dari persamaan(3).

$$Confidence = P(\frac{B}{A}) = \frac{\sum Transaksi\ Mengandung\ A\ dan\ B}{\sum Transaksi\ Mengandung\ A} \quad (3)$$

Algoritma *Fp-growth*

Algoritma *Fp-growth* merupakan salah satu algoritma dari teknik *association rule* yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul *(frequent itemset)* dalam sebuah kumpulan data [7].

Pengembangan dari algoritma apriori ini terletak dalam *scanning database* dan akurasi *rules*-nya. *Fp-growth* lebih memberikan keuntungan karena hanya dilakukan satu atau dua kali saja *scanning database* sedangkan apriori perlu melakukan *scanning database* berulang ulang. Pada apriori akurasi *rules*-nya lebih tinggi daripada *Fp-growth.* Namun, karena *scanning* dilakukan secara berulang, kecepatannya jadi lebih lambat. Walaupun begitu, kedua algoritma ini memiliki tujuan yang sama yaitu menentukan *frequent itemset* [7].

Algoritma *Fp-growth* memiliki tiga tahapan utama [8], yaitu: 1) Tahap pembangkitan *Conditional Pattern Base*, merupakan sub-*database* yang berisi *prefix path* dan *suffix pattern* (pola akhiran); 2) Tahap pembangkitan *Conditional fp-tree*, dimana pada tahap ini *support count* dari setiap *item* untuk *conditional pattern base* dijumlahkan; 3) Tahap pencarian *Frequent Itemset*, merupakan lintasan tunggal (*single path*), kemudian didapatkan *frequent item*set dengan melakukan kombinasi *item* untuk *conditional fp-tree.*

TABEL 1.
HASIL UJI COBA NILAI SUPPORT

No.	Tahun	
	2013-2013	2013-2015
1	0.18	0.01
2	0.32	0.1
3	0.38	0.01
4	0.1	0.02
5	0.06	0.02
6	0.12	0.07
7	0.01	
8	0.06	
9	0.04	
10	0.01	
11	0.02	
12	0.07	
13	0.02	
14	0.16	
15	0.06	
16	0.15	
17	0.35	
18	0.54	
Rata-rata	0.15	0.04

TABEL 2.
HASIL UJI COBA NILAI CONFIDENCE

No.	Tahun	
	2013-2013	2013-2015
1	0.82	0.75
2	0.78	0.31
3	0.63	0.70
4	0.28	0.78
5	0.33	0.61
6	0.36	1
7	0.23	
8	0.32	
9	0.26	
10	0.80	
11	0.33	
12	0.55	
13	1	
14	0.88	
15	0.72	
16	0.55	
17	0.85	
18	0.83	
Rata-rata	0.58	0.69

Lift ratio

Lift ratio digunakan untuk mengukur seberapa penting *rule* yang telah terbentuk berdasarkan nilai *support* dan *confidence*. *Lift ratio* adalah perbandingan antara *confidence* dengan nilai *benchmark confidence*. *Benchmark confidence* adalah perbandingan antara jumlah semua *item consequent* terhadap total jumlah transaksi [3]. Rumus *benchmark confidence* dan *lift ratio* dapat dilihat pada persamaan(4) dan persamaan(5).

$$Benchmark\ Confidence = \frac{Nc}{N} \quad (4)$$

$$Lift\ ratio = \frac{Confidence(A,C)}{benchmark\ confidence(A,C)} \quad (5)$$

dimana Nc adalah jumlah transaksi dengan *item* dalam *consequent,* dan N adalah jumlah transaksi *database.* Apabila nilai *lift ratio* lebih besar dari 1, maka menunjukkan adanya manfaat dari aturan tersebut. Lebih tinggi nilai *lift ratio* maka lebih besar kekuatan asosiasinya.

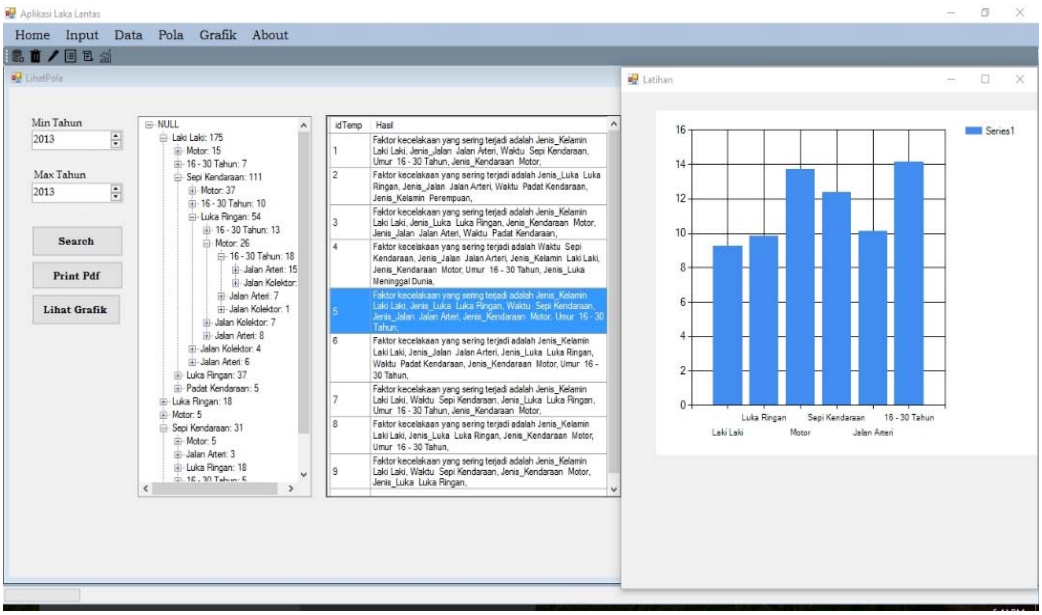
Blok Diagram Sistem

Gambar 1 merupakan diagram blok sistem secara garis besar. Data kecelakaan lalu lintas yang diperoleh dilakukan proses *pre-processing* terlebih dahulu sehingga menghasilkan data yang siap digunakan untuk proses *mining*. Data-data yang telah melewati proses tersebut akan di-*mining*-kan dengan algoritma *Fp-growth* untuk menghasilkan sebuah pola hubungan kecelakaan lalu lintas.

Flowchart

Pada Gambar 2 menjelaskan alur kerja sistem secara umum. Untuk dapat mengakses sistem, admin harus *login* terlebih dahulu kedalam sistem untuk dapat dicek kevalidannya. Setelah berhasil *login*, admin dapat meng-*import* data kecelakaan ke sistem. Sebelum proses pengolahan *Fp-growth*, admin harus memilih rentang waktu kejadian kecelakaan terlebih dahulu. Setelah itu, dilakukan proses *Fp-growth* pada data yang telah dipilih lalu sistem pun akan menampilkan pola yang dihasilkan.

Pada Gambar 3 menjelaskan cara kerja dari proses algoritma *Fp-growth.* Sebelumnya tentukan nilai minimum *support* dan nilai minimum *confidence*. Selanjutnya bangun *tree* dengan cara menentukan frekuensi kemunculan setiap *item* beserta prioritasnya untuk kemudian dipetakan kedalam *tree.* Setelah pembentukan *fp-tree* bangkitkan *conditional pattern base* dengan cara melakukan pembacaan pada *tree* dengan awalan *path* dari bawah ke atas. Kemudian dilakukan pembangkitan *conditional fp-tree* dimana pada tahap ini *support count* untuk tiap *item* pada setiap *conditional pattern base* akan dijumlahkan. *Item* yang memiliki jumlah *support count* lebih besar dengan *minimum support* akan dibangkitkan menjadi sebuah *tree* yang disebut *conditional fp-tree.* Tahapan selanjutnya yaitu pencarian *frequent itemset* dengan cara melakukan subsets dari *conditional fp-tree* terhadap *item* sehinga menghasilkan *frequent itemset.* Apabila nilai *confidence* yang dihasilkan pada subset yang terbentuk lebih besar dari nilai minimum *confidence* yang ditentukan maka *rule* tersebut dapat digunakan.



Gambar 5. Halaman untuk melihat pola

Use Case Diagram

Gambar 4 menunjukkan bahwa admin dapat melakukan semua fungsi yang ada pada sistem. Sebelum menjalankan semua fungsinya, admin harus *login* terlebih dahulu ke sistem.

3. Hasil dan Analisis

Tampilan Antarmuka

Gambar 5 merupakan tampilan dari sistem yang dibangun. Halaman untuk melihat pola tersebut akan menampilkan *tree* dan pola sesuai dengan tahun yang dipilih oleh pengguna berdasarkan algoritma *Fp-growth*.

Pengujian dan Analisis

Pengujian dilakukan untuk mengukur tingkat keakurasian dari metode *Fp-growth*. Pengujian ini menggunakan *lift ratio*, akan tetapi sebelumnya dilakukan pengujian dalam menentukan nilai minimum *support* dan *confidence* terbaik.

Pengujian Minimum Support dan Minimum Confidence

Pengujian nilai *minimum support* ini digunakan untuk mengetahui nilai *minimum support* yang tepat yang akan digunakan dalam sistem. Nilai *minimum support* itu sendiri merupakan syarat minimum jumlah kemunculan *item* secara bersamaan dalam keseluruhan transaksi. Pengujian ini dilakukan dengan menguji coba data dengan nilai *minimum support* terendah yaitu 0.01. Berikut hasil pengujian nilai *minimum support*.

Berdasarkan Tabel 1 hasil yang didapat melalui pengujian *minimum support* pada dataset 2013-2013 didapatkan 18 pola dengan nilai rata rata *support* yaitu 0.15 sedangkan pada dataset 2013-2015 yaitu 0.04. Sehingga peneliti mengambil nilai *support* terkecil yaitu 0.04 sebagai nilai *support* yang akan digunakan pada sistem penetuan pola kecelakaan ini. Pemilihan nilai *minimum support* yang paling kecil dikarenakan nilai *minimum support* adalah nilai acuan dalam perhitungan sebuah *itemset*. Hanya *itemset* yang nilai *support* nya sama atau lebih besar dari nilai *minimum support* saja yang akan dijadikan sebuah pola. Hal ini juga dikarenakan jika menggunakan nilai *minimum support* 0.15 maka pada *dataset* 2013-2015 tidak akan ada pola yang berhasil terseleksi dikarenakan tidak adanya pola yang memiliki nilai *support* 0.15 atau lebih, sehingga peneliti menggunakan nilai 0.4 sebagai nilai *minimum support* yang tepat.

Berdasarkan Tabel 2 hasil yang didapat melalui pengujian *minimum confidence* pada dataset 2013-2013 dihasilkan 18 pola dengan nilai rata rata *confidence* sebesar 0.58. Sedangkan pada dataset 2013-2015, dihasilkan 6 pola dengan nilai rata-rata yaitu 0.69. Berdasarkan hasil tersebut, peneliti mengambil nilai *confidence* terkecil yaitu 0.58 yang kemudian dibulatkan menjadi 0.6 sebagai nilai *confidence* yang akan digunakan pada sistem penentuan pola kecelakaan ini.

TABEL 3.
HASIL PENGUJIAN *LIFT RATIO*

No	Pola	Lift
1	Faktor kecelakaan dengan Jenis_Kelamin Laki Laki, Jenis_Jalan Jalan Arteri, Waktu Sepi Kendaraan, Umur 16 - 30 Tahun	1.05
2	Faktor kecelakaan dengan Jenis_Kelamin Laki Laki, Jenis_Jalan Jalan Arteri, Umur 16 - 30 Tahun, Jenis_Luka Luka Ringan, Tempat_Kejadian Jl Yos Sudarso	1.02
3	Faktor kecelakaan dengan Jenis_Kelamin Laki Laki, Jenis_Jalan Jalan Arteri, Umur 16 - 30 Tahun, Tempat_Kejadian Jl Yos Sudarso	1.06

Pengujian Lift ratio

Berdasarkan uji coba dataset yang telah dilakukan didapatkan nilai minimum *support* yaitu 0.04 dan minimum *confidence* yaitu 0.6. Pengujian selanjutnya yaitu *lift ratio* yang bertujuan untuk mengetahui kuat tidaknya aturan asosiasi yang telah terbentuk dengan persamaan rumus *lift ratio* pada persamaan(5). Hasilnya adalah analisis *lift ratio* untuk data 2013-2013 (1 tahun), dan 2013-2015 (3 tahun) yang direpresentasikan sebagai persamaan(6).

$$Lift\ (x \rightarrow Y) = \frac{Confidence\ (X \rightarrow Y)}{Supp\ (Y)} \qquad (6)$$

dimana, nilai Supp (Y) didapat dari persamaan(7).

$$Supp\ (Y) = \frac{jumlah\ Y}{jumlah\ data\ keseluruhan} \qquad (7)$$

Adapun nilai dari pengujian *lift ratio* dapat dilihat pada Tabel 3. Berdasarkan pengujian *lift ratio* dapat diambil analisis bahwa nilai *support* dan nilai *confidence* akan mempengaruhi besarnya nilai *lift ratio*. Semakin rendah nilai *support* dan semakin tinggi nilai *confidence* maka nilai *lift ratio* akan semakin besar. Begitu pula dengan semakin tinggi nilai *support* dan semakin rendah nilai *confidence* maka nilai *lift ratio* akan semakin kecil. Besar kecilnya nilai *lift ratio* menunjukkan kuatnya hubungan dalam aturan asosiasi tersebut. Hal ini ditunjukkan dengan jika nilai *lift ratio* lebih besar dari 1 berarti hubungan antar *item* pada pola yang dihasilkan semakin kuat.

Pola yang memiliki nilai *lift ratio* tertinggi pada dataset 2013-2013 adalah kecelakaan dengan jenis luka yaitu luka ringan pada jenis jalan arteri di waktu sepi kendaraan dengan korban berjenis kelamin perempuan yang memiliki nilai *lift ratio* 1.19. Kemudian pada dataset 2013-2015 nilai *lift ratio* tertinggi yaitu pada kecelakaan dengan jenis kelamin korban laki-laki pada jalan Yos Sudarso yang merupakan kategori jalan arteri dengan rentang umur korban 16-30 tahun dengan nilai *lift ratio* 1.06.

4. Kesimpulan

Berdasarkan uraian yang telah dijelaskan, maka dapat diambil kesimpulan bahwa aplikasi penentuan pola kecelakaan lalu lintas dengan algoritma *Fp-growth* yang dibangun dapat berjalan dengan baik dan sesuai dengan tujuan yang diharapkan. Meskipun demikian, aplikasi ini hanya membantu pihak terkait dalam membuat keputusan untuk meminimalisir terjadinya kecelakaan, bukan untuk membuat keputusan dalam membuat kebijakan.

Berdasarkan pengujian nilai *minimum support* dan pengujian nilai *minimum confidence* didapatkan nilai *minimum support* yang tepat yaitu 0.04 dan *nilai confidence* yang tepat adalah 0.6. Besar kecilnya nilai *minimum support* dan *confidence* ini nantinya akan mempengaruhi jumlah pola yang dihasilkan.

Berdasarkan pengujian *lift ratio* didapatkan kesimpulan bahwa semakin tinggi nilai *support* dan semakin tinggi nilai *confidence* maka nilai *lift ratio* yang dihasilkan akan semakin kecil. Besar kecilnya nilai *lift ratio* menunjukkan kuatnya hubungan dalam aturan asosiasi tersebut. Jika nilai *lift ratio* lebih besar dari 1 berarti pola yang dihasilkan semakin kuat hubungan antar *item*-nya.

Adapun saran yang dapat diberikan untuk penelitian selanjutnya yaitu dapat menambahkan atribut yang lain yang memiliki kategorinya jangan terlalu banyak seperti tempat kejadian kecelakaan serta menggunakan dataset yang jauh lebih besar dari sekarang, misalnya data yang diolah dari tahun 2011 sampai saat ini 2017. Dan juga penelitian selanjutnya dapat menerapkan algoritma asosiasi yang berbeda seperti *eclat*.

Referensi

[1] Ririanti. "Implementasi Algoritma FP-Growth pada Aplikasi Prediksi Persediaan Sepeda Motor (Studi Kasus PT. Pilar Deli Labumas)," *Pelita Informatika Budi Darma*, vol. 6, pp. 139-144, 2014.

[2] World Health Organization. (2017). Road traffic injuries. [online] Available at: http://www.who.int/violence_injury_prevention/road_traffic/en/ [Diakses pada 25 Okt. 2017].

[3] Bin.go.id. (2017). Kecelakaan Lalu Lintas Menjadi Pembunuh Terbesar Ketiga | Badan Intelijen Negara Republik Indonesia. [online] Available at: http://www.bin.go.id/awas/detil/197/4/21/03/2013/kecelakaan-lalu-lintas-pembunuh-terbesar-ketiga [Diakses pada 25 Okt. 2017].

[4] Fitriany Sitanggang, L., "Aplikasi Data mining Association Rule dengan Algoritma FP-Growth Untuk Mengenali Pola Pembelian

Pelanggan pada Toko Kue (Studi Kasus: L'Cheese Factory)," Politeknik Caltex Riau, 2014.

[5] Hakim, L., "Penentuan Pola Kecelakaan Lalu Lintas Menggunakan Metode *Association Rule* dengan Algoritma Apriori," *University Research Colloquium,* pp. 73-81, 2015.

[6] Todingrara, M., "Tinjauan Kriminologis terhadap Pelanggaran Lalu Lintas yang Menimbulkan Kecelakaan Berakibat Kematian (Studi Kasus Polres Tana Toraja Tahun 2009-2012)," Universitas Hasanuddin, 2013.

[7] Arifin, R. N., "Implementasi Agoritma *Frequent Pattern Growth* (FP-Growth) Menentukan Asosiasi antar Produk (Studi Kasus: Nadiamart)," *Jurnal Teknik ITS*, pp. 68-76, 2015.

[8] Fitriyani, "Implementasi Algoritma FP-Growth Menggunakan *Association Rule* pada *Market Base Analysis*," *Jurnal Informatika*, vol. 2, pp. 296-305, 2015.

Jurnal Sistem Informasi (Journal of Information Systems). 2/13 (2017), 118-124
DOI: http://dx.doi.org/10.21609/jsi.v13i2.551

IMPLEMENTASI ALGORITMA *FP-GROWTH* DALAM PENENTUAN POLA HUBUNGAN KECELAKAAN LALU LINTAS

Rizky Fitria[1], Warnia Nengsih[2], dan Dini Hidayatul Qudsi[3]

[1,2,3] Program Studi Sistem Informasi, Politeknik Caltex Riau, Jl. Umban Sari (Patin) No. 1 Rumbai, Pekanbaru, Riau, 28265, Indonesia

Email: [1]rizky13si@mahasiswa.pcr.ac.id, [2]warnia@pcr.ac.id, [3]dinihq@pcr.ac.id

Abstrak

Tingginya jumlah kecelakaan yang terjadi menjadi alasan penting bagi pihak Satuan Lalu Lintas, salah satunya Polresta Pekanbaru untuk mengetahui hubungan antara factor-faktor penyebab kecelakaan. Hubungan tersebut dipelajari dengan menggunakan teknik *data mining*. Teknik ini bertujuan untuk menemukan informasi berupa pola yang dapat menjadi acuan dalam pengambilan kebijakan di Satuan Lalu Lintas Polresta Pekanbaru. Teknik *data mining* yang digunakan yaitu teknik *association rule* dengan Algoritma *Fp-growth*. Algoritma ini menerapkan struktur *data tree* untuk mengetahui pola kecelakaan lalu lintas. Pola tersebut ditentukan oleh dua parameter, yaitu *support* (nilai penunjang) dan *confidence* (nilai kepastian). Sistem dibuat berbasis *desktop* dengan bahasa pemograman Visual Basic.Net. Sistem ini menghasilkan pola kecelakaan yang sering terjadi. Berdasarkan pengujian *lift ratio,* didapatkan pola kecelakaan yang paling sering terjadi, yaitu faktor kecelakaan dengan jenis_luka adalah luka ringan, jenis_jalan adalah jalan arteri, waktu adalah padat kendaraan, dan jenis_kelamin adalah perempuan dengan nilai *lift ratio* 1.20%. Sedangkan hasil pengujian skala Likert didapatkan bahwa 88.09% pengguna merasa dimudahkan dengan sistem ini dan dapat membantu mereka dalam mengetahui pola kecelakaan yang sering terjadi.

Kata kunci: *kecelakaan lalu lintas, associattion rule, Fp-growth, visual basic.net*

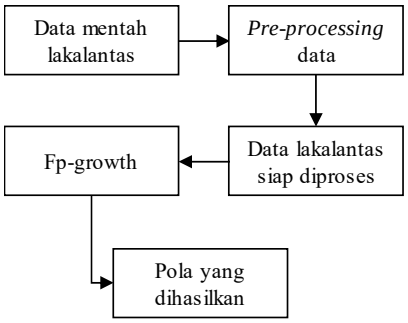
Abstract

The high number of accidents that happened is an important reason for the Traffic Unit, such as for Polresta Pekanbaru to determine the relationship between the factors causing the accident. The relationship is studied using data *mining* technique. This technique aims to find information in the form of patterns that can be a reference for Pekanbaru Police Traffic Unit in making policy. Data *mining* technique used in this research is association *rule* technique with *Fp-growth* Algorithm. This algorithm applies data *tree* structure to learn the traffic accident pattern. The pattern is determined by two parameters, namely support (the value of support) and confidence (the value of certainty). The system is developed as a desktop-based system using Visual Basic.Net programming language. This system generates *frequent* crash patterns. Based on the lift ratio test, the most *frequent* accident pattern, ie accident factor with *jenis_luka* is minor injury, *jenis_jalan* is arterial road, waktu is high density of vehicles, and *jenis_kelamin* is women with lift ratio value 1.20%. While Likert scale test results obtained that 88.09% of users feel facilitated with this system that it can help them in knowing the pattern of accidents that often occur.

Keywords: *traffic accident, association rule, Fp-growth, visual basic.net*

1. Pendahuluan

Kecelakaan lalu lintas (lakalantas) adalah suatu peristiwa yang tidak diduga dan tidak disengaja yang melibatkan kendaraan atau pengguna jalan lain yang mengakibatkan korban manusia dan kerugian harta benda [1]. Hal ini dapat kita lihat dalam kehidupan sehari-hari, terjadinya lakalantas mengakibatkan kerugian yang sangat besar dan bahkan dapat menghilangkan nyawa seseorang. Kejadian ini menjadi hal yang mengerikan yang terjadi di beberapa negara yang umumnya adalah negara-negara berkembang, dimana urusan transportasi masih dalam tahap pembangunan dan perbaikan. Pada saat ini, India merupakan negara dengan jumlah kematian terbanyak akibat kecelakaan lalu lintas, sedangkan Indonesia menempati urutan kelima (berdasarkan data WHO tahun 2011 [2]). Sementara, menurut Badan Intelijen Negara, kecelakaan lalu lintas di Indonesia menjadi pembunuh terbesar ketiga, dibawah penyakit jantung koroner dan TBC [3]. Bahkan, kecelakaan lalu lintas menjadi penyebab



Gambar 1. Diagram blok sistem secara garis besar

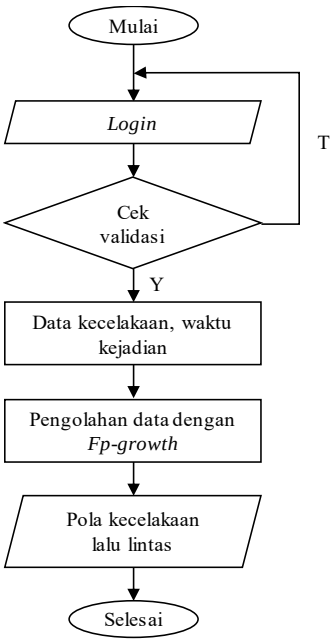
utama kematian anak-anak di dunia, dengan rentang usia 10-24 tahun. Terdapat sekitar 400.000 korban dengan usia dibawah 25 tahun yang meninggal di jalan raya, dengan rata rata angka kematian 1000 orang setiap harinya.

Untuk mendapatkan pola kecelakaan lalu lintas, diperlukan suatu teknik *data mining* yaitu teknik *association rule* untuk menemukan pola hubungan ketekaitan suatu *item*. Algoritma yang digunakan dari teknik *association rule* ini adalah adalah algoritma *Fp-growth. Fp-growth* adalah sebuah metode dalam *data mining* untuk mencari *frequent itemset* tanpa menggunakan *candidate generation. Fp-growth* ini merupakan pengembangan dari algoritma apriori. Kekurangan dari algoritma apriori diperbaiki dengan menghilangkan *candidate generation,* karena *Fp-growth* menggunakan konsep pembangunan *tree* dalam pencarian *frequent itemset.* Hal tersebut membuat algoritma *Fp-growth* lebih cepat dibandingkan algoritma apriori.

Pada penelitian ini, peneliti telah menganalisis hubungan kecelakaan lalu lintas di Kota Pekanbaru dengan variabel-variabel tertentu. Variabel yang akan diteliti antara lain yaitu waktu kejadian, jenis pekerjaan, usia, jenis kelamin, jenis kecelakaan dan jenis luka, jenis kendaraan dan jenis jalan. Penelitian ini menggunakan teknik *Association Rule* yaitu *Fp-growth* untuk mengetahui pola yang terbentuk dari faktor-faktor penyebab terjadinya kecelakaan. Penelitian ini diharapkan dapat membantu sistem yang ada untuk menghasilkan pengetahuan baru yang lebih bermanfaat bagi pihak Satuan Lalu Lintas Polresta Pekanbaru.

Penelitian Terdahulu

Penelitian yang dilakukan oleh [4] membahas tentang implementasi algoritma *Fp-growth* untuk menganalisis data transaksi yang terjadi disebuah toko kue yaitu L'Cheese pekanbaru. Pada pengimplementasiannya, peneliti menerapkan algoritma *Fp-growth* untuk menganalisis pola pembelian pelanggan dengan menganalisis penjualannya. Hasilnya



Gambar 2. Alur kerja sistem secara umum

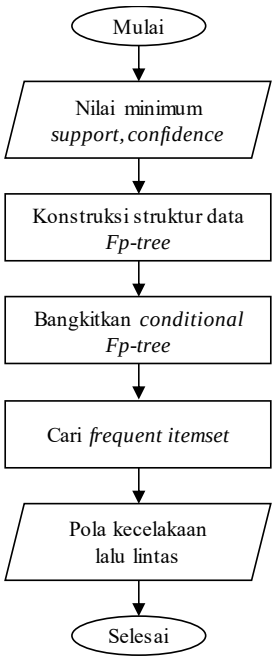
berupa *item* yang paling banyak terjual dan keterkaitan antara dua *item* atau lebih yang nantinya akan berdampak pada penjualan kue tersebut.

Penelitian yang dilakukan oleh [5] menganalisis terhadap penjualan sepeda motor untuk mengetahui sepeda motor mana yang lebih banyak dijual, dan berapa banyak persediaan yang diperlukan perusahaan untuk menyediakan sepeda motor. Untuk mengimplementasikan algoritma *Fp-growth* pada penelitian ini, dilihat dari banyaknya *item* yang terjual, lalu membuat rangkaian *tree* dengan *fp-tree* untuk mengetahui banyaknya *frequent itemset* yang terjadi.

Peneliti [6] melakukan penelitian untuk mengetahui pola hubungan yang sering terbentuk dari faktor-faktor penyebab terjadinya kecelakaan di jalan raya khususnya pada Kabupaten Sleman. Penelitian ini menggunakan metode *association rule* algoritma *apriori.* Faktor yang diambil dalam penelitian ini adalah waktu, profesi, usia, jenis kelamin, dan Surat Izin Mengemudi (SIM).

2. Method

Penelitian yang dibahas pada artikel ini merupakan penelitian untuk membangun sebuah aplikasi *desktop* yang akan menganalisis pola hubungan kecelakaan lalu lintas menggunakan metode *association rule* yaitu algoritma *Fp-growth.* Faktor yang diambil dalam penelitian ini adalah waktu kejadian, jenis pekerjaan, usia, jenis kelamin, jenis kecelakaan, jenis luka, jenis kendaraan dan jenis jalan.



Gambar 3. Cara kerja dari proses algoritma *fp-growth*

Teknik *Association Rule*

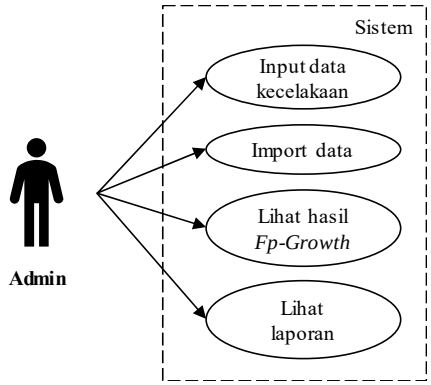
Analisis asosiasi adalah teknik *data mining* untuk menemukan aturan asosiatif antara kombinasi suatu *item*. Contoh aturan asosiatif dari analisis pembelian di suatu pasar swalayan adalah seperti berapa besar kemungkinan seorang pelanggan membeli roti bersamaan dengan susu [4]. *Association Rule* ditentukan oleh dua parameter, yaitu: 1) *Support* adalah suatu ukuran yang menunjukkan seberapa besar tingkat dominasi/persentase suatu kombinasi *item* dari keseluruhan transaksi, 2) *Confidence* adalah suatu ukuran yang menunjukkan kuatnya hubungan antar *item* dalam aturan asosiatif. Metodologi yang mendasari metode ini adalah analisis pola frekuensi tinggi dan pembentukan aturan asosiatif.

Analisis Pola Frekuensi Tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam *database*. Nilai *support* sebuah *item* diperoleh dengan persamaan(1). Sedangkan nilai *support* dari 2 *item* diperoleh dari persamaan(2).

$$Support(A) = \frac{\sum Transaksi\ Mengandung\ A}{\sum Transaksi} x100\% \quad (1)$$

$$Support(A,B) = P(A \cap B) = \frac{\sum Transaksi\ Mengandung\ A\ dan\ B}{\sum Transaksi} x100\% \quad (2)$$



Gambar 4. *Use case diagram* untuk pengguna "Admin"

Pembentukan Aturan Asosiatif

Setelah semua pola frekuensi tertinggi di*item*ukan, baru dicari aturan *asosiatif* yang memenuhi sarat minimum untuk *confidence* dengan menghitung *confidence* aturan *asosiatif* A > B nilai *confidence* dari aturan A > B diperoleh dari persamaan(3).

$$Confidence = P(\frac{B}{A}) = \frac{\sum Transaksi\ Mengandung\ A\ dan\ B}{\sum Transaksi\ Mengandung\ A} \quad (3)$$

Algoritma *Fp-growth*

Algoritma *Fp-growth* merupakan salah satu algoritma dari teknik *association rule* yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul *(frequent itemset)* dalam sebuah kumpulan data [7].

Pengembangan dari algoritma apriori ini terletak dalam *scanning database* dan akurasi *rules*-nya. *Fp-growth* lebih memberikan keuntungan karena hanya dilakukan satu atau dua kali saja *scanning database* sedangkan apriori perlu melakukan *scanning database* berulang ulang. Pada apriori akurasi *rules*-nya lebih tinggi daripada *Fp-growth.* Namun, karena *scanning* dilakukan secara berulang, kecepatannya jadi lebih lambat. Walaupun begitu, kedua algoritma ini memiliki tujuan yang sama yaitu menentukan *frequent itemset* [7].

Algoritma *Fp-growth* memiliki tiga tahapan utama [8], yaitu: 1) Tahap pembangkitan *Conditional Pattern Base*, merupakan sub-*database* yang berisi *prefix path* dan *suffix pattern* (pola akhiran); 2) Tahap pembangkitan *Conditional fp-tree*, dimana pada tahap ini *support count* dari setiap *item* untuk *conditional pattern base* dijumlahkan; 3) Tahap pencarian *Frequent Itemset*, merupakan lintasan tunggal (*single path*), kemudian didapatkan *frequent item*set dengan melakukan kombinasi *item* untuk *conditional fp-tree.*

TABEL 1.
HASIL UJI COBA NILAI SUPPORT

No.	Tahun	
	2013-2013	2013-2015
1	0.18	0.01
2	0.32	0.1
3	0.38	0.01
4	0.1	0.02
5	0.06	0.02
6	0.12	0.07
7	0.01	
8	0.06	
9	0.04	
10	0.01	
11	0.02	
12	0.07	
13	0.02	
14	0.16	
15	0.06	
16	0.15	
17	0.35	
18	0.54	
Rata-rata	0.15	0.04

TABEL 2.
HASIL UJI COBA NILAI CONFIDENCE

No.	Tahun	
	2013-2013	2013-2015
1	0.82	0.75
2	0.78	0.31
3	0.63	0.70
4	0.28	0.78
5	0.33	0.61
6	0.36	1
7	0.23	
8	0.32	
9	0.26	
10	0.80	
11	0.33	
12	0.55	
13	1	
14	0.88	
15	0.72	
16	0.55	
17	0.85	
18	0.83	
Rata-rata	0.58	0.69

Lift ratio

Lift ratio digunakan untuk mengukur seberapa penting *rule* yang telah terbentuk berdasarkan nilai *support* dan *confidence*. *Lift ratio* adalah perbandingan antara *confidence* dengan nilai *benchmark confidence*. *Benchmark confidence* adalah perbandingan antara jumlah semua *item consequent* terhadap total jumlah transaksi [3]. Rumus *benchmark confidence* dan *lift ratio* dapat dilihat pada persamaan(4) dan persamaan(5).

$$Benchmark\ Confidence = \frac{Nc}{N} \quad (4)$$

$$Lift\ ratio = \frac{Confidence(A,C)}{benchmark\ confidence(A,C)} \quad (5)$$

dimana Nc adalah jumlah transaksi dengan *item* dalam *consequent,* dan N adalah jumlah transaksi *database.* Apabila nilai *lift ratio* lebih besar dari 1, maka menunjukkan adanya manfaat dari aturan tersebut. Lebih tinggi nilai *lift ratio* maka lebih besar kekuatan asosiasinya.

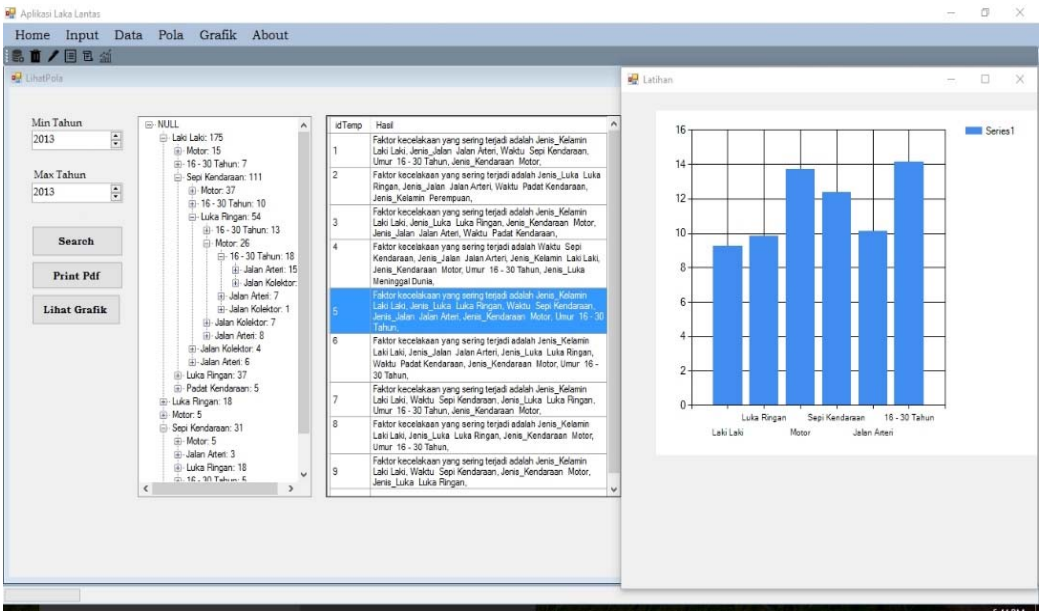
Blok Diagram Sistem

Gambar 1 merupakan diagram blok sistem secara garis besar. Data kecelakaan lalu lintas yang diperoleh dilakukan proses *pre-processing* terlebih dahulu sehingga menghasilkan data yang siap digunakan untuk proses *mining*. Data-data yang telah melewati proses tersebut akan di-*mining*-kan dengan algoritma *Fp-growth* untuk menghasilkan sebuah pola hubungan kecelakaan lalu lintas.

Flowchart

Pada Gambar 2 menjelaskan alur kerja sistem secara umum. Untuk dapat mengakses sistem, admin harus *login* terlebih dahulu kedalam sistem untuk dapat dicek kevalidannya. Setelah berhasil *login*, admin dapat meng-*import* data kecelakaan ke sistem. Sebelum proses pengolahan *Fp-growth*, admin harus memilih rentang waktu kejadian kecelakaan terlebih dahulu. Setelah itu, dilakukan proses *Fp-growth* pada data yang telah dipilih lalu sistem pun akan menampilkan pola yang dihasilkan.

Pada Gambar 3 menjelaskan cara kerja dari proses algoritma *Fp-growth.* Sebelumnya tentukan nilai minimum *support* dan nilai minimum *confidence*. Selanjutnya bangun *tree* dengan cara menentukan frekuensi kemunculan setiap *item* beserta prioritasnya untuk kemudian dipetakan kedalam *tree.* Setelah pembentukan *fp-tree* bangkitkan *conditional pattern base* dengan cara melakukan pembacaan pada *tree* dengan awalan *path* dari bawah ke atas. Kemudian dilakukan pembangkitan *conditional fp-tree* dimana pada tahap ini *support count* untuk tiap *item* pada setiap *conditional pattern base* akan dijumlahkan. *Item* yang memiliki jumlah *support count* lebih besar dengan *minimum support* akan dibangkitkan menjadi sebuah *tree* yang disebut *conditional fp-tree.* Tahapan selanjutnya yaitu pencarian *frequent itemset* dengan cara melakukan subsets dari *conditional fp-tree* terhadap *item* sehinga menghasilkan *frequent itemset.* Apabila nilai *confidence* yang dihasilkan pada subset yang terbentuk lebih besar dari nilai minimum *confidence* yang ditentukan maka *rule* tersebut dapat digunakan.



Gambar 5. Halaman untuk melihat pola

Use Case Diagram

Gambar 4 menunjukkan bahwa admin dapat melakukan semua fungsi yang ada pada sistem. Sebelum menjalankan semua fungsinya, admin harus *login* terlebih dahulu ke sistem.

3. Hasil dan Analisis

Tampilan Antarmuka

Gambar 5 merupakan tampilan dari sistem yang dibangun. Halaman untuk melihat pola tersebut akan menampilkan *tree* dan pola sesuai dengan tahun yang dipilih oleh pengguna berdasarkan algoritma *Fp-growth*.

Pengujian dan Analisis

Pengujian dilakukan untuk mengukur tingkat keakurasian dari metode *Fp-growth*. Pengujian ini menggunakan *lift ratio*, akan tetapi sebelumnya dilakukan pengujian dalam menentukan nilai minimum *support* dan *confidence* terbaik.

Pengujian Minimum Support dan Minimum Confidence

Pengujian nilai *minimum support* ini digunakan untuk mengetahui nilai *minimum support* yang tepat yang akan digunakan dalam sistem. Nilai *minimum support* itu sendiri merupakan syarat minimum jumlah kemunculan *item* secara bersamaan dalam keseluruhan transaksi. Pengujian ini dilakukan dengan menguji coba data dengan nilai *minimum support* terendah yaitu 0.01. Berikut hasil pengujian nilai *minimum support*.

Berdasarkan Tabel 1 hasil yang didapat melalui pengujian *minimum support* pada dataset 2013-2013 didapatkan 18 pola dengan nilai rata rata *support* yaitu 0.15 sedangkan pada dataset 2013-2015 yaitu 0.04. Sehingga peneliti mengambil nilai *support* terkecil yaitu 0.04 sebagai nilai *support* yang akan digunakan pada sistem penetuan pola kecelakaan ini. Pemilihan nilai *minimum support* yang paling kecil dikarenakan nilai *minimum support* adalah nilai acuan dalam perhitungan sebuah *itemset*. Hanya *itemset* yang nilai *support* nya sama atau lebih besar dari nilai *minimum support* saja yang akan dijadikan sebuah pola. Hal ini juga dikarenakan jika menggunakan nilai *minimum support* 0.15 maka pada *dataset* 2013-2015 tidak akan ada pola yang berhasil terseleksi dikarenakan tidak adanya pola yang memiliki nilai *support* 0.15 atau lebih, sehingga peneliti menggunakan nilai 0.4 sebagai nilai *minimum support* yang tepat.

Berdasarkan Tabel 2 hasil yang didapat melalui pengujian *minimum confidence* pada dataset 2013-2013 dihasilkan 18 pola dengan nilai rata rata *confidence* sebesar 0.58. Sedangkan pada dataset 2013-2015, dihasilkan 6 pola dengan nilai rata-rata yaitu 0.69. Berdasarkan hasil tersebut, peneliti mengambil nilai *confidence* terkecil yaitu 0.58 yang kemudian dibulatkan menjadi 0.6 sebagai nilai *confidence* yang akan digunakan pada sistem penentuan pola kecelakaan ini.

TABEL 3.
HASIL PENGUJIAN *LIFT RATIO*

No	Pola	Lift
1	Faktor kecelakaan dengan Jenis_Kelamin Laki Laki, Jenis_Jalan Jalan Arteri, Waktu Sepi Kendaraan, Umur 16 - 30 Tahun	1.05
2	Faktor kecelakaan dengan Jenis_Kelamin Laki Laki, Jenis_Jalan Jalan Arteri, Umur 16 - 30 Tahun, Jenis_Luka Luka Ringan, Tempat_Kejadian Jl Yos Sudarso	1.02
3	Faktor kecelakaan dengan Jenis_Kelamin Laki Laki, Jenis_Jalan Jalan Arteri, Umur 16 - 30 Tahun, Tempat_Kejadian Jl Yos Sudarso	1.06

Pengujian Lift ratio

Berdasarkan uji coba dataset yang telah dilakukan didapatkan nilai minimum *support* yaitu 0.04 dan minimum *confidence* yaitu 0.6. Pengujian selanjutnya yaitu *lift ratio* yang bertujuan untuk mengetahui kuat tidaknya aturan asosiasi yang telah terbentuk dengan persamaan rumus *lift ratio* pada persamaan(5). Hasilnya adalah analisis *lift ratio* untuk data 2013-2013 (1 tahun), dan 2013-2015 (3 tahun) yang direpresentasikan sebagai persamaan(6).

$$Lift\ (x \rightarrow Y) = \frac{Confidence\ (X \rightarrow Y)}{Supp\ (Y)} \qquad (6)$$

dimana, nilai Supp (Y) didapat dari persamaan(7).

$$Supp\ (Y) = \frac{jumlah\ Y}{jumlah\ data\ keseluruhan} \qquad (7)$$

Adapun nilai dari pengujian *lift ratio* dapat dilihat pada Tabel 3. Berdasarkan pengujian *lift ratio* dapat diambil analisis bahwa nilai *support* dan nilai *confidence* akan mempengaruhi besarnya nilai *lift ratio*. Semakin rendah nilai *support* dan semakin tinggi nilai *confidence* maka nilai *lift ratio* akan semakin besar. Begitu pula dengan semakin tinggi nilai *support* dan semakin rendah nilai *confidence* maka nilai *lift ratio* akan semakin kecil. Besar kecilnya nilai *lift ratio* menunjukkan kuatnya hubungan dalam aturan asosiasi tersebut. Hal ini ditunjukkan dengan jika nilai *lift ratio* lebih besar dari 1 berarti hubungan antar *item* pada pola yang dihasilkan semakin kuat.

Pola yang memiliki nilai *lift ratio* tertinggi pada dataset 2013-2013 adalah kecelakaan dengan jenis luka yaitu luka ringan pada jenis jalan arteri di waktu sepi kendaraan dengan korban berjenis kelamin perempuan yang memiliki nilai *lift ratio* 1.19. Kemudian pada dataset 2013-2015 nilai *lift ratio* tertinggi yaitu pada kecelakaan dengan jenis kelamin korban laki-laki pada jalan Yos Sudarso yang merupakan kategori jalan arteri dengan rentang umur korban 16-30 tahun dengan nilai *lift ratio* 1.06.

4. Kesimpulan

Berdasarkan uraian yang telah dijelaskan, maka dapat diambil kesimpulan bahwa aplikasi penentuan pola kecelakaan lalu lintas dengan algoritma *Fp-growth* yang dibangun dapat berjalan dengan baik dan sesuai dengan tujuan yang diharapkan. Meskipun demikian, aplikasi ini hanya membantu pihak terkait dalam membuat keputusan untuk meminimalisir terjadinya kecelakaan, bukan untuk membuat keputusan dalam membuat kebijakan.

Berdasarkan pengujian nilai *minimum support* dan pengujian nilai *minimum confidence* didapatkan nilai *minimum support* yang tepat yaitu 0.04 dan *nilai confidence* yang tepat adalah 0.6. Besar kecilnya nilai *minimum support* dan *confidence* ini nantinya akan mempengaruhi jumlah pola yang dihasilkan.

Berdasarkan pengujian *lift ratio* didapatkan kesimpulan bahwa semakin tinggi nilai *support* dan semakin tinggi nilai *confidence* maka nilai *lift ratio* yang dihasilkan akan semakin kecil. Besar kecilnya nilai *lift ratio* menunjukkan kuatnya hubungan dalam aturan asosiasi tersebut. Jika nilai *lift ratio* lebih besar dari 1 berarti pola yang dihasilkan semakin kuat hubungan antar *item*-nya.

Adapun saran yang dapat diberikan untuk penelitian selanjutnya yaitu dapat menambahkan atribut yang lain yang memiliki kategorinya jangan terlalu banyak seperti tempat kejadian kecelakaan serta menggunakan dataset yang jauh lebih besar dari sekarang, misalnya data yang diolah dari tahun 2011 sampai saat ini 2017. Dan juga penelitian selanjutnya dapat menerapkan algoritma asosiasi yang berbeda seperti *eclat*.

Referensi

[1] Ririanti. "Implementasi Algoritma FP-Growth pada Aplikasi Prediksi Persediaan Sepeda Motor (Studi Kasus PT. Pilar Deli Labumas)," *Pelita Informatika Budi Darma*, vol. 6, pp. 139-144, 2014.

[2] World Health Organization. (2017). Road traffic injuries. [online] Available at: http://www.who.int/violence_injury_prevention/road_traffic/en/ [Diakses pada 25 Okt. 2017].

[3] Bin.go.id. (2017). Kecelakaan Lalu Lintas Menjadi Pembunuh Terbesar Ketiga | Badan Intelijen Negara Republik Indonesia. [online] Available at: http://www.bin.go.id/awas/detil/197/4/21/03/2013/kecelakaan-lalu-lintas-pembunuh-terbesar-ketiga [Diakses pada 25 Okt. 2017].

[4] Fitriany Sitanggang, L., "Aplikasi Data mining Association Rule dengan Algoritma FP-Growth Untuk Mengenali Pola Pembelian

Pelanggan pada Toko Kue (Studi Kasus: L'Cheese Factory)," Politeknik Caltex Riau, 2014.

[5] Hakim, L., "Penentuan Pola Kecelakaan Lalu Lintas Menggunakan Metode *Association Rule* dengan Algoritma Apriori," *University Research Colloquium,* pp. 73-81, 2015.

[6] Todingrara, M., "Tinjauan Kriminologis terhadap Pelanggaran Lalu Lintas yang Menimbulkan Kecelakaan Berakibat Kematian (Studi Kasus Polres Tana Toraja Tahun 2009-2012)," Universitas Hasanuddin, 2013.

[7] Arifin, R. N., "Implementasi Agoritma *Frequent Pattern Growth* (FP-Growth) Menentukan Asosiasi antar Produk (Studi Kasus: Nadiamart)," *Jurnal Teknik ITS*, pp. 68-76, 2015.

[8] Fitriyani, "Implementasi Algoritma FP-Growth Menggunakan *Association Rule* pada *Market Base Analysis*," *Jurnal Informatika*, vol. 2, pp. 296-305, 2015.

A Parallel FP-growth Algorithm Based on GPU

Hao JIANG
School of Computer Science and Engineering
Southeast University
Nanjing, China
E-mail: hjiang@seu.edu.cn

He MENG
School of Computer Science and Engineering
Southeast University
Nanjing, China
E-mail: menghe2012@sina.com

***Abstract*—This paper proposes and implements a parallel scheme of FP-growth algorithm and implements this parallel algorithm (PFP-growth algorithm). Experimental results show that, compared with FP-growth algorithm, PFP-growth algorithm is more efficient, and the larger the data set is, the lower the support threshold is, the more remarkable the speedup is.**

Keywords-frequent itemset mining; FP-growth; GPU; parallel computing

I. INTRODUCTION

Association rule mining is an important approach in data mining. Association rule mining means mining potentially valuable links between data items from a large number of data items. In general, association rule mining can be divided into two sub-problems: frequent item sets generation and rule generation. Frequent itemsets mining is the core of association rules mining algorithm, its performance greatly determines the overall performance of association rules mining. At present, the frequent itemsets mining algorithms are divided into three major categories: Apriori-like algorithm; FP-growth-like algorithm and Eclat-like algorithm. Among them, the FP-growth algorithm is used widely.

FP-growth algorithm is an association rules mining algorithm proposed by Han Jiawei and his colleagues in 2000 [1]. Compared with the Apriori algorithm, FP-growth has an order of magnitude acceleration in the mining speed. However, the FP-growth algorithm is a recursive algorithm, which leads to the performance of the algorithm limited by the development of CPU, moreover, the recursion stack will occupy a large amount of memory space. In this paper, by use of the advantages of GPU in parallel computing, on the basis of FP-growth algorithm, we propose a parallel frequent itemset mining algorithm based on GPU, called PFP-growth algorithm.

The experimental results show the powerful parallel computing ability of GPU. There is no doubt that parallelization using GPU is a trend in the research of data mining algorithms.

II. BACKGROUND AND RELATED WORKS

A. The Introduction of GPU Computing and CUDA

The modern GPUs are designed for highly parallel tasks, so they are very efficient at manipulating computer graphics. The GPU can execute many floating point operations in finite time. Modern GPUs are massively parallel architectures with thousands of computing cores and are easily programmable. The main difference between CPU and GPU is CPU has less but more powerful cores and GPU has more but less powerful cores. CPU consumes more power than GPU does. CPU is good for task parallelism and whereas GPU is good for data parallelism.

GPU computing uses GPU and CPU together to accelerate executions of applications. In GPU computing inherent sequential part of the code run on the CPU while parallel portion of code is executed using thousands of parallel threads that runs on the GPU. We can explore more algorithms on GPU, which has been sidelined because availability of less computational power.

CUDA is a parallel programming platform designed by NVIDIA for NVDIA GPUs. It can be used to tap the power of the GPU. CUDA Provides a small set of extensions to standard programming languages. The CPU and GPU are treated as separate devices that have their own memory. In CUDA serial code executes itself in CPU and is called Host while parallel code executes on GPU and is called device. A CUDA kernel, is a code that is to be executed in parallel by all the GPU threads. Kernel is launched from the main program. When the kernel is launched, system creates many threads on GPU that executes simultaneously. Each thread has unique id and the threads are grouped into a warp, warps are group into a block, blocks are grouped into a grid. The CUDA platform is easy to learn and it generate optimized code on GPU and thus we are using it in this work.

B. The Introduction of FP-growth Algorithm

FP-growth algorithm uses frequent pattern growth to mine frequent patterns without candidate generation [1, 2]. FP-growth algorithm is mainly divided into two steps: FP-tree construction and recursively mining FP-tree. In the process of mining frequent itemset, compared to the Apriori algorithm scanning the database repeatedly, the FP-growth algorithm only needs to scan the database twice. For the first scan, we can get frequent 1-itemsets, for the second scan, sort the items by the support count and delete the non-frequent items, then construct FP-tree by using the ordered frequent 1-itemsets. After FP-tree construction is completed, searching FP-tree by using FP-growth algorithm, find the conditional pattern bases of the corresponding itemsets, and then construct conditional FP-tree. Based on conditional FP-

978-1-5386-1412-9/17 $31.00 © 2017 IEEE
DOI 10.1109/ICEBE.2017.24

IEEE computer society

tree, the FP-growth algorithm is recursively implemented until all frequent itemsets is mined.

For example, about the transaction database shown in Figure 1, for the first scan, we can get the frequent 1-itemsets {f:4, c:4, a:3, b:3, m:3, p:3}.

DB:

TID	Items bought
101	{f,c,a,d,g,i,m,p}
102	{a,b,c,f,l,m,o}
103	{b,f,h,j,o}
104	{b,c,k,s,p}
105	{a,f,c,e,l,m,p,n}

min_support=3

Figure 1. A transaction database as running example.

According to the frequent 1-itemsets, delete the non-frequent items contained in each transaction of the databases, and merge and sort them. The result is shown in figure 2.

TID	Items bought
101	{f,c,a,d,g,i,m,p}
102	{a,b,c,f,l,m,o}
103	{b,f,h,j,o}
104	{b,c,k,s,p}
105	{a,f,c,e,l,m,p,n}

Frequent 1-itemsets: {f,c,a,b,m,p}

min_support=3

TID	order frequent items
101	{f,c,a,m,p:2}
102	{f,c,a,b,m:1}
103	{f,b:1}
104	{c,b,p:1}

Figure 2. The ordered frequent items

After the ordered frequent items is got, the FP-tree starts to be constructed. The process of construction is shown on the left of the figure 3. It is seen that the FP-tree consists of two parts: the header table and the prefix tree, in which the header table is the frequent 1-itemsets.

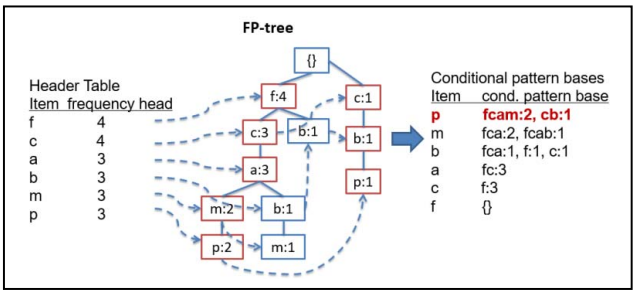


Figure 3. The construction of FP-growth and conditional frequent pattern bases

After FP-tree is constructed, mining frequent itemsets is started by using FP-growth algorithm. First, starting with the bottom item of the FP-tree, construct the conditional pattern base for each item. The construction process of the conditional pattern base is shown on the right side of figure 3. Then, accumulate the support count of each item in each conditional mode base, and filter the items below the threshold to construct the conditional FP-tree. Recursively mine each conditional FP-tree until the constructed new FP-tree is empty, or only one path is included, and then the FP-growth frequent itemset mining ends. When the constructed FP-tree is empty, the prefix is a frequent itemset. When only one path is included, all the itemsets in the path are frequent itemsets.

The performance of the FP-growth algorithm is improved by an order of magnitude over the Apriori algorithm. But the FP-growth algorithm using recursive method to mine frequent itemsets, occupy a large amount of memory space, and when the transaction contains a large number of items, it is easy to cause a stack overflow. At the same time, when FP-growth algorithm is implemented, FP-tree uses a linked storage structure, which makes it difficult to parallelize the algorithm on GPU.

III. THE PFP-GROWTH ALGORITHM

A. The Data Structure of PFP-Growth Algorithm

In the field of frequent itemsets mining, it is still difficult to implement the irregular data structure such as tree on GPU while parallelizing FP-growth algorithm on GPU. To solve the storage problem on GPU, the transaction data set FP-tree is implemented by an array on GPU, which is logically still a tree, denoted as FP-array [3].

FP-array is a dynamic two-dimensional array that records the relevant information about each node on the FP-tree, including the item's ID, the address of the parent node of the item in the FP-array and the support count of the item. The path of the current node to the root node can be checked by using the address of the parent node stored in the FP-array, and the support count of the current node is the support count of the path. The FP-array can replace FP-tree to mine frequent itemsets on GPU.

Before the construction of FP-array, we need to construct the FP-tree of the transaction data set firstly. The FP-tree constructed in the PFP-growth algorithm (named PFP-tree) is a little different from the FP-tree in the traditional FP-growth algorithm. Each node in PFP-tree has a pointer to its parent node, and each leaf node has a pointer to the next leaf node. The PFP-tree constructed from the database in figure 1 is shown in figure 4.

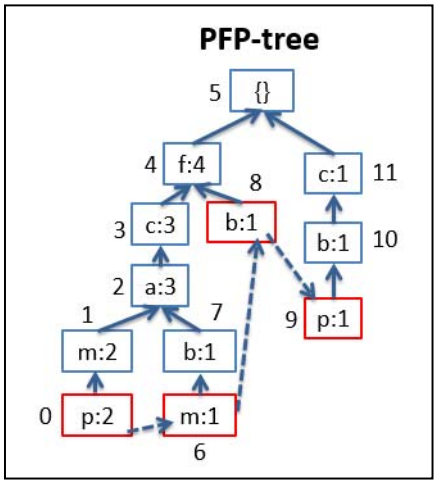


Figure 4. An example of PFP-tree

When the construction of PFP-tree completed, we start to construct the FP-array. Take the leftmost leaf node in FPNR-tree as the starting point, use the pointer pointing to the parent node to find the path of the leaf node to the root node, and put the information of passing nodes in the FP-array. In the searching process, we find that some internal nodes maybe appear on more than one path which leaf nodes to the root node, thus these internal nodes will repeat the record storage in FP-array.

To solve this problem, we introduce a Hash table to indicate the nodes already stored in the FP-array, which includes two properties, the address of the node and the index of the node in FP-array. Before put the information of one node in FP-array, we search the Hash table to determine whether the node has been recorded. If not, store the item's ID, the index in FP-array of its parent node and the support count in the FP-array, then put the node's address and index in the Hash table. If the Hash table has recorded the node, get the index, put it in FP-array with the item's ID and the support count, then stop the search of this path, and jump to the next leaf node to construct the FP-array. The figure 5 illustrates the process to construct FP-array. The algorithm to construct FP-array is shown in Algorithm 1.

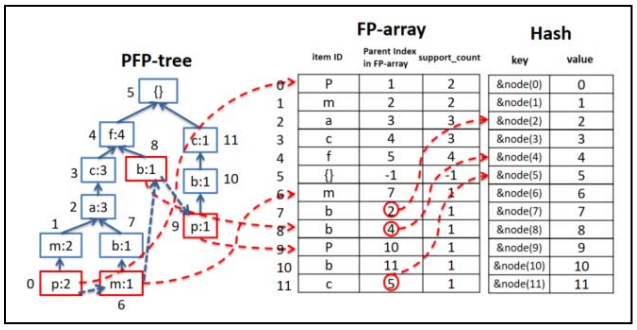


Figure 5. The contruction of FP-array

Algorithm 1 (constructing FP-array)

Input: PFP-tree, Hash table
Output: FP-array
1: for each node p in PFP-tree
2: store the p's info. in FP-array and record it in Hash
3: parentNode = the parent node of p
4: while(parentNode !={})
5: if parentNode of p in Hash then
6: get the index of the node
7: store the index with the item's ID and support count of p in FP-array
8: turn to step 1;
9: else
10: store the item's ID, index of parentNode and support count in FP-array
11: record parentNode in Hash table
12: p = parentNode

In order to mine frequent items in parallel on GPU, it needs to establish a new dynamic array, ElePos (Element Position), to store items which will be mined. ElePos includes the item's ID, the item's address in the FP-array and the support count of the item. The construction process of ElePos consists of two parts: Part one is initialized on CPU. By scanning the FP-array, the information of each of the frequent 1-items is written into ElePos. Part two is to update the ElePos on the GPU. After judging whether the candidate k+1 items are frequent on GPU, we can get the new frequent k+1-items. Then the ElePos is updated by the new frequent itemsets generated by all the threads. Figure 6 is an initialization ElePos obtained according to the FP-array in Figure 5.

ElePos

Item ID	Index in FP-array	support_count
p	0	2
m	1	2
a	2	3
c	3	3
f	4	4
m	6	1
b	7	1
b	8	1
p	9	1
b	10	1
c	11	1

Figure 6. An example of ElePos

B. *The Design of PFP-Growth Algorithm*

Aiming at some problems of FP-growth algorithm, this paper proposes a non-recursive parallel FP-growth algorithm based on GPU, or PFP-growth algorithm for short.

PFP-growth algorithm mainly includes two parts: Part one is to use the GPU to mine candidate k+1 items, such as algorithm 2; Part two is to judge whether the candidate k+1-items are frequent, and then generate the frequent k+1-items, and last, for the next mining, the ElePos is updated by the frequent k+1-items generated from all the threads, such as algorithm 3 [8, 9, 10, 11]. Both of these two algorithms are executed in parallel on GPU.

Algorithm 2 (mining candidate k+1-items)

Input: FP-array, item
Output: candidate k+1-items
1: t = the parent node of item
2: **if** t is not the root
3: generate a new candidate k+1-item, put it into k1_ElePos
4: update t = the parent node of t

5: turn to 2
6: **end if**

Algorithm 2 (Judging candidate k+1-items)

Input: k1_ElePoss, min_sup
Output: frequent k+1-items, ElePos
1: transform k1_ElePoss into Map<key, value>, take ID as key, support count as value, if the key has exited, sum the value
2: delete each element in Map which support count is lower than min_sup
3: wait for other threads to complete
4: update ElePos by frequent k+1-items generated from all the threads

The overall flow of the PFP-growth algorithm is shown in Figure 7. First, the algorithm constructs the FP-array from the data set, and then constructs ElePos according to the FP-array. During the process of mining, a dynamic array (k1_ElePos) like ElePos is needed to store the candidate k+1-items. Second, the FP-array and ElePos is transferred into the GPU memory, and the GPU allocates threads to execute the algorithm according to the maximum number of available threads and the number of elements in ElePos. If the number of elements in ElePos is less than the number of available threads in GPU, GPU directly allocates enough threads to parallel execute. If not, the ElePos is split in some segments for multiple processing. The candidate k+1-items are mined by each thread of GPU of algorithm 1, and they are stored into k1_ElePos. Third, the candidate k+1-items in k1_ElePos are judged to filter out items whose support counts are less than min_sup. Finally, the results of all threads are combined to generate the new ElePos for the next mining. When the mining is finished, the frequent itemsets are returned to CPU and outputted. Figure 8 is an example of the PFP-growth algorithm for mining frequent itemsets.

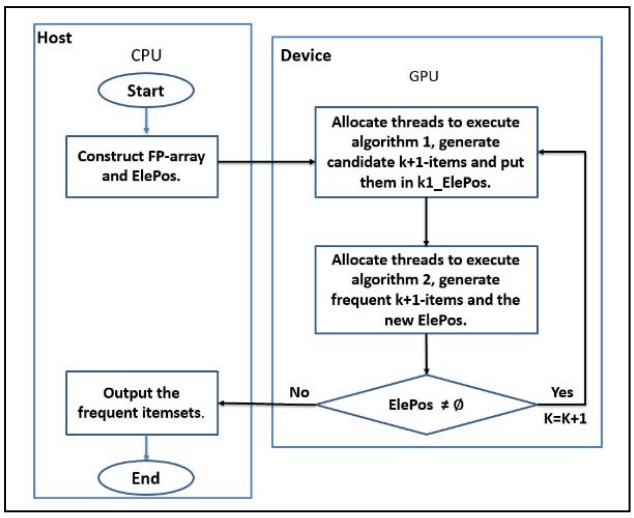


Figure 7. he flow of PFP-growth algorithm

Compared with the traditional FP-growth algorithm, the PFP-growth algorithm stores the transaction data set with FP-array, which is read only, so threads can be efficiently executed in parallel to mine frequent itemsets. As a result, it speeds up remarkably the data processing and reduces dramatically the run time. Although there are still some shortcomings in the PFP-growth algorithm, such as when the transaction data are too large and dense, the candidate itemsets list is too large, resulting in GPU memory is insufficient, but overall, compared with the traditional FP-growth algorithm, PFP-growth algorithm greatly enhances the efficiency of mining frequent itemsets.

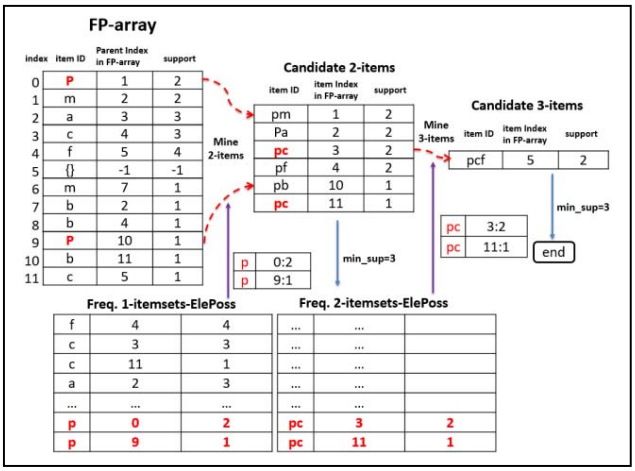


Figure 8. An example of PFP-growth for mining frequent items

IV. Experimental Results and Discussion

In this section, we implemented the PFP-growth algorithm and tested its performance comparison of PFP-growth with the traditional FP-growth algorithm. In the experiment, the CPU and GPU what we use are i7-4790K and GTX 980 respectively. The specific experimental environment is shown in Table Ⅰ. We report experimental results on four data sets, which are Retail, T10I4D100K, Chess and Mushroom. They all come from http://fimi.ua.ac.be/data/. Table Ⅱ presents some main information about them.

TABLE I. THE MAIN INFORMATION OF DATA SETS

Hardware Platform	CPU: Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz GPU: NVIDIA GeForce GTX 980 Ti Memory: 8GB(2*4GB Kingston DDR3L 1600MHZ) Disk: WDC10EZEX-60ZF5A0-1TB
Software Platform	OS: Windows 10 Professional Edition IDE: Visual Studio 2013

TABLE II. THE MAIN INFORMATION OF DATA SETS

Database	***Type***	***Items***	***Trans***
(a)retail	Sparse	16470	88162
(b)T40I10D100K	Sparse	1000	100000
(c)Chess	Dense	75	3196
(d)Mushroom	Dense	119	8124

The experimental results are shown in figure 8-11. They respectively present the comparison of the run times of PFP-

growth algorithm and FP-growth algorithm on each data set with different support threshold.

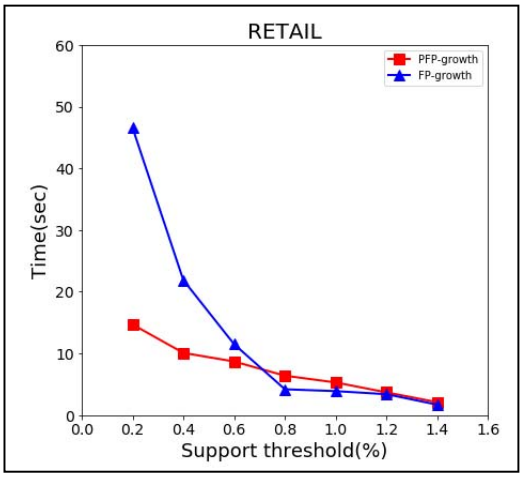


Figure 9. The result of RTAIL

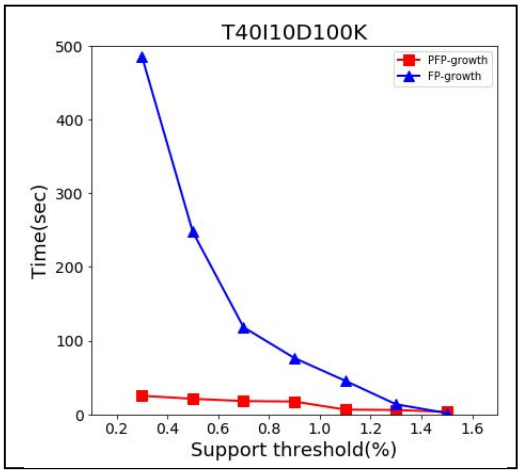


Figure 10. The result of T40I10D100K

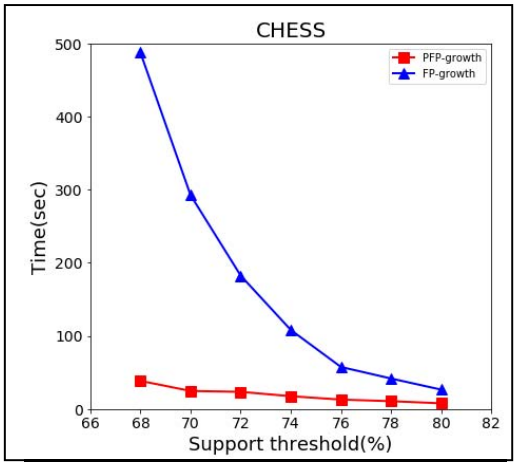


Figure 11. The result of CHESS

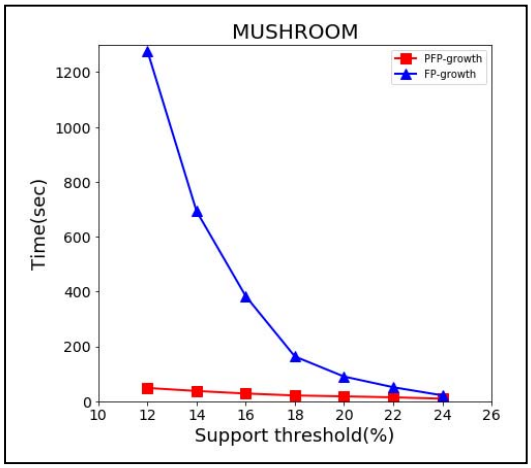


Figure 12. The result of MUSHROOM

From the experimental results, we can see that, no matter what kind of data set, with the support threshold decreasing, the difference in the run time between FP-growth algorithm and PFP-growth algorithm is getting larger and larger. In terms of time efficiency, the PFP-growth algorithm has obvious advantage. Especially in Figure 9, 10 and 11, it can be seen that in larger and denser data sets, PFP-growth algorithm has a much greater speedup. The experimental results show that, compared with FP-growth algorithm, the PFP-growth algorithm has dramatically improved the performance of mining frequent itemsets.

V. CONCLUTION

This paper firstly gives a brief introduction and analysis of the classical FP-growth algorithm. Because FP-growth algorithm uses recursive method to construct conditional pattern base and conditional pattern tree when mining frequent itemsets, it costs a lot of time and memory. When the data set is too large or the support threshold is too small, the efficiency of mining frequent itemsets is often unsatisfactory. To solve this problem, this paper proposes a PFP-growth algorithm based on GPU, which uses GPU's powerful parallel computing ability to mine frequent itemsets, greatly improves the mining efficiency. Experimental results show that, compared with the traditional FP-growth algorithm, PFP-growth algorithm has remarkable acceleration effect. This algorithm also reflects the tremendous advantages of GPU parallel computing, so it would be a trend to apply GPU parallel computing to other data mining algorithms.

ACKNOWLEDGMENT

This work was supported by Collaborative Innovation Center of Wireless Communications Technology and the National Natural Science Foundation of China (Grant No. 61672152).

REFERENCES

[1] J. Han, J. Pei, and Y. Yin, Mining frequent patterns without candidate generation. In Proceedings of ACM SIGMOD'00, pp 1-12, May 2000.

[2] J. Han, J. Pei and Y. Yin, “Mining frequent patterns without candidate generation: A frequent-pattern tree approach,” DATA MINING AND KNOWLEDGE DISCOVERY, vol. 8, pp. 53-87, 2004.

[3] H. JIANG, Y. LIAO and S. NI, “A New Algorithm for Mining Frequent Itemset Using Efficient Data Structure,” International Conference on Computer Science and Software Engineering, 2014.

[4] J. D. Owens, M. Houston and D. Luebke, “GPU computing,” PROCEEDINGS OF THE IEEE, vol. 96, pp. 879-899, 2008.

[5] M. Garland, S. Le Grand and J. Nickolls, “Parallel computing experiences with CUDA,” IEEE MICRO, vol. 28, pp. 13-27,2008.

[6] F. Zhang, Y. Zhang and J. D. Bakos, “Accelerating frequent itemset mining on graphics processing units,” JOURNAL OF SUPERCOMPUTING, vol. 66, pp. 94-117,2013.

[7] ZH. Zhou, W. Wang and R. Kumar, “Parallel Frequent Pattern Mining without Candidate Generation on GPUs,” 14th IEEE International Conference on Data Mining, PP. 1046-1052, 2014.

[8] F. Zhang, Y. Zhang, and J. Bakos, “Gpapriori: Gpu-accelerated Frequent Itemset Mining,” Proc. of the 2011 IEEE International Conference on Cluster Computing, pp. 590–594, 2011.

[9] J. Zhou, K. M. Yu and B. C. Wu, “Parallel Frequent Patterns Mining Algorithm on GPU,” Proc. of the 2010 IEEE International Conference on Systems Man and Cybernetics, 2010.

[10] S. Rathi and C. A. Dhote, “Parallel Implementation of FP Growth Algorithm on XML Data Using Multiple GPU,” 2nd International Conference on Information Systems Design and Intelligent Applications, vol. 339, pp. 581-589, 2015.

[11] H. Li, Y. Wang and D. Zhang, “PFP: Parallel FP-Growth for Query Recommendation,” ACM Conference on Recommender Systems, pp. 107-114, 2008.

Aloritman FP-Growth

Algoritma FP-Growth merupakan pengembangan dari algoritma Apriori. Sehingga kekurangan dari algoritma Apriori diperbaiki oleh algoritma FPGrowth. Frequent Pattern Growth adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (frequent itemset) dalam sebuah kumpulan data.

Pada algoritma Apriori diperlukan generate candidate untuk mendapatkan frequent itemsets. Akan tetapi, di algoritma FP-Growth generate candidate tidak dilakukan karena FP-Growth menggunakan konsep pembangunan tree dalam pencarian frequent itemsets. Hal tersebutlah yang menyebabkan algoritma FP-Growth lebih cepat dari algoritma Apriori. Karakteristik algoritma FPGrowth adalah struktur data yang digunakan adalah tree yang disebut dengan FP-Tree. Dengan menggunakan FP-Tree, algoritma FP-Growth dapat langsung mengekstrak frequent itemset dari FP-Tree. Penggalian itemset yang frequent dengan menggunakan algoritma FP-Growth akan dilakukan dengan cara membangkitkan struktur data tree atau disebut dengan FP-Tree. Metode FP-Growth dapat dibagi menjadi 3 tahapan utama yaitu sebagai berikut:

1. Tahap Pembangkitan conditional pattern base,
2. Tahap pebangkitan conditional FP-Tree, dan
3. Tahap pencarian frequent itemset.

PENERAPAN METODE *DATA MINING MARKET BASKET ANALYSIS* TERHADAP DATA PENJUALAN PRODUK BUKU DENGAN MENGGUNAKAN ALGORITMA APRIORI DAN *FREQUENT PATTERN GROWTH* (FP-GROWTH) : STUDI KASUS PERCETAKAN PT. GRAMEDIA

[1]Goldie Gunadi, [2]Dana Indra Sensuse

[1]Magister Ilmu Komputer Universitas Budi Luhur
send2goldie@gmail.com

[2]Ilmu Komputer Universitas Indonesia
dana@cs.ui.ac.id

ABSTRAK

Data mining merupakan proses analisa data untuk menemukan suatu pola dari kumpulan data tersebut. Data mining mampu menganalisa data yang besar menjadi informasi berupa pola yang mempunyai arti bagi pendukung keputusan. Salah satu teknik data mining yang dapat digunakan adalah association data mining atau yang biasa disebut dengan istilah market basket analysis. Market basket didefinisikan sebagai suatu itemset yang dibeli secara bersamaan oleh pelanggan dalam suatu transaksi. Market basket analysis adalah suatu alat yang ampuh untuk pelaksanaan strategi cross-selling. Metode ini dimulai dengan mencari sejumlah frequent itemset dan dilanjutkan dengan pembentukan aturan-aturan asosiasi (association rules). Algoritma Apriori dan frequent pattern growth (FP-growth) adalah dua algoritma yang sangat populer untuk menemukan sejumlah frequent itemset dari data-data transaksi yang tersimpan dalam basis data. Dalam penelitian ini algoritma Apriori dan frequent pattern growth (FP-growth) digunakan untuk membantu menemukan sejumlah aturan asosiasi dari basis data transaksi penjualan produk buku di Percetakan PT. Gramedia, sehingga untuk selanjutnya dapat digunakan sebagai pertimbangan dalam membuat strategi pemasaran dan penjualan yang efektif.

Kata kunci : *data mining*, *association rules*, *market basket analysis*, Apriori, *frequent pattern growth,* FP-*growth*

1. Pendahuluan

Perkembangan teknologi informasi telah memberikan kontribusi pada cepatnya pertumbuhan jumlah data yang dikumpulkan dan disimpan dalam basis data berukuran besar (gunung data). Dibutuhkan sebuah metode atau teknik yang dapat merubah gunungan data tersebut menjadi sebuah informasi berharga atau pengetahuan (*knowledge*) yang bermanfaat untuk mendukung pengambilan keputusan bisnis. Suatu teknologi yang dapat digunakan untuk mewujudkannya adalah *data mining*. *Data mining* adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Salah satu metode yang seringkali digunakan dalam teknologi *data mining* adalah metode asosiasi atau *association rule mining*. Di dalam bidang usaha *retail* metode *association rule mining* ini lebih dikenal dengan istilah analisa

keranjang belanja (*market basket analysis*). *Market basket analysis* adalah suatu metode analisa atas perilaku konsumen secara spesifik dari suatu golongan / kelompok tertentu. Sumber data dari *market basket analysis* antara lain dapat bersumber dari transaksi kartu kredit, kartu lotere, kupon diskon, panggilan keluhan pelanggan. *Market basket analysis* umumnya dimanfaatkan sebagai titik awal pencarian pengetahuan dari suatu transaksi data ketika kita tidak mengetahui pola spesifik apa yang kita cari. Kebutuhan *market basket analysis* berawal dari keakuratan dan manfaat yang dihasilkannya dalam wujud aturan assosiasi (*association rules*). Yang dimaksud dengan *association rules* adalah pola-pola keterkaitan data dalam basis data.

Percetakan PT. Gramedia adalah sebuah perusahaan yang bergerak di bidang industri percetakan dan telah beroperasi sejak tahun 1972. Hingga kini Percetakan PT. Gramedia telah melayani sekian banyak transaksi pesanan cetakan. Setiap data transaksi tersebut disimpan di dalam suatu sistem basis data melalui aplikasi sistem informasi manajemen.

Dalam penelitian ini penulis akan mencoba untuk melakukan suatu eksperimen terhadap data transaksi penjualan produk buku di Percetakan PT. Gramedia dengan menggunakan sebuah metode *market basket analysis* untuk melihat hubungan asosiasi (korelasi) antara sejumlah atribut penjualan. Algoritma yang akan digunakan adalah algoritma Apriori dan algoritma *frequent pattern growth* (FP-growth). Kedua algoritma ini digunakan untuk membentuk *frequent itemsets* yang nantinya akan dijadikan sebagai acuan untuk merumuskan aturan-aturan asosiasi yang dihasilkan oleh model *market basket analysis* dengan menggunakan alat bantu perangkat lunak aplikasi Weka versi 3.6. Aturan-aturan asosiasi yang dihasilkan oleh masing-masing algoritma Apriori dan FP-growth akan dievaluasi dan dianalisa untuk menemukan algoritma yang paling baik dan sesuai (*best-fit algorithm*) untuk permasalahan yang dikaji dalam penelitian ini.

Komponen-komponen penjualan utama yang akan digunakan sebagai acuan didalam penentuan hubungan asosiasi dengan menggunakan *market basket analysis* adalah :

a. Jenis *cover* buku
b. Teknik penjilidan
c. Penggunaan plastik buku
d. Penggunaan jenis kertas khusus untuk *cover* buku
e. Penggunaan jenis kertas khusus untuk isi buku
f. Pekerjaan-pekerjaan dalam pembuatan *cover* seperti : UV, *laminating*, *embossed*, dan *hot print*.

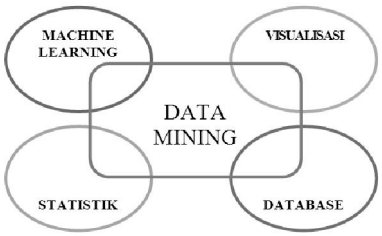
Dengan dilakukannya penelitian ini, diharapkan dapat memberikan hasil berupa informasi yang bermanfaat bagi pihak-pihak terkait dalam melakukan proses pengambilan keputusan manajerial, terutama yang berkaitan dengan perumusan atau pembuatan strategi pemasaran dan penjualan, khususnya untuk produk-produk buku di Percetakan PT. Gramedia.

2. Landasan Pemikiran

2.1. Tinjauan Pustaka

2.1.1. Data Mining

Menurut Turban dalam bukunya yang berjudul "*Decision Support Systems and Intelligent Systems*", *data mining* adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam basis data. *Data mining* adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai basis data besar [1].



Gambar 1. *Data mining* merupakan irisan dari berbagai disiplin [2].

2.1.2. Pengelompokan *Data Mining*

Menurut Larose dalam bukunya yang berjudul "*Discovering Knowledge in Data: An Introduction to Data Mining*", *data mining* dibagi menjadi beberapa kelompok berdasarkan tugas/pekerjaan yang dapat dilakukan, yaitu [1]:

a. Deskripsi
   Terkadang peneliti dan analisis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Deskripsi dari pola kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.
b. Estimasi
   Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik dari pada ke arah kategori. Model dibangun menggunakan baris data (*record*) lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi.
c. Prediksi
   Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang. Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.
d. Klasifikasi
   Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.
e. Pengklasteran (*Clusterring*)
   Pengklasteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas obyek-obyek yang memiliki kemiripan. Klaster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan *record* dalam klaster yang lain.
   Berbeda dengan klasifikasi, pada pengklasteran tidak ada variabel target. Pengklasteran tidak melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel target, akan tetapi, algoritma pengklasteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogen), yang mana kemiripan *record* dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal.
f. Asosiasi
   Tugas asosiasi dalam *data mining* adalah untuk menemukan atribut yang muncul dalam satu waktu. Salah satu implementasi dari asosiasi adalah *market basket analysis* atau analisis keranjang belanja, sebagaimana yang akan dibahas dalam tesis ini.

2.1.3. *Knowledge Discovery in Databases*

KDD terdiri dari tiga proses utama yaitu:

a. *Preprocessing*
   Preprocessing dilakukan terhadap data sebelum algoritma *data mining* diaplikasikan. Proses ini meliputi *data cleaning*, integrasi, seleksi dan transformasi.
b. *Data mining*
   Proses utama dalam KDD adalah proses *data mining*, dalam proses ini algoritma-algoritma *data mining* diaplikasikan untuk mendapatkan pengetahuan dari sumber data.
c. *Post processing*
   Hasil yang diperoleh dari proses *data mining* selanjutnya akan dievaluasi pada *post processing*.

2.1.4. *Market Basket Analysis*

Market basket analysis adalah suatu metodologi untuk melakukan analisis *buying habit* konsumen dengan menemukan asosiasi antar beberapa *item* yang berbeda, yang diletakkan konsumen dalam *shopping basket* (keranjang belanja) yang dibeli pada

suatu transaksi tertentu. Tujuan dari *market basket analysis* adalah untuk mengetahui produk-produk mana yang mungkin akan dibeli secara bersamaan [3].

2.1.5. Analisis Asosiasi

Analisis asosiasi atau *association rule mining* adalah teknik *data mining* untuk menemukan aturan asosiasi antara suatu kombinasi *item* [1]. *Interestingness measure* yang dapat digunakan dalam *data mining* adalah :

a. *Support*, adalah suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *item* atau *itemset* dari keseluruhan transaksi.
b. *Confidence*, adalah suatu ukuran yang menunjukkan hubungan antar dua *item* secara *conditional* (berdasarkan suatu kondisi tertentu).

Metodologi dasar analisis asosiasi terbagi menjadi 2 tahap, yakni melakukan analisa pola frekuensi tinggi (*frequent pattern*) dan berikutnya adalah proses pembentukan aturan asosiasi.

2.1.6. Algoritma Apriori

Algoritma Apriori adalah suatu algoritma dasar yang diusulkan oleh Agrawal & Srikant pada tahun 1994 untuk penentuan *frequent itemsets* untuk aturan asosiasi *boolean* [3].

Bentuk algoritma dari metode Apriori dapat dituliskan sebagai berikut [3]:

L1 = {frequent itemset with one element}
for ($k=2; L_{k-1} \neq \emptyset; k++$)
{
C_k = *apriori-gen*(L_{k-1}); *//pembuatan kandidat*
//baru
for all transactions t
{
C'_t = *subset*(C_k, *t*); *//kandidat yang //tampil pada t*
for all candidates $c \in C'_t$ *do*
c.count ++;
}
$L_k = \{ c \in C'_t \mid c.count \geq minsup \}$
}
return $\cup_k L_k$;

Dimana :

L : himpunan frequent itemset
minsup : minimum support
C : himpunan kandidat itemset
c : kandidat itemset
t : transaksi

2.1.7. Analisis Pola Frekuensi Tinggi dengan Algoritma Apriori

Mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam basis data.

Nilai *support* sebuah *item* diperoleh dengan menggunakan rumus berikut:

$$Support(A) = \frac{Jumlah\ transaksi\ mengandung\ A}{Total\ Transaksi}$$

Nilai *support* dari 2 *item* diperoleh dengan menggunakan rumus :

$$Support(A, B) = P(A \cap B)$$

$$Support(A, B) = \frac{\sum Transaksi\ mengandung\ A\ dan\ B}{\sum Transaksi}$$

Frequent itemset menunjukkan *itemset* yang memiliki frekuensi kemunculan lebih dari nilai minimum yang ditentukan (Φ). Misalkan Φ = 2, maka semua *itemsets* yang frekuensi kemunculannya lebih dari atau sama dengan 2 kali disebut *frequent*. Himpunan dari *frequent* k-*itemset* dilambangkan dengan F_k.

2.1.8. Algoritma FP-Growth

FP-growth adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. FP-growth menggunakan pendekatan yang berbeda dari paradigma yang digunakan pada algoritma Apriori.

2.1.9. Analisis Pola Frekuensi Tinggi dengan Algoritma FP-growth

Pada penentuan *frequent itemset* terdapat 2 tahap proses yang dilakukan yaitu: pembuatan FP-tree dan penerapan algoritma FP-growth untuk menemukan *frequent itemset*. Struktur data yang digunakan untuk mencari *frequent itemset*

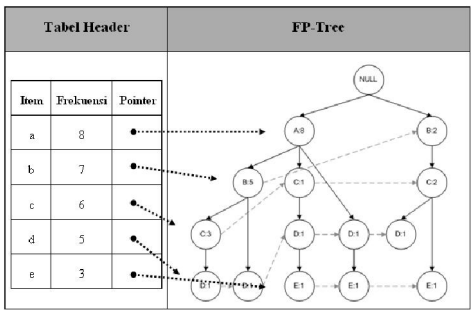
dengan algoritma FP-growth adalah perluasan dari penggunaan sebuah pohon *prefix*, yang biasa disebut adalah FP-tree. Dengan menggunakan FP-*tree*, algoritma FP-growth dapat langsung mengekstrak *frequent itemset* dari FP-*tree* yang telah terbentuk dengan menggunakan prinsip *divide and conquer*.

2.1.9.1.Pembuatan FP-*Tree*

FP-*tree* merupakan struktur penyimpanan data yang dimampatkan. FP-tree dibangun dengan memetakan setiap data transaksi ke dalam setiap lintasan tertentu dalam FP-*tree*. Karena dalam setiap transaksi yang dipetakan, mungkin ada transaksi yang memiliki *item* yang sama, maka lintasannya memungkinkan untuk saling menimpa. Semakin banyak data transaksi yang memiliki *item* yang sama, maka proses pemampatan dengan struktur data FP-tree semakin efektif.

Adapun FP- tree adalah sebuah pohon dengan definisi sebagai berikut:

a. FP-tree dibentuk oleh sebuah akar yang diberi label *null*, sekumpulan *sub-tree* yang beranggotakan *item-item* tertentu, dan sebuah tabel *frequent header*.
b. Setiap simpul dalam FP-tree mengandung tiga informasi penting, yaitu label *item*, menginformasikan jenis *item* yang direpresentasikan simpul tersebut, *support count*, merepresentasikan jumlah lintasan transaksi yang melalui simpul tesebut, dan *pointer* penghubung yang menghubungkan simpul-simpul dengan label *item* sama antar-lintasan, ditandai dengan garis panah putus-putus.



Gambar 2. Contoh FP-tree dan Tabel *Header*

2.1.9.2. Penerapan Algoritma FP-*Growth*

Setelah tahap pembangunan FP-tree dari sekumpulan data transaksi, akan diterapkan algoritma FP-*growth* untuk mencari *frequent itemset* yang signifikan. Algoritma FP-*growth* dibagi menjadi tiga langkah utama, yaitu :

a. Tahap Pembangkitan *Conditional Pattern Base*
   Conditional Pattern Base merupakan *sub-database* yang berisi *prefix path* (lintasan prefix) dan *suffix pattern* (pola akhiran). Pembangkitan *conditional pattern base* didapatkan melalui FP-tree yang telah dibangun sebelumnya.
b. Tahap Pembangkitan *Conditional FP-tree*
   Pada tahap ini, *support count* dari setiap *item* pada setiap *conditional pattern base* dijumlahkan, lalu setiap *item* yang memiliki jumlah *support count* lebih besar sama dengan *minimum support count* ξ akan dibangkitkan dengan *conditional* FP-tree.
c. Tahap Pencarian *frequent itemset*
   Apabila *Conditional FP-tree* merupa-kan lintasan tunggal (*single path*), maka didapatkan *frequent itemset* dengan melakukan kombinasi *item* untuk setiap *conditional* FP-*tree*. Jika bukan lintasan tunggal, maka dilakukan pembangkitan FP-*growth* secara rekursif.

2.1.10. Pembentukan Aturan Asosiasi

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiasi yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan asosiatif $A \Rightarrow B$.

Nilai *confidence* dari aturan $A \Rightarrow B$ diperoleh dengan rumus berikut.

$$Confidence = P(B \mid A) = \frac{\sum Transaksi\ mengandung\ A\ dan\ B}{\sum Transaksi\ mengandung\ A}$$

Untuk menentukan aturan asosiasi yang akan dipilih maka harus diurutkan berdasarkan *Support* × *Confidence*. Aturan diambil sebanyak *n* aturan yang memiliki hasil terbesar [1].

2.2. Tinjauan Studi

Dalam [4] diukur nilai *confidence* dari setiap aturan asosiasi yang dihasilkan. Aturan asosiasi yang diambil adalah aturan-aturan asosiasi yang memiliki nilai *confidence* lebih besar atau sama dengan nilai *minimum confidence* yang telah ditentukan .

Dalam tulisannya, [5] menggunakan hasil dari nilai *Support* dikalikan dengan *Confidence* terhadap masing-masing aturan asosiasi yang dihasilkan untuk mengukur kekuatan dari aturan asosiasi yang bersangkutan.

Dalam penelitiannya yang berjudul "*A Scalable Algorithm for the Market Basket Analysis*", Luís Cavique melakukan perbandingan performa terhadap akurasi dari dua algoritma asosiasi yaitu algoritma similis dan FP-growth, dengan menghitung tingkat akurasinya.

Fungsi yang digunakan untuk menghitung persentase akurasi dari algoritma similis terhadap FP-growth adalah :

$$\frac{\sum \text{Similis Support}}{\sum \text{FP} - \text{growth Support}} \times 100$$

2.3. Tinjauan Obyek Penelitian

2.3.1. Percetakan PT. Gramedia

Percetakan PT. Gramedia berdiri sejak tahun 1972, bergerak di bidang layanan jasa cetak koran, tabloid, buku, majalah dan material promosi. Disamping melayani pelanggan dalam kelompok Kompas Gramedia, juga telah dipercaya oleh pelanggan dari luar kelompok Kompas Gramedia untuk mencetak produk-produknya.

2.3.2. Visi dan Misi Perusahaan

Visi yang ingin dicapai oleh Percetakan PT. Gramedia adalah: "Menjadi percetakan kelas dunia terbaik di Asia Tenggara."

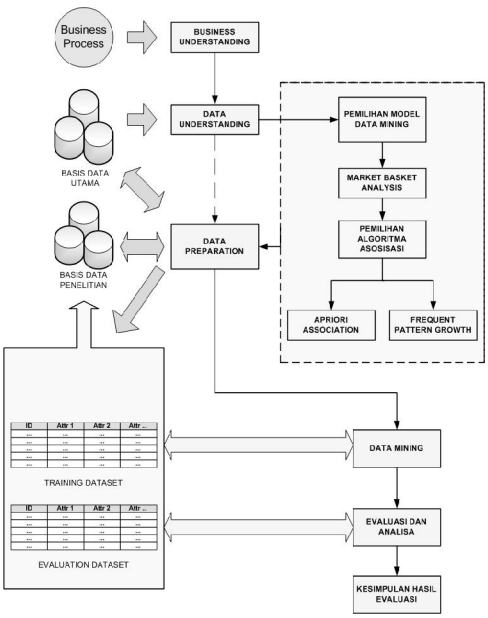
Misi dari Percetakan PT. Gramedia adalah :

a. Mendukung pencapaian sasaran bisnis pelanggan dengan memahami bisnis, menjamin keamanan produk, memberi kualitas yang dipilih, memberikan inovasi produk dan layanan yang memuaskan pelanggan.
b. Menyediakan layanan disain, percetakan, pengemasan dan layanan distribusi yang paling disukai pelanggan untuk mencerahkan dan mencerdaskan masyarakat.

2.4. Kerangka Pemikiran

Kerangka pemikiran penelitian *market basket analysis*, meliputi :

a. *Business Understanding*
b. Data Understanding
c. Pemilihan Model *Data Mining*
d. Pemilihan Algoritma Asosiasi
e. Data Preparation
f. Data Mining
g. Evaluasi dan Analisa



Gambar 3. Kerangka pemikiran penelitian *market basket analysis*

2.4.1. Evaluasi

Ada 2 faktor yang akan dievaluasi dalam penelitian ini, yaitu ukuran generalitas (*generality*) dan ukuran reliabilitas (*reliability*) dari *association rules* yang dihasilkan.

Ukuran generalitas digunakan untuk mengetahui seberapa tingkat kemunculan setiap *item* yang dirumuskan dalam aturan

asosiasi terhadap keseluruhan transaksi. Ukuran-ukuran yang digunakan adalah :

a. *Support*
Support adalah ukuran dari seberapa sering koleksi *item* dalam asosiasi terjadi bersama sebagai persentase dari seluruh transaksi.
Rumus :
$$s(A \Rightarrow B) = P(A \cap B)$$
dimana :
$$P(A \cap B) = \frac{\sum Transaksi\ mengandung\ A\ dan\ B}{\sum Transaksi}$$

b. *Coverage*
Coverage adalah ukuran seberapa sering koleksi setiap *item* yang dirumuskan pada sisi sebelah kiri (faktor penyebab) terjadi sebagai persentase dari seluruh transaksi.
Rumus :
$$P(A) = \frac{\sum Transaksi\ mengandung\ A}{\sum Transaksi}$$

Ukuran reliabilitas digunakan untuk mengetahui tingkat kehandalan dari aturan asosiasi yang dihasilkan dalam *association rules mining*. Ukuran-ukuran yang digunakan adalah :

a. *Confidence*
Confidence adalah ukuran ketidakpastian atau kepercayaan kelayakan yang terkait dengan setiap pola yang ditemukan.
Rumus :
$$\alpha(A \Rightarrow B) = P(B|A)$$
dimana :
$$P(B|A) = \frac{\sum Transaksi\ mengandung\ A\ dan\ B}{\sum Transaksi\ mengandung\ A}$$

b. *Added Value*
Added Value digunakan untuk mengukur berapa besar selisih antara nilai *confidence* dengan peluang terjadinya kombinasi setiap *item* pada sisi akibat pada aturan asosiasi terhadap seluruh transaksi.
Added Value $(A \Rightarrow B) = P(B|A) - P(B)$

c. *Correlation*
Correlation digunakan untuk menentukan apakah antara *itemset* yang satu dengan yang lainnya memiliki suatu ketergantungan atau tidak.
Correlation $(A \Rightarrow B)$ bernilai 0 jika A dan B adalah bersifat independen (tidak memiliki ketergantungan), dan 1 (nilai positif, >0) atau -1 (nilai negatif, <0) jika mereka tergantung. 1 menunjukkan daya tarik, dan -1 menunjukkan tolakan.
Rumus :

Correlation $(A \Rightarrow B) = P(A \cap B)/[P(A)P(B)]$
dimana :
P(A) = ∑Transaksi mengandung A
P(B) = ∑Transaksi mengandung B

2.4.2. Analisa

Analisa dilakukan terhadap sejumlah aturan asosiasi dengan nilai *confidence* terbesar yang dihasilkan oleh masing-masing algoritma Apriori dan FP-growth. Analisa dilakukan dengan menghitung tingkat kekuatan dan akurasi dari masing-masing aturan yang diimplementasikan terhadap *evaluation dataset* data transaksi penjualan terkini.

Untuk menghitung tingkat kekuatan dilakukan dengan rumus :
$$\frac{\sum_{i=1}^{n}(S_i \times C_i)}{n}$$
dimana:
n = jumlah aturan asosiasi
Si = nilai support untuk aturan asosiasi ke-i
Ci = nilai *confidence* untuk aturan asosiasi ke-i

Untuk menghitung besar persentase akurasi algoritma A terhadap algoritma B dilakukan dengan menggunakan rumus :
$$\frac{\sum Support\ Algoritma\ A}{\sum Support\ Algoritma\ B} \times 100$$

3. Rancangan Penelitian

3.1. Data Penelitian

Data yang akan digunakan di dalam penelitian ini adalah data transaksi pesanan cetakan di Percetakan PT. Gramedia yang terkumpul mulai dari tanggal 1 Januari 2005 hingga 1 Agustus 2010.

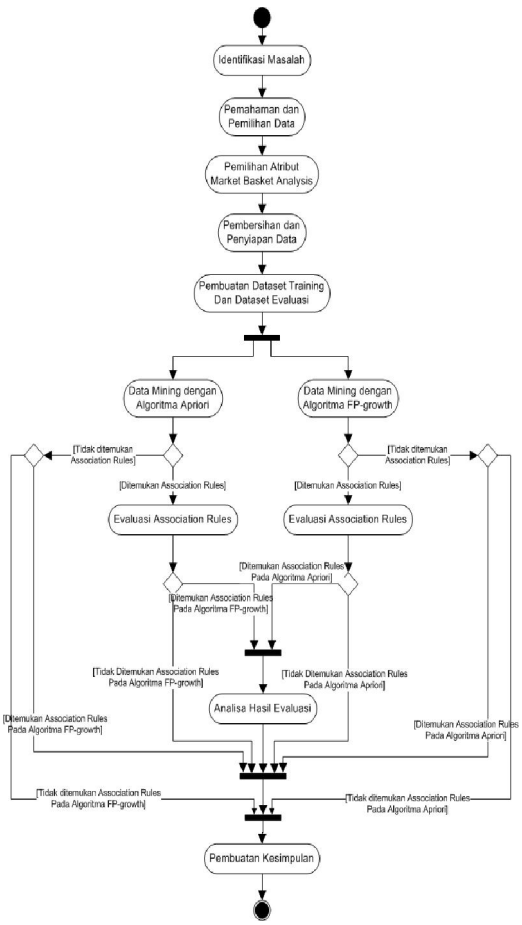
Tabel-tabel yang digunakan sebagai obyek penelitian ini adalah :

Tabel 1. Tabel obyek penelitian

No	Nama Tabel	*Tuples*
1	Tbl_Master_Order	147498
2	Tbl_Master_Pelanggan	2038
3	Tbl_Rincian_Order	258278
4	Tbl_Order_Sortir_Pak	143620
5	Tbl_Jenis_Permintaan	6
6	Tbl_Jenis_Produk	18
7	Tbl_Nama_Kertas	182
8	Tbl_Teknik_Jilid	14
9	Tbl_Jenis_Cover	7
10	Tbl_Jenis_Laminating	6
11	Tbl_Jenis_UV	4
12	Tbl_Master_Site	4
13	Tbl_Nama_Komponen	5

3.2. Kerangka Kerja

Kerangka kerja penelitian dapat dilihat dalam bentuk diagram aktivitas berikut:



Gambar 4. Diagram aktivitas penelitian

3.3. Perangkat Lunak

Aplikasi-aplikasi yang digunakan untuk membantu proses penelitian *market basket analysis* terdiri dari :

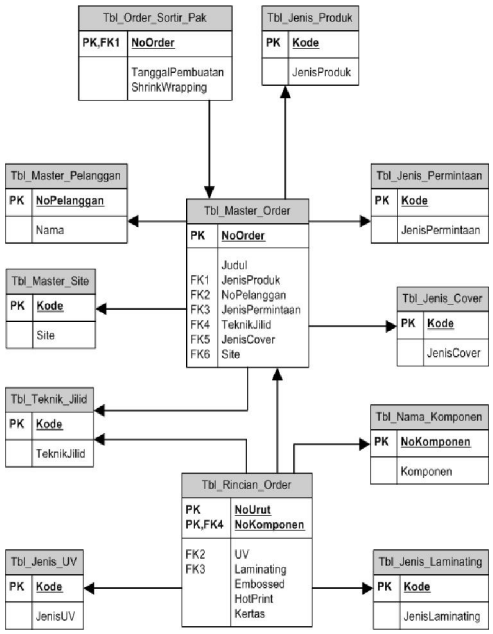
a. SQL Server 2005 Express Edition
   Dalam penelitian ini SQL Server 2005 Express Edition digunakan untuk menyimpan basis data yang akan digunakan dalam penelitian sebagai hasil dari proses pemilihan data penelitian. Setelah itu dilakukan proses pembersihan dan penyiapan data serta pembuatan *dataset* penelitian dengan menggunakan perangkat SQL Server Management Studio Express.

b. Weka 3.6
   Aplikasi Weka digunakan dalam penelitian untuk mengimplementa-sikan model *data mining* terhadap basis data penelitian. Weka juga digunakan untuk menampilkan presentasi knowledge yang dapat disimpulkan dari hasil penerapan model *data mining* terhadap sejumlah atribut penjualan produk buku di Percetakan PT. Gramedia.

4. Analisis Dan Interpretasi

4.1. Entity Relationship Diagram (ERD)

Hubungan antar tabel yang digunakan dapat digambarkan dengan menggunakan sebuah diagram relasi antar entitas (*Entity Relationship Diagram*).



Gambar 5. *Entity Relationship Diagram*

4.2. Pemilihan Data

Pemilihan data mengacu pada permasalahan yang telah diidentifikasi sebelumnya.

Hasil dari proses pemilihan data terhadap masing-masing tabel adalah :

Tabel 1. Tabel hasil pemilihan data

Nama Tabel	*Tuples*
Tbl_Master_Order_Ex	36558
Tbl_Rincian_Order_Ex	69673
Tbl_Order_Sortir_Pak_Ex	35875
Tbl_Master_Pelanggan_Ex	690
Tbl_Jenis_Permintaan_Ex	2
Tbl_Jenis_Produk_Ex	4
Tbl_Nama_Kertas_Ex	54
Tbl_Teknik_Jilid_Ex	9
Tbl_Jenis_Cover_Ex	5
Tbl_Jenis_Laminating_Ex	4
Tbl_Jenis_UV_Ex	3
Tbl_Master_Site_Ex	4
Tbl_Nama_Komponen_Ex	3

4.3. Pemilihan Atribut

Dari setiap *item* data akan dihitung besar persentase jumlah frekuensi kemunculan *item* tersebut terhadap jumlah transaksi keseluruhan. Penentuan nilai batas minimum (*Treshold*) sebesar 0,2 %, sehingga setiap *item* data yang memiliki persentase kemunculan melebihi atau sama dengan 0,2 % akan terpilih menjadi atribut.

Data-data persentase dari masing-masing komponen dapat dilihat pada sejumlah tabel berikut :

Tabel 2. Data frekuensi kandidat atribut *cover* buku

Item	Frek	Persen
Soft Cover	**34678**	**96,54**
Semi Hard Cover	**57**	**0,16**
Hard Cover	**1185**	**3.3**
Total	**35920**	**100**

Tabel 3. Data frekuensi kandidat atribut teknik penjilidan

Item	Frek	Persen
Benang	**3**	**0,008**
Kawat Spiral	**230**	**0,63**
Punch	**8**	**0,02**
Kawat	**5127**	**14,05**
Lem	**29075**	**79,67**
Lem + Benang	**1962**	**5,38**
Lipat	**10**	**0,03**
Sisip	**1**	**0,002**
Potong Jadi	**77**	**0,21**
Total	**36493**	**100**

Tabel 4. Data frekuensi kandidat atribut teknik UV

Item	Frek	Persen
UV Doft	**16798**	**47,76**
UV Spot	**4765**	**13,55**
Lainnya	**13608**	**38,69**
Total	**35171**	**100**

Tabel 5. Data frekuensi kandidat atribut teknik *laminating*

Item	Frek	Persen
Laminating Gloss	**2865**	**8,15**
Laminating Doft	**7344**	**20,88**
Lainnya	**24962**	**70,97**
Total	**35171**	**100**

Tabel 6. Data frekuensi kandidat atribut teknik *embossed* dan *hot print*

Item	Frek	Persen
Embossed	**2877**	**8,18**
Hot Print	**295**	**0,8**

Tabel 7. Data frekuensi kandidat atribut penggunaan plastik *shrink*

Item	Frek	Persen
Shrink	**26992**	**75,24**

Tabel 8. Data frekuensi kandidat atribut jenis kertas *cover* buku

Item	**Frek**	**Persen**
Art Paper	**2601**	**7,4**
Karton Board	**1109**	**3,15**
Art Karton	**30489**	**86,69**
Newsprint	**9**	**0,02**
Lainnya	**963**	**2,74**
Total	**35171**	**100**

Tabel 9. Data frekuensi kandidat atribut jenis kertas halaman isi buku

Item	**Frek**	**Persen**
Art Paper	**4228**	**12,38**
Newsprint	**13307**	**39**
HVS/ Woodsfree	**1713**	**5**
HVS Sheet	**10048**	**29,42**
Koran Sheet	**4346**	**12,72**
Lainnya	**510**	**0,01**
Total	**34152**	**100**

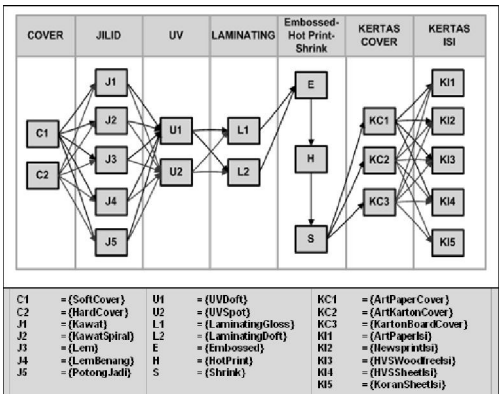
Dari hasil proses pemilihan atribut, dapat diperoleh sejumlah atribut yang memenuhi syarat *treshold* minimum dan dapat digunakan dalam *market basket analysis*. Seluruh atribut penelitian yang merupakan hasil seleksi dari setiap komponen penjualan produk buku dapat dilihat pada tabel berikut :

Tabel 10. Data frekuensi atribut terpilih

Komponen	**Atribut Terpilih**	**Jumlah**
Jenis *cover*	{Soft Cover}, {Hard Cover}	2
Teknik jilid	{Kawat}, {Lem}, {Lem + Benang}, {Kawat Spiral}, {Potong Jadi}	5
Shrink	{Shrink}	1
Jenis kertas *cover*	{Art Paper}, {Karton Board}, {Art Karton}	3
Jenis kertas isi	{Art Paper}, {Newsprint}, {HVS/Woodfree}, {HVS Sheet}, {Koran Sheet}	5
UV	{UV Doft }, {UV Spot}	2
Laminating	{Laminating Gloss}, {Laminating Doft}	2
Embossed	{Embossed}	1
Hot Print	{Hot Print}	1

4.3.1. Kombinasi Atribut

Kombinasi atribut yang dapat digunakan untuk pencarian aturan asosiasi tampak seperti pada gambar dibawah ini :



Gambar 6. Kombinasi atribut penelitian

4.3.2. Penentuan Atribut Penelitian

Dari sekian banyak kombinasi atribut yang ada, kombinasi atribut yang nantinya akan digunakan dalam penelitian ini adalah kombinasi dari atribut-atribut yang memiliki persentase kemunculan paling besar dari masing-masing komponen.

Tabel 11. Atribut yang digunakan dari masing-masing komponen

Kompo-nen	**Atribut Terpilih**	**Atribut Penelitian**
Jenis *cover*	{SoftCover = 6,54%}, {HardCover = 3,3%}	{SoftCover}
Teknik jilid	{Kawat = 14,05%}, {Lem = 79,67%}, {LemBenang = 5,38%}, {KawatSpiral = 0,63%}, {PotongJadi = 0,21%}	{Lem}
UV	{UVDoft = 47,76%}, {UVSpot = 13,55%}	{UVDoft}
Lamina-ting	{LaminatingGloss = 8,15%}, {LaminatingDoft = 20,88%}	{Laminating Doft}
Emboss-ed	{Embossed = 8,18%}	{Embossed}
Hot Print	{HotPrint = 0,8 %}	{HotPrint}
Shrink	{Shrink = 75,24%}	{Shrink}
Jenis kertas *cover*	{ArtPaperCover = 7,4 %}, {KartonBoardCover = 3,15%}, {ArtKartonCover = 86,69%}	{ArtKarton Cover}
Jenis kertas isi	{ArtPaperIsi = 12,38%}, {NewsprintIsi = 39%}, {HVSWoodfreeIsi = 5%}, {HVSSheetIsi = 29,42%}, {KoranSheetIsi = 12,72%}	{Newsprint Isi}

Berdasarkan nilai persentase terbesar dari masing-masing atribut, dengan

demikian atribut-atribut yang akan digunakan dalam penelitian adalah : jenis *cover hard cover*, teknik penjilidan dengan menggunakan lem, proses : UV *doft*, laminating *doft*, *embossed* dan *hot print*, penggunaan plastik *shrink*, jenis kertas Art Karton untuk *cover* buku serta jenis kertas Newsprint untuk halaman isi buku.

4.4. Pembersihan dan Penyiapan Data

Perintah SQL yang digunakan untuk melakukan pembersihan data adalah perintah DML SQL.

Berikut adalah jumlah *tuples* terakhir dari masing-masing tabel transaksi setelah dilakukannya pembersihan data :

a. Tbl_Master_Order : 32821 *Tuples*
b. Tbl_Rincian_Order : 69996 Tuples
c. Tbl_Order_Sortir : 32821 Tuples

4.5. Pembuatan Dataset

Dataset dibuat berdasarkan atribut-atribut yang telah dipilih pada data transaksi penjualan produk buku.

Berikut adalah struktur *dataset* yang dibuat :

Tabel 12. Struktur *dataset* penelitian

Nama Kolom	Tipe Data	Atribut Kolom	Keterangan
ID	*INTE-GER*	*Primary Key*	ID transaksi penjualan
SoftCover	*CHAR(1)*	*Not Null*	Atribut jenis *cover*
HardCover	*CHAR(1)*	*Not Null*	Atribut Jenis *cover*
Kawat	*CHAR(1)*	*Not Null*	Atribut teknik penjilidan
KawatSpiral	*CHAR(1)*	*Not Null*	Atribut teknik penjilidan
Lem	*CHAR(1)*	*Not Null*	Atribut teknik penjilidan
LemBenang	*CHAR(1)*	*Not Null*	Atribut teknik penjilidan
PotongJadi	*CHAR(1)*	*Not Null*	Atribut teknik penjilidan
UVDoff	*CHAR(1)*	*Not Null*	Atribut jenis UV
UVSpot	*CHAR(1)*	*Not Null*	Atribut jenis UV
LaminatingGloss	*CHAR(1)*	*Not Null*	Atribut jenis *laminating*
LaminatingDoft	*CHAR(1)*	*Not Null*	Atribut jenis *laminating*
Embossed	*CHAR(1)*	*Not Null*	Atribut *embossed*
HotPrint	*CHAR(1)*	*Not Null*	Atribut *hotprint*
Shrink	*CHAR(1)*	*Not Null*	Atribut *shrink*
ArtPaperCover	*CHAR(1)*	*Not Null*	Atribut kertas cover
ArtKartonCover	*CHAR(1)*	*Not Null*	Atribut kertas cover
KartonBoardCover	*CHAR(1)*	*Not Null*	Atribut kertas cover
ArtPaperIsi	*CHAR(1)*	*Not Null*	Atribut kertas halaman isi
NewsprintIsi	*CHAR(1)*	*Not Null*	Atribut kertas halaman isi
HVSWoodfreeIsi	*CHAR(1)*	*Not Null*	Atribut kertas halaman isi
HVSSheetIsi	*CHAR(1)*	*Not Null*	Atribut kertas halaman isi
KoranSheetIsi	*CHAR(1)*	*Not Null*	Atribut kertas halaman isi

Dataset dengan struktur tersebut diatas, dibuat sebanyak 3 buah dalam bentuk tabel dengan nama Tbl_Dataset_Training, Tbl_Dataset_Evaluation1, dan Tbl_Dataset_Evaluation2. Masing-masing *dataset* tersebut kemudian diisi dengan data-data sebagai berikut :

a. Tabel Tbl_Dataset_Training diisi dengan seluruh data transaksi penjualan yang terdapat pada basis data penelitian.
b. Tabel Tbl_Dataset_Evaluation1 diisi dengan 5000 data transaksi penjualan terakhir yang terdapat pada basis data penelitian.
c. Tabel Tbl_Dataset_Evaluation2 diisi dengan 10000 data transaksi penjualan yang diambil secara acak dari basis data penelitian. Data acak yang diambil adalah data-data transaksi penjualan mulai tanggal 1 Januari 2008.

4.6. Data Mining

Proses data mining dilakukan terhadap tabel Tbl_Dataset_Training dengan menggunakan aplikasi Weka versi 3.6. Sepuluh aturan asosiasi dengan nilai *confidence* terbesar diambil untuk dievaluasi dan dianalisa dalam penelitian ini.

4.6.1. Aturan Asosiasi Algoritma Apriori

Aturan-aturan asosiasi yang diperoleh dari penerapan *market basket analysis* dengan algoritma apriori adalah seperti pada tabel berikut :

Tabel 13. Aturan asosiasi hasil algoritma Apriori

Urt	A	B	F_A	F_{AB}	Cnf
1	{Lem=y}, {Shrink=y}, {ArtKarton Cover=y}	{SoftCover=y}	23849	23846	1
2	{Lem=y}, {ArtKarton Cover=y}	{SoftCover=y}	26042	26035	1
3	{Lem=y}, {Shrink=y}	{SoftCover=y}	24078	24068	1
4	{Lem=y}	{SoftCover=y}	26705	26684	1
5	{Shrink=y}, {ArtKarton Cover=y}	{SoftCover=y}	24864	24814	1

Tabel 13. Aturan asosiasi hasil algoritma Apriori (Lanjutan)

Urt	A	B	F_A	F_{AB}	Cnf
6	{ArtKartonCover=y}	{SoftCover=y}	29947	29878	1
7	{SoftCover=y}, {Lem=y}, {Shrink=y}	{ArtKarton Cover=y}	24068	23846	0,99
8	{Lem=y}, {Shrink=y}	{ArtKartonCover =y}	24078	23849	0,97
9	{Lem=y}, {Shrink=y}	{SoftCover=y}, {ArtKartonCover =y}	24078	23846	0,94
10	{SoftCover=y}, {Shrink=y}	{ArtKartonCover =y}	25086	24814	0,94

Aturan-aturan asosiasi yang dihasilkan oleh algoritma Apriori ditentukan oleh sejumlah kombinasi *items* sebagai berikut :

Tabel 14. Kombinasi *items* pada aturan asosiasi Apriori

Ukuran *Itemset*	*Items*
2	{Lem, SoftCover}, {ArtKartonCover, SoftCover}
3	{Lem, ArtKartonCover, SoftCover}, {Lem, Shrink, SoftCover}, {Shrink, ArtKartonCover, SoftCover}, {Lem, Shrink, ArtKartonCover}
4	{Lem, Shrink, ArtKartonCover, SoftCover}

Dari tabel di atas dapat dilihat bahwa algoritma apriori menemukan adanya hubungan antara teknik penjilidan lem, penggunaan plastik shrink, jenis kertas Art Karton untuk *cover* buku, dan jenis *cover* soft cover, yang digambarkan oleh aturan-aturan asosiasi yang dihasilkannya.

4.6.2. Aturan Asosiasi Algoritma FP-growth

Aturan-aturan asosiasi yang diperoleh dari penerapan *market basket analysis* dengan algoritma FP-growth adalah seperti pada tabel berikut :

Tabel 15. Aturan asosiasi hasil algoritma FP-growth

Urt	A	B	F_A	F_{AB}	Cnf
1	{Shrink=y}, {UVDoff=y}, {NewsprintIsi=y}	{Lem=y}	8199	8189	1
2	{Shrink=y}, {NewsprintIsi=y}	{Lem=y}	12306	12228	0,99

Tabel 15. Aturan asosiasi hasil algoritma FP-growth (Lanjutan)

3	{Lem=y}, {UVDoff=y}, {NewsprintIsi=y}	{Shrink=y}	8336	8189	0,98
4	{Lem=y}, {NewsprintIsi=y}	{Shrink=y}	12465	12228	0,98
5	{Shrink=y}, {UVDoff=y}	{Lem=y}	13529	13247	0,98
6	{UVDoff=y}, {NewsprintIsi=y}	{Lem=y}	8540	8336	0,98
7	{UVDoff=y}, {NewsprintIsi=y}	{Shrink=y}	8540	8199	0,96
8	{NewsprintIsi=y}	{Lem=y}	12998	12465	0,96
9	{UVDoff=y}, {NewsprintIsi=y}	{Lem=y}, {Shrink=y}	8540	8189	0,96
10	{NewsprintIsi=y}	{Shrink=y}	12998	12306	0,95

Aturan-aturan asosiasi yang dihasilkan oleh algoritma FP-growth ditentukan oleh sejumlah kombinasi *items* sebagai berikut :

Tabel 16. Kombinasi *items* pada aturan asosiasi FP-growth

Ukuran *itemset*	*Items*
2	{NewsprintIsi, Lem}, {NewsprintIsi, Shrink}
3	{Shrink, NewsprintIsi, Lem}, {Shrink, UVDoff, Lem}, {UVDoff, NewsprintIsi, Lem}, {UVDoff, NewsprintIsi, Shrink}
4	{Shrink, UVDoff, NewsprintIsi, Lem}

Dari tabel di atas dapat dilihat bahwa algoritma FP-growth menemukan adanya hubungan antara penggunaan plastik *shrink*, proses UV Doft pada *cover*, jenis kertas Newsprint untuk isi halaman buku, dan teknik jilid lem, yang digambarkan oleh aturan-aturan asosiasi yang dihasilkannya.

4.7. Evaluasi Aturan Asosiasi

Evaluasi aturan asosiasi dilakukan dengan menggunakan 2 buah *evaluation dataset*, yang telah disiapkan sebelumnya. Evaluasi dilakukan dengan mengukur generalitas dan reliabilitas dari masing-masing aturan yang dihasilkan.

4.7.1. Evaluasi Aturan Asosiasi Apriori

Hasil evaluasi terhadap *evaluation dataset* yang tersimpan pada tabel Tbl_Dataset_Evaluation1 dapat dilihat pada tabel berikut :

Tabel 17. Evaluasi aturan asosiasi Apriori terhadap Tbl_Dataset_Evaluation1

Aturan Urut	Generalitas		Reliabilitas		
	Support	*Cover-age*	*Confidence*	*Added Value*	*Correlation*
1	0,6806	0,6806	1	0,0434	0,4157
2	0,7658	0,7664	0,9992	0,0426	0,4444
3	0,6896	0,6896	1	0,0434	0,4189
4	0,7868	0,7876	0,9989	0,0423	0,451
5	0,711	0,7124	0,998	0,0414	0,426
6	0,9112	0,9138	0,9971	0,0405	0,4871
7	0,6806	0,6896	0,9869	0,0731	0,4244
8	0,6806	0,6896	0,9869	0,0731	0,4244
9	0,6806	0,6896	0,9869	0,0757	0,4251
10	0,711	0,7212	0,9858	0,072	0,4348
Rata-rata	0,72978	0,73404	0,99397	0,05475	0,43518

Hasil evaluasi terhadap *evaluation dataset* yang disimpan pada tabel Tbl_Dataset_Evaluation2 dapat dilihat pada tabel berikut :

Tabel 18. Evaluasi aturan asosiasi Apriori terhadap Tbl_Dataset_Evaluation2

Aturan Urut	Generalitas		Reliabilitas		
	Support	*Coverage*	*Confidence*	*Added Value*	*Correlation*
1	0,7107	0,711	0,9995	0,0464	0,427
2	0,7851	0,7857	0,9992	0,0461	0,4515
3	0,7189	0,7195	0,9991	0,046	0,4298
4	0,8106	0,8121	0,9981	0,045	0,4592
5	0,7361	0,7382	0,9971	0,044	0,4352
6	0,9014	0,9048	0,9962	0,0431	0,4851
7	0,7107	0,7189	0,9885	0,0837	0,4377
8	0,711	0,7195	0,9881	0,0833	0,4377
9	0,7107	0,7195	0,9877	0,0863	0,4384
10	0,7361	0,7459	0,9868	0,082	0,4459
Rata-rata	0,75313	0,75751	0,99403	0,06059	0,44475

4.7.2. Evaluasi Aturan Asosiasi FP-growth

Hasil evaluasi terhadap *evaluation dataset* yang disimpan pada tabel Tbl_Dataset_Evaluation1 dapat dilihat pada tabel berikut :

Tabel 19. Evaluasi aturan asosiasi FP-growth terhadap Tbl_Dataset_Evaluation1

Aturan Urut	Generalitas		Reliabilitas		
	Support	*Coverage*	*Confidence*	*Added Value*	*Correlation*
1	0,2326	0,2338	0,9948	0,2072	0,2277
2	0,338	0,3418	0,9888	0,2012	0,2992
3	0,2326	0,24	0,9691	0,2163	0,2342

Tabel 19. Evaluasi aturan asosiasi FP-growth terhadap Tbl_Dataset_Evaluation1 (Lanjutan)

Aturan Urut	Generalitas		Reliabilitas		
	Support	*Coverage*	*Confidence*	*Added Value*	*Correlation*
4	0,338	0,346	0,9768	0,224	0,3076
5	0,2992	0,304	0,9842	0,1966	0,274
6	0,24	0,2506	0,9577	0,1701	0,2311
7	0,2338	0,2506	0,9329	0,1801	0,233
8	0,346	0,3594	0,9627	0,1751	0,3016
9	0,2326	0,2506	0,9281	0,2385	0,2473
10	0,3418	0,3594	0,951	0,1982	0,3073
Rata-rata	0,28346	0,29362	0,96461	0,20073	0,2663

Hasil evaluasi terhadap *evaluation dataset* yang disimpan pada tabel Tbl_Dataset_Evaluation2 dapat dilihat pada tabel berikut :

Tabel 20 Evaluasi aturan asosiasi FP-growth terhadap Tbl_Dataset_Evaluation2

Aturan Urut	Generalitas		Reliabilitas		
	Support	*Coverage*	*Confidence*	*Added Value*	*Correlation*
1	0,2462	0,2468	0,9975	0,1854	0,2325
2	0,3606	0,3635	0,992	0,1799	0,3067
3	0,2462	0,2486	0,9903	0,2094	0,2391
4	0,3606	0,3634	0,9922	0,2113	0,3151
5	0,3524	0,3599	0,9791	0,167	0,3006
6	0,2486	0,2562	0,9703	0,1582	0,2327
7	0,2468	0,2562	0,9633	0,1824	0,2379
8	0,3634	0,3735	0,9729	0,1608	0,3065
9	0,2462	0,2562	0,9609	0,2414	0,2523
10	0,3635	0,3735	0,9732	0,1923	0,3148
Rata-rata	0,30345	0,30978	0,97917	0,18881	0,27382

4.8. Analisa Hasil Evaluasi

Analisa hasil evaluasi dilakukan dengan menghitung tingkat kekuatan dari aturan-aturan asosiasi yang dihasilkan oleh masing-masing algoritma. Tingkat kekuatan dari aturan asosiasi ditentukan oleh nilai *support* yang mewakili aspek generalitas dan nilai *confidence* yang mewakili aspek reliabilitas.

Berikut adalah pengukuran tingkat kekuatan terhadap aturan-aturan asosiasi yang dihasilkan oleh algoritma Apriori :

Tabel 21. Tingkat kekuatan aturan asosiasi algoritma Apriori

Aturan Asosiasi	*Support*	*Confidence*	*Supp×Conf*
{Lem, Shrink, ArtKartonCover}⇒{SoftCover}	0,6806	1	0,6806
{Lem, ArtKartonCover}⇒{SoftCover}	0,7658	0,9992	0,765187
{Lem, Shrink}⇒{SoftCover}	0,6896	1	0,6896
{Lem}⇒{SoftCover}	0,7868	0,9989	0,785935
{Shrink, ArtKartonCover}⇒{SoftCover}	0,711	0,998	0,709578
{ArtKartonCover}⇒{SoftCover}	0,9112	0,9971	0,908558
{SoftCover, Lem, Shrink}⇒{ArtKartonCover}	0,6806	0,9869	0,671684
{Lem, Shrink}⇒{ArtKartonCover}	0,6806	0,9869	0,671684
{Lem, Shrink}⇒{SoftCover, ArtKartonCover}	0,6806	0,9869	0,671684
{SoftCover, Shrink}⇒{ArtKartonCover}	0,711	0,9858	0,700904
Tingkat kekuatan aturan asosiasi			0,725541

Sedangkan hasil pengukuran tingkat kekuatan terhadap aturan-aturan asosiasi yang dihasilkan oleh algoritma FP-growth adalah :

Tabel 22. Tingkat kekuatan aturan asosiasi algoritma FP-growth

Aturan Asosiasi	*Sup-port*	*Confidence*	*Supp× Conf*
{Shrink, UVDoff, NewsprintIsi}⇒{Lem}	0,2326	0,9948	0,23139
{Shrink, NewsprintIsi}⇒{Lem}	0,338	0,9888	0,334214
{Lem, UVDoff, NewsprintIsi}⇒{Shrink}	0,2326	0,9691	0,225413
{Lem, NewsprintIsi}⇒{Shrink}	0,338	0,9768	0,330158
{Shrink, UVDoff}⇒{Lem}	0,2992	0,9842	0,294473
{UVDoff, NewsprintIsi}⇒{Lem}	0,24	0,9577	0,229848
{UVDoff, NewsprintIsi}⇒{Shrink}	0,2338	0,9329	0,218112
{NewsprintIsi}⇒{Lem}	0,346	0,9627	0,333094
{UVDoff, NewsprintIsi}⇒{Lem, Shrink}	0,2326	0,9281	0,215876
{NewsprintIsi}⇒{Shrink}	0,3418	0,951	0,325052
Tingkat kekuatan aturan asosiasi			0,273763

Dari hasil analisa diketahui bahwa tingkat kekuatan aturan-aturan asosiasi yang dihasilkan oleh algoritma Apriori lebih besar dibandingkan dengan yang dihasilkan oleh algoritma FP-growth, penyebabnya adalah rendahnya nilai *support* yang menggambarkan terjadinya kombinasi *items* dalam aturan terhadap keseluruhan data transaksi terkini penjualan produk buku.

Tingkat akurasi dari algoritma Apriori terhadap FP-growth adalah sebesar:

$$\frac{\sum Support\ Apriori}{\sum Support\ FP-growth} \times 100 = \frac{7,2978}{2,8346} \times 100 = 257,4543\ \%$$

5. Kesimpulan Dan Saran

Beberapa kesimpulan yang dapat diambil dari hasil penelitian ini adalah :

a. Dari hasil observasi terhadap data-data sejumlah komponen transaksi penjualan produk buku yang terjadi percetakan PT. Gramedia hingga tanggal 1 Agustus 2010, telah ditemukan sebanyak 22 atribut-atribut yang dapat dimanfaatkan untuk melakukan penelitian *market basket analysis*.
b. Penelitian *market basket analysis* dengan menggunakan algoritma Apriori dan FP-growth terhadap atribut- atribut penjualan produk buku telah menghasilkan sejumlah aturan asosiasi yang berbeda antara algoritma yang satu dengan yang lainnya.
c. Analisa yang dilakukan terhadap tingkat kekuatan aturan-aturan asosiasi menunjukkan bahwa aturan-aturan asosiasi yang dihasilkan oleh algoritma Apriori memiliki tingkat kekuatan yang lebih tinggi dibandingkan dengan yang dihasilkan oleh algoritma FP-growth.

Beberapa saran yang dapat diberikan antara lain :

a. Perbaikan prosedur kerja yang berhubungan dengan proses pemasukan data-data transaksi ke dalam sistem basis data perlu mendapat perhatian yang khusus sehingga ketersediaan data pada saat diperlukan dapat lebih terjamin.
b. Dapat dilakukan penelitian lebih lanjut dengan memanfaatkan algoritma-algoritma *association data mining* lainnya, seperti: *Generalized Rule Induction, Algoritma Hash Based, Predictive Apriori,* dan *Tertius* untuk

pencarian aturan-aturan asosiasi berdasarkan atribut-atribut penjualan produk buku di Percetakan PT. Gramedia.

c. Perlu dilakukan penelitian untuk mengkaji kelemahan pada algoritma FP-growth yang mengakibatkan lemahnya aturan-aturan asosiasi yang dihasilkan melalui penerapan *market basket analysis* terhadap atribut penjualan produk buku di Percetakan PT. Gramedia.
d. Pemanfaatan teknologi *data warehouse* pada basis data SQL Server sehingga proses penyiapan data yang merupakan syarat awal untuk melakukan *data mining* dapat dilakukan dengan lebih cepat dan efisien. Disamping itu, kualitas dan integritas data yang tersimpan dalam *data warehouse* lebih terjamin.

Daftar Pustaka

[1] Kusrini dan Emha Taufiq Luthfi, "ALGORITMA *DATA MINING*", PENERBIT ANDI, 2009.
[2] Budi Santosa, "*Data Mining* : Teknik Pemanfaatan Data untuk Keperluan Bisnis", GRAHA ILMU, 2007
[3] Jiawei Han dan Micheline Kamber, "*Data Mining: Concepts and Techniques* : *Chapter 6. Mining Association Rules in Large Databases*", Simon Fraser University, 2000.
[4] Gregorius Satia Budhi, Andreas Handojo dan Christine Oktavina Wirawan, 2009, "Algoritma *Generalized Sequential Pattern* Untuk Menggali Data Sekuensial Sirkulasi Buku Pada Perpustakaan Uk Petra", Seminar Nasional Aplikasi Teknologi Informasi, 2009.
[5] Kusrini, "Penerapan Algortima Apriori Pada *Data Mining* Untuk Mengelompokkan Barang Berdasarkan Kecenderungan Kemunculan Bersama Dalam Satu Transaksi", E-learning Center STMIK AMIKOM Yogyakarta, 2007
[6] CRISP-DM, "*Cross Industry Standard Process for Data Mining*", *http:// www.crisp-dm.org/Process/index.htm*, (Diakses 20 Agustus 2010).
[7] Fayyad, U., Shapiro, G. Piatetsky, dan Smyth, N Padhraic, "*From Data Mining to Knowledge Discovery in Databases*", Al Magazine, Hal. 37-54, 1996.

FP-tree and SVM for Malicious Web Campaign Detection

Michał Kruczkowski[1,2],
Ewa Niewiadomska-Szynkiewicz[1,3(✉)], and Adam Kozakiewicz[1,3]

[1] Research and Academic Computer Network (NASK), Warsaw, Poland
{michal.kruczkowski,adam.kozakiewicz}@nask.pl
[2] Institute of Computer Science, Polish Academy of Science, Warsaw, Poland
[3] Institute of Control and Computation Engineering, Warsaw University of Technology, Warsaw, Poland
ens@ia.pw.edu.pl

Abstract. The classification of the massive amount of malicious software variants into families is a challenging problem faced by the network community. In this paper (The work was supported by the EU FP7 grant No. 608533 (NECOMA) and "Information technologies: Research and their interdisciplinary applications", POKL.04.01.01-00-051/10-00.) we introduce a hybrid technique combining a frequent pattern mining and a classification technique to detect malicious campaigns. A novel approach to prepare malicious datasets containing URLs for training the supervised learning classification method is provided. We have investigated the performance of our system employing frequent pattern tree and Support Vector Machine on the real database consisting of malicious data taken from numerous devices located in many organizations and serviced by CERT Polska. The results of extensive experiments show the effectiveness and efficiency of our approach in detecting malicious web campaigns.

Keywords: Malware campaign · URL · FP-tree · FP-growth · SVM

1 Introduction to Campaign Identification

Recently, numerous attacks have threatened the operation of the Internet [3,8,10]. One of the major threats on the Internet is malicious software, often referred to as malware. Malware is a software designed to perform unwanted actions on computer systems such as gather sensitive information, or disrupt and even damage computer systems. In case of finding a server vulnerability (similar to the disease [7]) affecting multiple web pages, the attacker can automate the exploitation and launch an infection campaign in order to maximize the number of potential victims. Hence, the campaign determines a group of incidents that have the same objective and employ the same dissemination strategy [4]. Detecting malicious web campaigns has become a major challenge in Internet today [3,9,12]. Therefore, strategies and mechanisms for effective and fast detection of malware are crucial components of most network security systems.

© Springer International Publishing Switzerland 2015
N.T. Nguyen et al. (Eds.): ACIIDS 2015, Part II, LNAI 9012, pp. 193–201, 2015.
DOI: 10.1007/978-3-319-15705-4_19

In this paper we propose and investigate the use of frequent pattern mining [1,5] and supervised learning classification [6,13] for malicious campaign identification. The detection process is based on the analysis of URLs (Uniform Resource Locators). Each URL expressed by (1)

$$scheme : //domain : port/path?query_string\#fragment_id \tag{1}$$

consists of the following components, i.e., scheme, domain name or IP address, port, path, query string and a fragment identifier (some schemes also allow a username and password in front of the domain). The query string consists of pairs of keys and values (also called attributes). By malicious URL we denote such URL that is created with malicious purposes. It can serve malicious goals, for example download any type of malware, which can be contained in phishing or spam messages, etc. It is obvious that the ultimate goal of obfuscation is to make each URL unique. Due to the fact that malicious URLs are generated by tools which employ the same obfuscation mechanisms for a given campaign we assume that usually intruders keep some parts of the URL static, while other parts are changed systematically and in an automated fashion. Moreover, since the duration of a given campaign is relatively short, we assume that the obfuscation mechanisms employed for such campaign do not change significantly during this time. We have designed and developed the system for malware campaigns identification. Our system utilizes FP-tree data structure of tokenized malicious URLs to form a training dataset for learning Support Vector Machine classifier. Finally, the trained SVM is used for on-line classification of new malware URLs into related or unrelated with malicious campaigns.

2 FP-Tree Structure and FP-Growth Algorithm

Frequent pattern mining is a widely used technique for discovering interesting relations between data items in large databases. It is employed today in many application areas including malware detection, Web usage mining, etc. In our research we use the frequent pattern tree structure (FP-tree) – a prefix structure for storing quantitative information about frequent patterns in a database – and the FP-growth algorithm for frequent pattern discovery using a divide-and-conquer strategy, both developed by J. Han et al. [5]. The FP-growth algorithm operates in two steps: 1) the FP-tree compact structure is constructed, 2) frequent patterns are extracted from the FP-tree. In the first step the occurrence of patterns in the input transaction database is counted. Next, infrequent patterns are discarded, frequent patterns are sorted by descending order of their frequency in the database, and the FP-tree structure is built. Common, usually most frequent patterns are shared. Therefore, FP-tree provides high compression close to tree root and can be processed quickly. Recursive growth is applied to extract the frequent patterns. FP-growth starts from the bottom of the tree structure (longest branches), by finding all patterns matching given condition. New tree is created, etc. Recursive growth ends when no patterns meet the condition, and processing continues on the remaining main branches of the original FP-tree.

3 Support Vector Machine Classifier

The Support Vector Machine (SVM) [13] is a supervised learning classification method widely used in data mining research. The concept of SVM is to classify each data sample into one of two categories: positive class denoted by "+1" and negative class denoted by "-1". Thus, the goal is to determine a decision boundary, which divides data into two sets (one for each class), a plane for $n \leq 3$ or hyperplane for $n > 3$, where n denotes the size of a dataset. Next, all the measurements on one side of this boundary are classified as belonging to "+1" class and all those on the other side as belonging to "-1" class. The problem is that many such hyperplanes can be determined, and the best one has to be selected. Hence, SVM tries to learn the decision boundary which gives the best generalization. A good separation is achieved by the hyperplane that has the largest distance to the nearest data sample of any class – a wider margin implies the lower generalization error of the classifier. To select the maximum margin hyperplane an optimization problem is formulated and solved for a given training dataset. The original SVM model – linear classifier – was developed by V.N. Vapnik, [13]. The hyperplane is calculated as a solution of a quadratic optimization problem. However, in many practical applications the datasets are not linearly separable in the data space. To determine the hyperplane the original space is mapped into a much higher-dimensional space using nonlinear kernel functions. The kernel function $K(x_i, x_j)$ defines the similarity between a given pair of objects. A large value of $K(x_i, x_j)$ indicates that x_i and x_j are similar and a small value indicates that they are dissimilar. Various kernel functions and described in literature [6].

Finally, the trained SVM can be used to classify the samples from a new dataset based only on the knowledge about their attributes.

4 FP-SVM System

The architecture of our system for malicious campaigns identification (FP-SVM) is presented in Fig. 1. It consists of three main modules: a database collecting malicious data, a frequent pattern mining module that implements the FP-growth algorithm for frequent patterns discovery and a data classification module that uses the SVM method to classify malicious datasets containing URLs as related or not to a campaign. To produce the SVM classifier a training dataset collecting malicious data (precisely URLs) that have already been classified into campaigns is required. Unfortunately, access to databases collecting malicious data divided into campaigns (if they exist) is restricted. Therefore, the only sensible solution is to define some patterns of URLs related with campaigns based on analysis of existing databases collecting malicious data, and generate a set of samples containing these patterns. It is assumed that all these samples are related with campaigns, and can be used to train the SVM classifier. In our FP-SVM system the FP-growth algorithm is applied to produce the training dataset. A fixed number N of URLs are selected from the database, and the

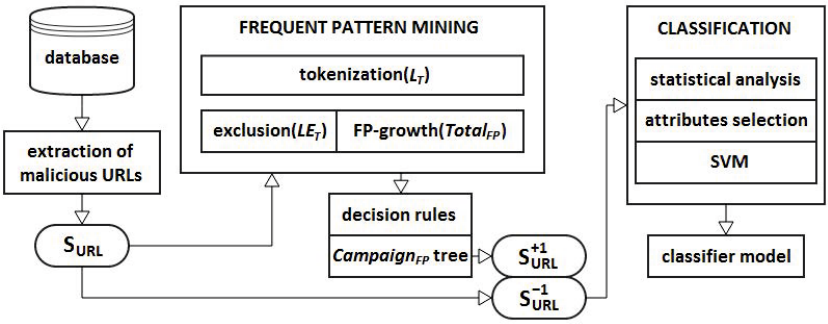


Fig. 1. The architecture of the FP-SVM system

input dataset $S_{URL} = \{URL_1, URL_2, \ldots, URL_N\}$ for frequent pattern mining is created. Each sample from S_{URL} is tokenized. The simple heuristic rules to break up a given URL (stream of text) into shorter strings described in literature [2,4] are adopted. Each sample is cut by a specific characters that are typical for URLs (1), i.e., "/", ".", "?", "#", etc. As a final result of this operation we obtain a set of tokens L_T. This set of tokens becomes input for further processing such as parsing and typical subsequence mining. The goal is to reduce the size of the dataset consisting of extracted tokens and finally speed up the FP-tree generation. The typical URL's attributes which do not carry valuable information, such as the schemes: "http", "https", domain name parts "www", "org", "com", "waw", etc. and extensions: "exe", "php", "html", "xhtml" are excluded from the set L_T. Once the final set of tokens LE_T is built, the FP-growth algorithm is employed to discover frequent tokens and the FP-tree structure $Total_{FP}$ storing quantitative information about frequent tokens from LE_T is constructed. In this tree each node (besides root) represents an extracted token that is shared by all subtrees consisting of itself and all the nodes beneath it. Each path in the tree shows a set of tokens that co-occur in URLs. Thus two URLs that contain several identical frequent tokens and differ in several infrequent tokens share a common path. The root is the node that has no superior and separates all disjoint subtrees. The $Total_{FP}$ tree structure is analysed. Simple decision rules are used for data processing. These rules define the characteristics of each URL that is suspected to belong to any campaign. The final FP-tree structure $Campaign_{FP}$ formed by URLs with these characteristics is created. $S^{+1}_{URL} and S^{-1}_{URL}$. Both these datasets form a training set of samples that is used to produce the SVM classifier model. The relevant attributes used for URL classification are selected. The commonly used attributes assigned to URLs are: date, time, address (IP, ASN), length of address, domain name, length of domain name, number of subdomains, path name, length of path name, number of subpaths, length of query, number of queries, country code, confidence of code. The selection of adequate attributes is a key feature that guarantees the effective and efficient classification. The

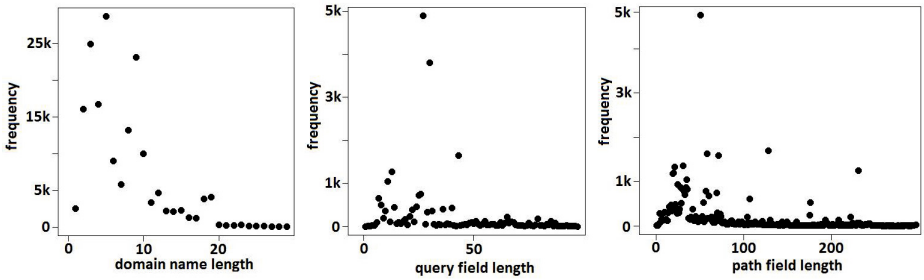


Fig. 2. Distribution of URLs; various attributes

preliminary statistical analysis on the malicious dataset is often performed to select the set of these attributes. The results of such analysis performed on the example set of malicious data is presented in Fig. 2. This figure shows the frequency of occurrence of respectively, lengths of domain name, query and path. It can be seen that in case of two attributes, i.e., length of domain name and length of path the distribution is similar After preliminary analysis of the available data the following attributes have been selected in the FP-SVM system: date, time, address (IP, ASN), length of address, domain name, number of subdomains, path name, number of queries, country code, confidence of code. Next step of the SVM algorithm is to learn the decision boundary. Four variants of the SVM classifier with linear and nonlinear kernels are implemented in our FP-SVM system. The following nonlinear kernels are provided: polynomial function: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d$, radial basis function: $K(x_i, x_j) = exp(\gamma \parallel x_i x_j \parallel^2)$, sigmoid function: $K(x_i, x_j) = tanh(\gamma x_i^T x_j + r)$, where $\gamma > 0$, r, and d denote kernel parameters. Finally, the trained SVM classifier can be employed by malware detection systems to classify malicious URLs taken directly from the Internet as related or unrelated to known malicious campaigns. Note that suspicious, unverified URLs can also be analyzed – if they are found likely to be part of a campaign, their malicious status is confirmed. It is obvious that the classification model has to be continuously updated. The data about new Internet security threats should be used in training process.

5 Performance Evaluation

5.1 The N6 Platform

The FP-SVM system was used to classify heterogeneous data from a real malware URL database of the n6 platform to identify malicious campaigns. The n6 platform [11] developed at NASK (Research and Academic Computer Network) is used to monitor computer networks, store and analyse data about incidents, threats, etc. The n6 database collects data taken from various sources,

including security organizations, software providers, independent experts, etc., and monitoring systems serviced by CERT Polska. The datasets contain URLs of malicious websites, addresses of infected machines, open DNS resolvers, etc. Most of the data is updated daily. Information about malicious sources is provided by the platform as URL's, domain, IP addresses, names of malware, etc. We have performed a preliminary analysis of 27 700 560 URLs collected in year 2013 and stored in the n6 database. Figure 3 depicts the amount of malicious URLs observed per day (in a selected month) and per month. As we can see, the average number of observed URLs is about 80 000 daily or 2 300 000 monthly.

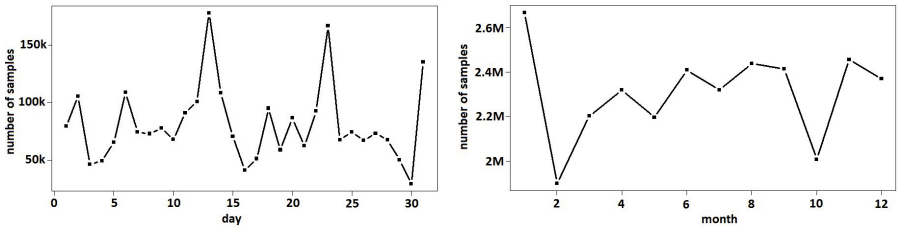


Fig. 3. Malicious URLs detected during a day and during a month

5.2 Case Study Results

The aim of the experiments was to validate the FP-SVM system on the n6 dataset. First, N malicious URLs were selected from the n6 malware database. They formed the S_{URL} training set. Next, frequent pattern analysis was applied and the $Total_{FP}$ tree with nodes representing tokens extracted from URLs from the S_{URL} dataset was constructed. The following rule was used to extract the subtree $Campaign_{FP}$ from the original $Total_{FP}$ tree. We assumed that all URLs containing m common tokens in a sequence were suspected to be related with the same campaign. Hence, short branches with less than m nodes were excluded from the $Total_{FP}$ tree. The $Total_{FP}$ tree and the approach to the $Campaign_{FP}$ tree generation are presented in Fig. 4. Table 1 presents the results of the application of the FP-growth algorithm and our decision rule with $m = 4$ to the S_{URL} dataset. It contains the number of detected malware campaigns and average number of URL's tokens related with one campaign.

$S^{+1}_{URL} and S^{-1}_{URL}$ produce the SVM classifier model. Various sizes of S_{URL} were considered, i.e., $N = n \cdot 100000$, $n = 1, 2, 3, 4, 5$. The decision borders for classification were calculated and validated for each size of S_{URL}. Finally, the trained SVM classifier was applied to malware campaign detection. It was used to classify the dataset consisting of 80 000 malicious URLs (unrelated with URLs from training dataset) into two categories: +1 – URLs related with any campaign, -1 – URLs unrelated with any campaign. Then, the quality of the classification was assessed. The following commonly used criteria were considered: classification accuracy (CA) – ratio of number of correctly identified URLs to the size of the

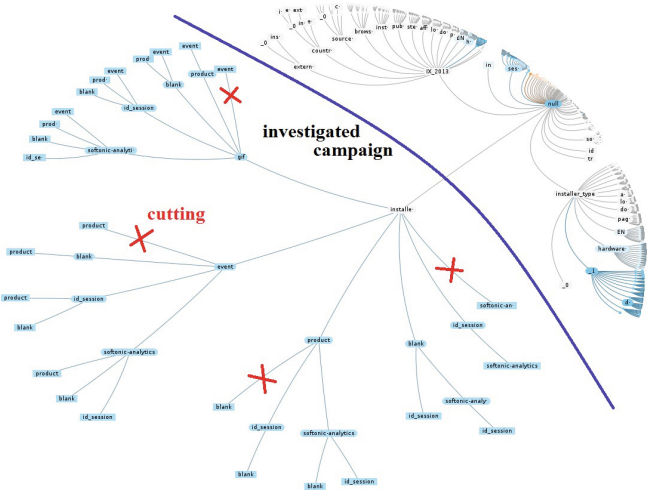


Fig. 4. Frequent patterns tree and a campaign extraction

Table 1. Identification of malicious campaigns; the S_{URL} dataset

Number of URLs (number of IP)	500 000 (108 745)
Number of unique URLs (number of IP)	69 906 (5 321)
Number of detected campaigns	72
Average number of related attributes	84 per campaign

dataset, sensitivity – the proportion of positives that are correctly identified as such, specificity – the proportion of negatives that are correctly identified as such, accuracy of a test. The accuracy of the test is measured by AUC (Area Under ROC Curve – the measure that shows how well the test separates the URLs being tested) and another popular measure – F-measure. The value of precision shows how close the separated URLs are to each other. The values of all mentioned criteria obtained for various variants of SVM classifier (providing linear and nonlinear kernel functions) and a training dataset S_{URL} consisting of 250 000 malware URLs are collected in Table 2. Figure 5 shows the accuracy of the classification for various sizes of the training dataset. In general, the

Table 2. Evaluation of SVM classification; $N = 250000$ URLs

	Radial	Sigmoid	Polynomial	Linear
CA	0.7035	0.5571	0.5000	0.3495
Sensitivity	0.8750	0.7083	0.4375	0.6042
Specificity	0.8202	0.6685	0.8258	0.6011
AUC	0.9250	0.8653	0.8367	0.7823
F-measure	0.6885	0.4823	0.4200	0.3919
Precision	0.5676	0.3656	0.4038	0.2900

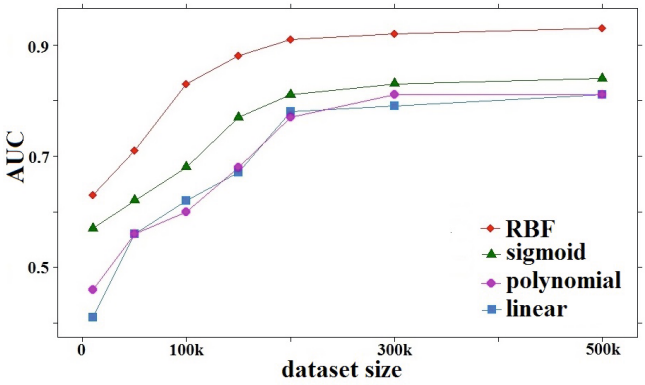


Fig. 5. Area Under ROC Curve values for various sizes of the training dataset S_{URL}

results presented in Fig. 5 and Table 2 confirm that the accuracy of classification strongly depends on the size and quality of a training dataset. Moreover, it is very important to choose the adequate kernel function. The achieved classification accuracy ranged from about 35% to 70%, accuracy of the test from 78% to 93%, sensitivity from 44% to 88%, and specificity from 60% to 82%. Explicitly, the worst results were obtained for the SVM variant implementing a linear kernel function. The best results – the best values of all criteria – were obtained when employing the radial basis kernel function. It is worth to mention that due to our assumption that the same obfuscation mechanisms is used to the malicious URLs generation, and usually intruders keep some parts of the URL static we are able to identify the URLs related to a given campaign. In our approach we use the most frequent tokens (the static parts of malicious URLs) in the classification process.

6 Summary and Conclusion

Data mining methods hold a great potential toward detection of malicious software. To identify malicious web campaigns we have adopted a technique incorporating a combination of frequent pattern mining and supervised learning methods. In general, the presented results of experiments confirm that our classification system employing the FP-growth algorithm and the SVM method gives satisfactory results. This technique can be successfully used to analyse a huge amount of dynamic, heterogenous, unstructured and imbalanced network data. However, accuracy of this approach depends heavily on the selection and calibration of parameters, especially the choice of a kernel function and of the rules for selection of URLs for the training dataset. As a final conclusion we can say that it can be expected that our FP-SVM system can be successfully implemented in intrusion detection systems as a malicious campaign sensor. In our future work we plan to extend our system with the tool mapping a detected malicious software to a given campaign.

References

1. Argawal, C., Li, Y., Wang, J.: Frequent pattern mining with uncertain data. In: Proc. of 15th Inter. Conf. on Knowledge Discovery and Data Mining (ACM SIGKDD), pp. 29–38 (2009)
2. Calais, P., Pires, D., Neto, D., Meira, W., Hoepers, C., Steding-Jessen, K.: A campaign-based characterization of spamming strategies. In: CEAS 2008, pp. 1–6 (2008)
3. Gandotra, E.: Malware analysis and classification: A survey. Journal of Information Security **5**, 56–64 (2014)
4. Gao, H., Hu, J., Wilson, C., Li, Z., Chen, Y., Zhao, B.: Detecting and characterizing social spam campaigns. In: Proc. of the 10th ACM SIGCOMM Conference on Internet Measurement, pp. 35–47 (2010)
5. Han, Y., Pei, Y., Yin, Y.: Mining frequent patterns without candidate generation. In: Proc. of SIGMOD, pp. 1–12 (2000)
6. Jebara, T.: Multi-task feature and kernel selection for svms. In: Proc. of Inter. Conf. on Machine Learning, pp. 55–63 (2004)
7. Radu, V.: Application. In: Radu, V. (ed.) Stochastic Modeling of Thermal Fatigue Crack Growth. ACM, vol. 1, pp. 63–70. Springer, Heidelberg (2015)
8. Kozakiewicz, A., Felkner, A., Kijewski, P., Kruk, T.: Application of bioinformatics methods to recognition of network threats. JTIT, pp. 23–27 (2007)
9. Kruczkowski, M., Niewiadomska-Szynkiewicz, E.: Support vector machine for malware analysis and classification. In: Proc. of IEEE/WIC/ACM Inter. Conf. on Web Intelligence, pp. 1–6 (2014)
10. Lasota, K., Kozakiewicz, A.: Analysis of the similarities in malicious dns domain names. In: Lee, C., Seigneur, J.-M., Park, J.J., Wagner, R.R. (eds.) STA 2011 Workshops. CCIS, vol. 187, pp. 1–6. Springer, Heidelberg (2011)
11. NASK: n6 platform (2014). http://www.cert.pl/news/tag/n6
12. de Oliveira, I.L., Grégio, A.R.A., Cansian, A.M.: A malware detection system inspired on the human immune system. In: Murgante, B., Gervasi, O., Misra, S., Nedjah, N., Rocha, A.M.A.C., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2012, Part IV. LNCS, vol. 7336, pp. 286–301. Springer, Heidelberg (2012)
13. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)

Int. J. of Advanced Networking and Applications
Volume:01, Issue:06, Pages: 400-404 (2010)

Implementation of Web Usage Mining Using APRIORI and FP Growth Algorithms

B.Santhosh Kumar
Department of Computer Science, C.S.I. College of Engineering, Ketti- 643 215. The Nilgiris.
Email: b.santhoshkumar@csice.edu.in

K.V.Rukmani
Department of Computer Science, C.S.I. College of Engineering, Ketti- 643 215. The Nilgiris.
Email: rukmani_arun@csice.edu.in

ABSTRACT

Web Usage Mining is the application of data mining techniques to discover interesting usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. Usage data captures the identity or origin of Web users along with their browsing behavior at a Web site. Web usage mining itself can be classified further depending on the kind of usage data considered. They are web server data, application server data and application level data. Web server data correspond to the user logs that are collected at Web server. Some of the typical data collected at a Web server include IP addresses, page references, and access time of the users and is the main input to the present Research. This Research work concentrates on web usage mining and in particular focuses on discovering the web usage patterns of websites from the server log files. The comparison of memory usage and time usage is compared using Apriori algorithm and Frequent Pattern Growth algorithm.

Keywords : Apriori, Data cleaning, FP Growth, FP-tree, Web Usage Mining,

Date of Submission: 16, February 2010 Date of Acceptance: 08, A pril 2010

1. INTRODUCTION

The Web is a huge, explosive, diverse, dynamic and mostly unstructured data repository, which supplies incredible amount of information, and also raises the complexity of how to deal with the information from the different perspectives of view, users, web service providers, business analysts. The users want to have the effective search tools to find relevant information easily and precisely. The Web service providers want to find the way to predict the users' behaviors and personalize information to reduce the traffic load and design the Web site suited for the different group of users. The business analysts want to have tools to learn the user/consumers' needs. All of them are expecting tools or techniques to help them satisfy their demands and/or solve the problems encountered on the Web. Therefore, Web mining becomes a popular active area and is taken as the research topic for this investigation.

Web Usage Mining [4], [5] is the application of data mining techniques to discover interesting usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. Usage data captures the identity or origin of Web users along with their browsing behavior at a Web site. Web usage mining itself can be classified further depending on the kind of usage data considered. They are web server data, application server data and application level data. Web server data correspond to the user logs that are collected at Web server. Some of the typical data collected at a Web server include IP addresses, page references, and access time of the users and is the main input to the present Research. This Research work concentrates on web usage mining and in particular focuses on discovering the web usage patterns of websites from the server log files.

1.1 APRIORI ALGORITHM

The present Research work is designed to operate on log files. The algorithm [7] attempts to find subsets which are common to at least a minimum number C (the cutoff, or confidence threshold) of the item sets. The system operates in the following three modules.

- Preprocessing module
- Apriori or FP Growth Algorithm Module
- Association Rule Generation
- Results

The preprocessing module converts the log file, which normally is in ASCII format, into a database like format, which can be processed by the Apriori algorithm.

The second module is performed in two steps.

- Frequent Item set generation
- Rules derivation

1.2 FP GROWTH ALGORITHM:

The FP Growth algorithm operates in the following four modules.

- Preprocessing module
- FP Tree an FP Growth Module
- Association Rule Generation
- Results

The preprocessing modules convert the log file, which normally is in ASCII format, into a database like format, which can be processed by the FP Growth algorithm.

The 2nd module is performed in two steps.

- FP Tree generation
- Applying FP Growth to generate association rules

FP tree is a compact data structure that stores important, crucial and quantitative information about frequent patterns.

The main components of FP tree are:

- It consists of one root labeled as "root", a set of item prefix sub-trees as the children of the root, and a frequent-item header table.
- Each node in the item prefix sub-tree consists of three fields: item-name, count, and node-link, where item-name registers which item this node represents, count registers the number of transactions represented by the portion of the path reaching this node, and node-link links to the next node in the FP-tree carrying the same item-name, or null if there is none.
- Each entry in the frequent-item header table consists of two fields, (1) item-name and (2) head of node-link, which points to the first node in the FP-tree carrying the item-name.

Second, an FP-tree-based pattern-fragment growth mining method is developed, which starts from a frequent length-1 pattern (as an initial suffix pattern), examines only its conditional-pattern base (a "sub-database" which consists of the set of frequent items co-occurring with the suffix pattern), constructs its (conditional) FP-tree, and performs mining recursively with such a tree. The pattern growth is achieved via concatenation of the suffix pattern with the new ones generated from a conditional FP-tree. Since the frequent item set in any transaction is always encoded in the corresponding path of the frequent-pattern trees, pattern growth ensures the completeness of the result.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The Research work was initiated through a system study and analysis phase, where significant study was conducted to understand the existing system. Using Apriori algorithm for weblog mining is a novel technique. The explosive growth of the World Wide Web (WWW) in recent years has turned the web into the largest source of available online data.

- Situations like several unrelated topics in a single web page may lead to confusion and make harder to reach the information that the visitors are looking for.
- The design of the whole site (interface, content, structure, usability, etc.) is one of the most important aspects for any institution that wants to survive in the cyberspace.
- Understand the way user browses the site and find out which is the most frequent used link and pattern of using the features available in the site.

All these information is available online but are hidden for the users. Presently, there is no powerful that can analyze this hidden information and this Research work uses web usage mining (WUM) Apriori based approach for analyzing the visitor browsing behavior.

2.2 LIMITATIONS OF APRIORI ALGORITHM

Apriori algorithm, in spite of being simple, has some limitation. They are,

- It is costly to handle a huge number of candidate sets. For example, if there are 10^4 frequent 1-item sets, the Apriori algorithm will need to generate more than 10^7 length-2 candidates and accumulate and test their occurrence frequencies. Moreover, to discover a frequent pattern of size 100, such as $\{a_1, \ldots, a_{100}\}$, it must generate $2^{100} - 2 \sim 10^{30}$ candidates in total. This is the inherent cost of candidate generation, no matter what implementation technique is applied.
- It is tedious to repeatedly scan the database and check a large set of candidates by pattern matching, which is especially true for mining long patterns.

In order to overcome the drawback inherited in Apriori, an efficient FP-tree based mining method, FP-growth, which contains two phases, where the first phase constructs an FP tree, and the second phase recursively Researches the FP tree and outputs all frequent patterns.

2.3 PROPOSED SYSTEM

The main goal of the proposed system is to identify usage pattern from web log files of a website. Apriori [7] and FP Growth Algorithm is used for this purpose. The main goal of the proposed system is to identify usage pattern from web log files of a website. For this purpose, the usage of apriori and FP Growth algorithms are proposed. Both are influential algorithms for mining frequent item sets for boolean association rules [1], [9]. In computer science and data mining, Apriori is a classic algorithm for learning association rules [10]. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or

details of a website frequentation).

Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k - 1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The key concepts in this algorithm are

- Frequent Item sets: The sets of item which has minimum support (denoted by Lifor ith-Item set).
- Apriori Property: Any subset of frequent item set must be frequent.
- Join Operation: To find L_k, a set of candidate k-item sets is generated by joining L_{k-1} with itself.

The advantages of using apriori algorithm are

- Uses large item set property.
- Easily parallelized
- Easy to implement

2.4 ADVANTAGES OF FP GROWTH ALGORITHM

The major advantages of FP-Growth algorithm is,
- Uses compact data structure
- Eliminates repeated database scan

FP-growth is an order of magnitude faster than other association mining algorithms and is also faster than tree-Researching. The algorithm reduces the total number of candidate item sets by producing a compressed version of the database in terms of an FP-tree. The FP-tree stores relevant information and allows for the efficient discovery of frequent item sets.

The algorithm consists of two steps:
- Compress a large database into a compact, *Frequent-Pattern tree* (FP-tree) structure
  – highly condensed, but complete for frequent pattern mining and avoid costly database scans
- Develop an efficient, FP-tree-based frequent pattern mining method (FP-growth)
  – A divide-and-conquer methodology: decompose mining tasks into smaller ones and avoid candidate generation: sub-database test only

2.5 ADVANTAGE OF FP-TREE STRUCTURE

The most significant advantage of the FP-tree is that the algorithm scans the tree only twice. Apart from this major advantage, the others include,

- Completeness:
  The FP-tree contains all the information related to mining frequent patterns (given the min_support threshold)

- Compactness:
  – The size of the tree is bounded by the occurrences of frequent items
  – The height of the tree is bounded by the maximum number of items in a transaction

Three major steps performed are starting the processing from the end of list L:

- Construct conditional pattern base for each item in the header table
- Construct conditional FP-tree from each conditional pattern base
- Recursively mine conditional FP-trees and grow frequent patterns obtained so far. If the conditional FP-tree contains a single path, simply enumerate all the patterns

3. SYSTEM DESIGN AND DEVELOPMENT

3.1 INPUT DESIGN

The input design is the process of converting user-oriented input to a computer-based format. The goal of the input design is to make the data entry easier, logical and error free. In the present Research work, the input is taken from the web log file. The web log file has the extension .log and contains ASCII characters. A log file is a text file in which every page request made to the web server is recorded. For each request the corresponding log file contains the following information:

IP address of the computer making the request;
- User ID, (this field is not used in most cases);
- date and time of the request;
- a status field indicating if the request was successful;
- size of the file transferred;
- Referring URL, that is, the URL of the page which contains the link that generated the request; name and version of the browser being used.

This information can be used to reconstruct the user navigation sessions within the site from which the log data originates. In an ideal scenario, each user is allocated an unique IP address whenever an access is made to a given web site. Moreover, it is expected that a user visits the site more than once, each time possibly with a different goal in mind. Once the domain-dependent data transformation phase is completed, the resulting transaction data must be formatted to conform to the data model of the appropriate data mining task. For instance,

Volume:01, Issue:06, Pages: 400-404 (2010)

the format of the data for the [2],[12] discovery task may be different than the format necessary for mining sequential patterns. Finally, a query mechanism will allow the user (analyst) to provide more control over the discovery process by specifying various constraints.

2.5 OUTPUT DESIGN

The main objective of any system is the generation of reports. It has various uses. Some of them are,

- For the users, reports provide source of information required.
- They provide permanent hard copy of the results of transactions.

Careful consideration has been given in the designing of the reports as it helps in decision-making process. In the present work, the performance of the system is judged using two metrics. The first one is the amount of memory used and the second one is the time taken for the algorithm to create the association rules [3], [9]. It was found that the FP Growth algorithm is fast and uses only very small amount of memory when compared with Apriori algorithm.

4. CODING STANDARDS

4.1 APRIORI ALGORITHM

In computer science and data mining, Apriori is a classic algorithm for learning association rules [6],[11]. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k - 1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The key concepts in this algorithm are

- Frequent Item sets: The sets of item which has minimum support (denoted by Lifor ith-Item set).
- Apriori Property: Any subset of frequent item set must be frequent.
- Join Operation: To find L_k, a set of candidate k-item sets is generated by joining L_{k-1} with itself.

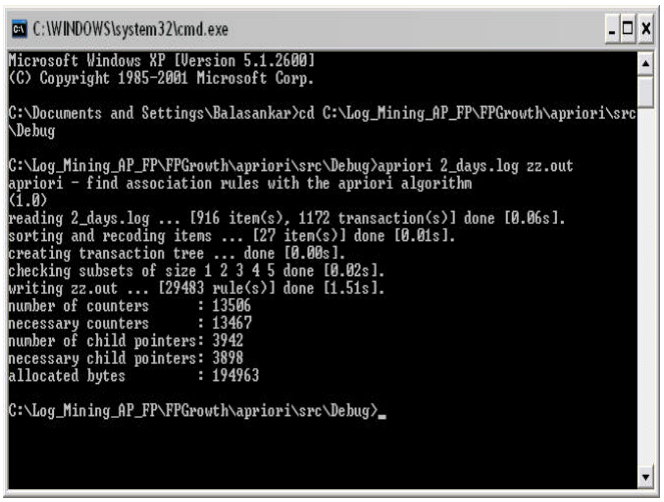
The advantages of using Apriori algorithm are

- Uses large item set property.
- Easily parallelized
- Easy to implement

The Apriori algorithm is an efficient algorithm for finding all frequent item sets. It implements level-wise search using frequent item property and can be additionally optimised. The Apriori algorithm used is given below.

- L_k: Set of frequent item sets of size k (with min support)
- C_k: Set of candidate item set of size k (potentially frequent item sets)

L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) **do**
  C_{k+1} = candidates generated from L_k;
  for each transaction t in database do
    increment the count of all candidates in C_{k+1} that are contained in t
  L_{k+1} = candidates in C_{k+1} with min_support
return $\cup_k L_k$;



4.2 FP GROWTH ALGORITHM

Definition of FP-Tree

FP-tree is a frequent pattern tree consists of one root labeled as "null". It has a set of *item prefix sub trees* as the children of the root, and a *frequent-item header table*. Each node in the *item prefix sub trees* has three fields:

- item-name to register which item this node represents,
- count, the number of transactions represented by the portion of the path reaching this node, and
- Node-link that links to the next node in the FP-tree carrying the same item-name, or null if there is none.

Each entry in the *frequent-item header table* has two fields,

- item-name, and
- head of node-link that point to the first node in the FP-tree carrying the item-name.

5. CONCLUSION

Web usage mining is the application of data mining techniques to discover usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. Web usage mining consists of three phases, namely preprocessing, pattern discovery, and pattern analysis. This Research work implements each of these phases. One of the algorithms which is very simple to use and easy to implement is the Apriori algorithm. This algorithm is used in the present Research work to generate association rules that associates the usage pattern of the clients for a particular website. The output of the system was in terms of memory usage and speed of producing association rules.

The main drawback of Apriori algorithm is that the candidate set generation is costly, especially if a large number of patterns and/or long patterns exist. The main drawback of FP-growth algorithm is the explosive quantity of lacks a good candidate generation method. Future research can combine FP-Tree with Apriori candidate generation method to solve the disadvantages of both apriori and FP-growth. In future the algorithm can be extended to web content mining, web structure mining, etc. The work can also be extended to extract information from image files.

6. ACKNOWLEDGEMENTS

"Implementation of web usage mining using Apriori and FP growth algorithms" research is supported by the management of ***C.S.I College of Engineering, Ketti***, The Nilgiris, whose support we are pleased to acknowledge. We are also grateful to our colleagues in CSICE for useful discussions, and thank to our beloved CSICE students.

REFERENCES

[1] Kotsiantis S, Kanellopoulos D., *Association Rules Mining: A Recent Overview, GESTS International Transactions on Computer Science and Engineering*, Vol.32 (1), 2006, pp. 71-82

[2] Agrawal R, Srikant R., "*Fast Algorithms for Mining Association Rules*", VLDB. Sep 12-15 1994, Chile, 487-99, pdf, ISBN 1-55860-153-8.

[3] Mannila H, Toivonen H, Verkamo A I., "*Efficient algorithms for discovering association rules." AAAI Workshop on Knowledge Discovery in Databases* (SIGKDD). July 1994, Seattle, 181-92.

[4] Tan, P. N., M. Steinbach, V. Kumar, "*Introduction to Data Mining"*, Addison-Wesley, 2005, 769pp.

[5] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques With Java Implementation*, 2nd ed. San Mateo, CA: Morgan Kaufmann, 2005.

[6] P. Becuzzi, M. Coppola, and M. Vanneschi, "*Mining of Association Rules in Very Large Databases: A Structured Parallel Approach,"* Proc. Europar-99, vol. 1685, pp. 1441-1450, Aug. 1999.

[7] R. Jin and G. Agrawal, *"An Efficient Implementation of Apriori Association Mining on Cluster of SMPs,"* Proc. Workshop High Performance Data Mining (IPDPS 2001), Apr. 2001.

[8] J. Han and M. Kamber, *"Data Mining: Concepts and Techniques"* .Morgan Kaufmann Publishers, 2000.

[9] E.-H. Han, G. Karypis, and V. Kumar, *"Scalable Parallel Data mining for Association Rules,"* Proc. ACM SIGMOD 1997, May 1997.

[10] E-H. Han, G. Karypis, and V. Kumar, *"Scalable Parallel Data mining for Association Rules,"* IEEE Trans. Data and Knowledge Eng.,vol. 12, no. 3, May/June 2000.

[11] H. Cokrowijoyo, D. Taniar, *A framework for mining association rules in Data Warehouses,* Proc. IDEAL 2004, Lecture Notes in Computer Science, vol. 3177, Springer, Berlin, 2004, pp. 159–165.

[12] L. Dehaspe, L. Raedt, *Mining association rules in multiple relations*, Proc. ILP'97, Lecture Notes in Computer Science, vol. 1297, Springer, Berlin, 1997, pp. 125–132.

Authors Biography

B. Santhosh Kumar received Bachelor degree in Computer Science & Engg from Bharathiar University and Master of Engineering in Computer Science & Engg from Anna University Trichy. He has published several papers in Data mining. Currently he is working as a lecturer- Computer Science & Engg in CSI College of Engg, Ketti.

K.V. Rukmani received Master of Computer Application from Mother Teresa Womens' University and Master of Engineering in Computer Science & Engg from Anna University Coimbatore. She has published several papers in Data mining. Currently she is working as an Asst Professor and HoD in charge- Computer Science & Engg in CSI College of Engg, Ketti.

ISSN
2548-9712

ANALISA DAN PERBANDINGAN METODE ALGORITMA APRIORI DAN FP-GROWTH UNTUK MENCARI POLA DAERAH STRATEGIS PENGENALAN KAMPUS STUDI KASUS DI STKIP ADZKIA PADANG

Domi Sepri[1], M. Afdal[2]

[1]Pascasarjana Universitas Putra Indonesia "YPTK" Padang
[2]Program Studi Sistem Informasi UIN Suska Riau
[1]domisepri@gmail.com,[2]afdal.afatih@gmail.com

Abstrak

Sekolah Tinggi Keguruan dan Ilmu Pendidikan (STKIP) ADZKIA merupakan salah satu institusi pendidikan formal di kota Padang yang disahkan oleh pemerintah. Persaingan di dalam dunia bisnis, khususnya dalam bidang pendidikan membuat pihak perguruan tinggi harus mencari pola sasaran daerah yang strategis dalam pengenalan sekolah. Dengan semakin banyaknya STKIP di Kota Padang, membuat setiap sekolah berusaha mencari calon siswa baru kedaerah-daerah yang potensial. Salah satu cara yang dapat dilakukan untuk penentuan daerah strategis adalah dengan menggunakan teknik *DataMining*. Dari data-data mahasiswa yang ada disekolah dapat diolah mengunakan algoritma *Apriori* dan *FP-Growth* yang menjadi informasi baru untuk dimanfaatkan oleh dalam menentukan daerah yang strategis. Dalam penelitian ini penulis mencoba membandingkan hasil dari algoritma *Apriori* dan *FP-Growth* yang menggunakan data mahasiswa angkatan 2015/2016 dengan nilai *minsupport* = 0.05% dan nilai *minconfidence* = 0.7% telah diperoleh 19 *Association Rule* dan 2 rule tertinggi yang dapat dijadikan sebagai pengetahuan baru serta acuan berharga pada lingkup penelitian ini.

Kata Kunci: *Data Mining,Association Rule, Apriori, FP-Growth.*

1. PENDAHULUAN

1.1 Latar Belakang

Salah satu masalah yang sering dihadapi oleh instansi Perguruan tinggi swasta yang baru berdiri adalah masih belum maksimalnya jumlah mahasiswa yang diharapkan sesuai dari target instansi tersebut. Ini disebabkan oleh kurang maksimalnya strategi promosi dalam memperkenalkan kampusnya ke daerah potensial calon mahasiswa.

Pada Beberapa penelitian tentang *Data Mining*, telah dijelaskan pentingnya data mining dalam dunia pendidikan terutama dalam prediksi-prediksi kemajuan dunia pendidikan. Literatur Internasional terdapat contoh adanya penelitian tentang prediksi siswa yang harus mendapatkan perhatian khusus dari data – data siswa yang putus sekolah (*Baradwaj, 2010).*

Pada penelitian ini penulis mencoba membandingkan algoritma *Apriori* dan *FP-Growth* yang merupakan teknik Assosiation Rules dalam menentukan pola terbaik yang akan dijadikan rujukan dalam medapatkan pola-pola yang tersedia sebagai bahan mempromosikan kampus dikarenakan *FP-Growth* dan *Apriori*merupakan dua algoritma yang sama-sama mempunyai struktur datayang kuat, sehingga menghasilkan peluang *generate rule* yang baik (Yıldız Barış, 2010).

1.2. Perumusan Masalah

Berdasarkan latar belakang di atas, maka dirumuskan permasalahannya sebagai berikut :

1. Bagaimana menerapkan algoritma *Apriori* dan *FP-Growth* di dalam promosi pendidikan sehingga dalam proses tersebut dapat menentukan pola strategis untuk mempromosikan kampus?
2. Bagaimana melakukan pengujian Metode *Apriori* dengan *Tanagra* dan *FP-Growth* dengan *RapidMiner* sehingga menghasilkan penyelesaian masalah dalam mencari strategi promosi yang tepat?
3. Bagaimana perbandingan algoritma *Apriori* dan *FP-Growth* dalam menentukan pola daerah strategis pengenalan kampus ?

ISSN 2548-9712

1.3. Tujuan Penelitian

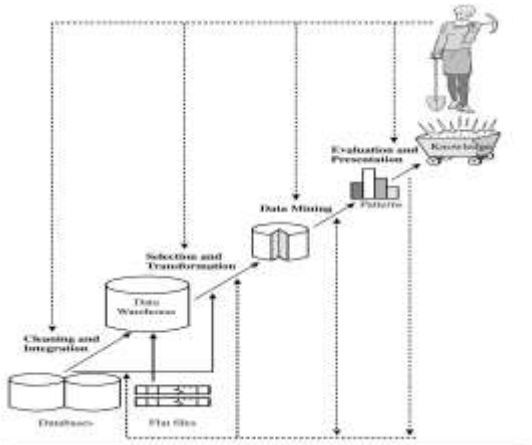
Penelitian ini bertujuan untuk memecahkan permasalahan yang telah dirumuskan di atas yaitu :

1. Untuk mengetahui strategi yang cocok untuk promosi pendidikan di STKIP ADZKIA Padang.
2. Untuk menganalisa keefektifan metode *Apriori* dan *FP- Growth* di dalam strategi promosi pendidikan di STKIP ADZKIA Padang.
3. Untuk melakukan pengujian metode *Apriori* mrnggunakan *Tanagra 1.4.50* dan *FP-Growth* menggunakan *tools Rapid-Miner 7.0.001* sehingga menghasilkan strategi promosi pendidikan yang terbaik.
4. Menghasilkan sebuah hubungan antara variabel atau item berdasarkan implementasi dari algoritma *Apriori & FP-Growth* .
5. Menguji hubungan variable atau item tersebut untuk mendapatkan pengetahuan baru dari proses ekstraksi menggunakan *Tools* yang ada.

2. LANDASAN TEORI

2.1. Knowledge Discovery in Database (KDD)

Menurut Tampubolon, *et al* (2013) *Knowledge Discovery in Database* (KDD) didefinisikan sebagai ekstraksi informasi potensial, implisit dan tidak dikenal dari sekumpulan data. Proses *Knowlegde Discovery in Database* melibatkan hasil proses *Data Mining* (proses pengekstrak kecenderungan suatu pola data), kemudian mengubah hasilnya secara akurat menjadi informasi yang mudah dipahami.



Gambar 2.1. *Data Mining* Sebagai Sebuah Langkah Dalam Proses *Knowledge Discovery*

2.2. Data Mining

Data Mining adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual (Benni R Siburian, 2013).

2.3. Association Rule

Menurut Tampubolon, *et al* (2013) Analisis asosiasi atau *association rule mining* adalah teknik *Data Mining* untuk menemukan aturan assosiatif antara suatu kombinasi item. Algoritma aturan asosiasi akan menggunakan data latihan, sesuai dengan pengertian *Data Mining*, untuk menghasilkan pengetahuan. Pengetahuan untuk mengetahui *item-item* belanja yang sering dibeli secara bersamaan dalam suatu waktu. Aturan asosiasi yang berbentuk "*if…then…*" atau "jika…maka…" merupakan pengetahuan yang dihasilkan dari fungsi aturan asosiasi. Aturan ini dihitung dari data yang sifatnya probabilistik.

Metodologi dasar analisis asosiasi terbagi menjadi dua tahap:

1. Analisa pola frekuensi tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai support dalam *database*. Nilai *support* sebuah *item* diperoleh dengan rumus berikut:

$$Support(A) = \frac{JumlahTransaksiMengandungA}{TotalTransaksi} x\ 100\%$$

2. Pembentukan aturan assosiatif

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan assosiatif yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan assosiatif A _B Nilai *confidence* dari aturan A _B diperoleh dari rumus berikut:

$$Confidence(A \rightarrow B) = \frac{Jumlah\ TransaksiMengandungA\ \&\ B}{Jumlah\ TransaksiMengandung\ A} x100\%$$

2.4. Algoritma Apriori

Apriori adalah suatu algoritma yang sudah sangat dikenal dalam melakukan pencarian frequent itemset dengan menggunakan teknik association rule. Algoritma *Apriori* menggunakan knowledge mengenai frequent itemset yang telah diketahui

ISSN 2548-9712

sebelumnya, untuk memproses informasi selanjutnya. Pada algoritma *Apriori* untuk menentukan kandidat-kandidat yang mungkin muncul dengan cara memperhatikan minimum support (Erwin, 2009).

Proses utama yang dilakukan dalam algoritma Apriori untuk mendapat *frequent itemset* yaitu (Erwin, 2009) :

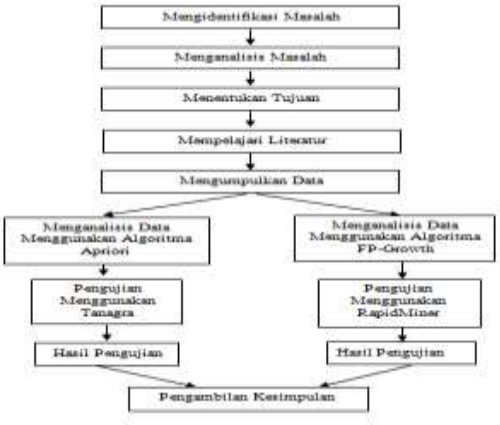
1. *Join* (penggabungan)
   Proses ini dilakukan dengan cara pengkombinasian item dengan item yang lainnya hingga tidak dapat terbentuk kombinasi lagi.
2. *Prune* (pemangkasan)
   Proses pemangkasan yaitu hasil dari item yang telah dikombinasikan kemudian dipangkas dengan menggunakan minimum *support* yang telah ditentukan oleh *user*.

2.5. Algoritma FP-Growth

Menurut Goldie (2012), *FP-growth* adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. *FP-growth* menggunakan pendekatan yang berbeda dari paradigma yang digunakan pada algoritma Apriori.

3. METODOLOGI PENELITIAN

Dalam metodologi penelitian ada urutan kerangka kerja yang harus diikuti, urutan kerangka kerja ini merupakan gambaran dari langkah–langkah yang harus dilalui agar penelitian ini bisa berjalan dengan baik. Kerangka kerja yang harus diikuti bisa dilihat pada gambar 3.1.



Gambar III.1 Kerangka Kerja

Berdasarkan gambar 3.1 dijabarkan urutan–urutan langkah kerja sebagai berikut :

1, Mengidentifikasi Masalah

Masalah yang diidentifikasi dalam penelitian ini untuk mendapatkan strategi daerah untuk promosi yang efektif dan efisien dalam mendapatkan calon mahasiswa baru ditengah ketatnya persaingan antar Perguruan Tinggi.

2. Menganalisis Masalah

Analisis masalah pada penelitian ini dilakukan dengan dua metode yaitu metode deskriptif dan metode komparatif.

a. Metode Deskriptif
   Pada metode ini data yang ada dikumpulkan, disusun, dikelompokkan dan dianalisis sehingga diperoleh beberapa gambaran yang jelas pada masalah yang akan dibahas.
b. Metode Komperatif
   Pada metode ini analisis dilakukan dengan cara membandingkan teori dan praktek sehingga diperoleh gambaran yang jelas tentang persamaan dan perbedaan di antara algoritma Apriori dan FP-Growth.

3. Menentukan Tujuan

Menetukan tujuan akhir dari penelitian ini.

4. Mempelajari Literatur

Literatur-literatur yang dipakai sebagai bahan referensi dalam penelitian ini adalah dari jurnal-jurnal ilmiah, modul pembelajaran dan buku tentang *Data Mining*. literatur-literatur ini akan menjadi pedoman untuk melakukan penelitian agar memudahkan proses penelitian.

5. Pengujian Menggunakan *Tanagra 1.4.50* dan *Rapidminer 7.0.001*

Pada tahap ini, *rule* di tes kembali atau diuji lagi menggunakan sistem *data mining* yang sudah ada. *Tools* yang digunakan sebagai pengujian sistem adalah *Rapidminer* dan Tanagra 1.4.50.

6. Pengambilan Kesimpulan

Setelah diuji, hasil analisis antara cara manual dan pengujian memanfaatkan *tools* akan terlihat perbandingannya. Langkah berikutnya adalah mengambil kesimpulan dari analisa penerapan pada algoritma FP-Growth maupun Apriori terhadap *knowledge* yang baru didapat yaitu berupa penentuan dalam menentukan strategi dalam mempromosikan pendidikan.

ISSN 2548-9712

4. ANALISA DAN PERANCANGAN

Pada penelitian ini proses analisa dibagi dalam 2 jenis, yaitu analisa pada menggunakan algoritma *Apriori* dan analisa data menggunakan algoritma *FP-Growth*. Karena terdapat perbedaan langkah – langkah pengolahan data pada kedua algoritma tersebut sebagai bahan dalam membandingkan kedua algoritma tersebut.

4.1. Pemilihan Variabel

Data sampel yang diuji dalam penelitian ini sebanyak 20 sampel data mahasiswa baru di kampus STKIP ADZKIA TA. 2015 format seperti berikut :

1. Asal Sekolah (SMA Negeri, SMA Swasta, SMK Negeri, SMK Swasta, MA Negeri, MA Swasta)
2. Jurusan di SMA (IPA, IPS, Kejuruan)
3. Alamat Asal (Kabupaten Asal)
4. Sumber Informasi STKIP

Tabel 1 Data Sampel 20 mahasiswa

No	Kabupaten	SMA Asal	Jurusan	Informasi STKIP
1	[illegible]	[illegible]	IPS	Brosur
2	[illegible]	[illegible]	IPA	Teman/ Orang Lain
3	Padang	[illegible]	[illegible]	Teman/ Orang Lain
4	[illegible]	[illegible]	IPS	Brosur
5	[illegible]	[illegible]	[illegible]	Brosur
6	[illegible]	[illegible]	IPS	Teman/ Orang Lain
7	Padang	[illegible]	IPA	Brosur
8	[illegible]	[illegible]	IPS	Brosur
9	[illegible]	[illegible]	IPS	Brosur
10	[illegible]	[illegible]	IPS	Brosur
11	[illegible]	[illegible]	IPA	Teman/ Orang Lain
12	[illegible]	[illegible]	Akuntansi	Teman/ Orang Lain
13	Padang	[illegible]	IPS	Brosur
14	[illegible]	[illegible]	IPS	Teman/ Orang Lain
15	[illegible]	[illegible]	[illegible]	Teman/ Orang Lain
16	[illegible]	[illegible]	IPA	Teman/ Orang Lain
17	[illegible]	[illegible]	IPS	Teman/ Orang Lain
18	[illegible]	[illegible]	IPS	Brosur
19	Padang	[illegible]	Akuntansi	Brosur
20	[illegible]	[illegible]	IPS	Teman/ Orang Lain

4.2. Analisa dengan Algoritma Apriori

Proses berikutnya pada algoritma *Apriori* adalah menseleksi item yang tidak berada di bawah *Support* yang ditentukan. Dalam hal ini *minimal support* yang ditentukan adalah 30% sehingga akan menghasilkan seleksi item

Tabel IV.2 item Minimum Support 30%

Item	Frekwensi	*Support* (Transaksi/ Total Transaksi)	Persen *Support*
PS	14	14/20 = 0.7	70%
SMAN	15	15/20 = 0.75	75%
IPS	11	11/20 = 0.55	55%
BR	11	11/20 = 0.55	55%
TM	9	9/20 = 0.45	45%

Setelah dilakukan kombinasi 2 item, maka akan dihasilkan kombinasi item yang berada di bawah nilai *Support*

Tabel 3 Item Minimum Support 30%

Item	Frekwensi	*Support* (Transaksi/ Total Transaksi)	Persen *Support*
PS, SMAN	12	12/20 = 0.6	60%
PS, IPS	9	15/20 = 0.75	75%
PS, BR	7	11/20 = 0.55	55%
PS, TM	7	11/20 = 0.55	55%
SMAN, IPS	11	9/20 = 0.45	45%
SMAN, BR	8	8/20 = 0.4	40 %
SMAN, TM	7	7/20 = 0.35	35 %
IPS, BR	7	7/20 = 0.35	35%

Setelah itu akan dilakukan kombinasi 3 itemset akan dihasilkan kombinasi di atas *Support* 30% yaitu :

Tabel 4 Item Minimum Support 30%

Item	Frekwensi	*Support* (Transaksi/ Total Transaksi)	Persen *Support*
PS, SMAN, IPS	9	9/20 = 0.45	45%
PS, SMAN, BR	6	6/20 = 0.3	30%
PS, SMAN, TM	6	6/20 = 0.3	30%
PS, IPS, BR	6	6/20 = 0.3	30%
SMAN, IPS, BR	7	7/20 = 0.35	35 %

Setelah itu akan dilakukan kombinasi 4 itemset akan dihasilkan kombinasi di atas *Support* 30% yaitu :

Tabel 5 Item Minimum Support 30%

Item	Frekwensi	*Support* (Transaksi/ Total Transaksi)	Persen *Support*
PS, SMAN, IPS, BR	6	6/20 = 0.3	30%

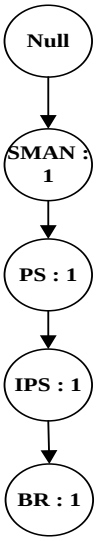
4.3. Analisa Data dengan Algoritma *FP-Growth*

TID 1 : SMAN, PS, IPS, BR

Proses pembentukan *tree* pada tahap *TID 1* adalah dengan membuat cabang *null* yang

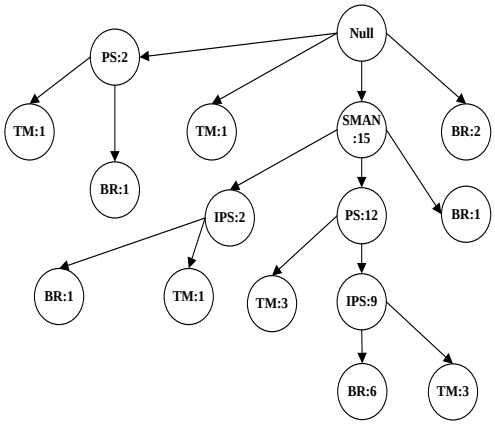
ISSN 2548-9712

merupakan pangkal terbentuknya pohon pada algoritma *FP-Growth*, dari null maka akan dibuat cabang berturut-turut mulai dari SMAN, PS, IPS, hingga BR.



Gambar 4.1 Proses FP-Tree TID 1

TID 20 : SMAN, IPS, TM



Gambar 4.2 Proses FP-Tree TID *20*

Setalah proses dari *FP-Tree* selesai dan dihasilkan *Frequent itemset* sebagai berikut:

Tabel 6 Kombinasi Itemset Algoritma FP-Growth

Suffix	*Frequent Itemset*
TM	{TM, PS}, {TM, SMAN}, {TM, PS, SMAN}
BR	{BR, IPS}, {BR, PS }, {BR, SMAN}, {BR, PS, IPS}, {BR, PS, SMAN}, {BR, IPS, SMAN}, {BR, PS, IPS, SMAN}
IPS	{IPS, PS}, {IPS, SMAN}, {IPS, PS, SMAN}
PS	{PS, SMAN }
SMAN	-

5. HASIL DAN PEMBAHASAN

5.1. Implementasi Sistem

Implementasi yang dilakukan adalah melakukan penerapan semua data mahasiswa baru STKIP ADZKIA 2015/2016 untuk mendapatkan hasil pemetaan pola daerah strategis untuk pengenalan kampus menggunakan algoritma *Apriori* dan *FP-Growth* ke dalam 2 aplikasi data mining, yaitu :

1. Algoritma *Apriori* menggunakan *Tanagra 1.4.50.*
2. Algoritma FP-Growth menggunakan *RapidMiner 7.0.001.*

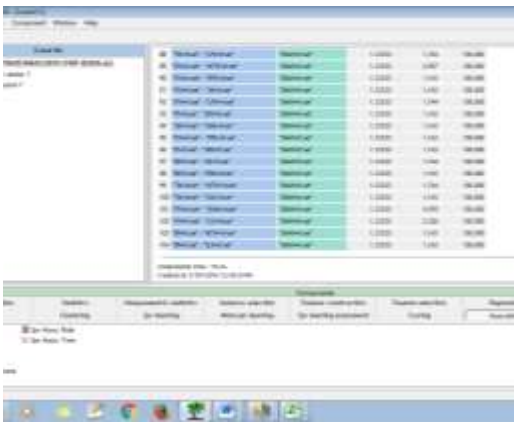
Spesifikasi inputan nilai *Support* dan *Confidence* yang dimasukkan dalam pengujian mengggunakan *Tanagra* dan *RapidMiner* adalah :

1. Nilai Min *Support* adalah : 0.01
2. Nilai Min *Confidence* : 1.0

Nilai ini disesuaikan dengan kebutuhan untuk mendapatkan pola strategis dari masing-masing daerah.

5.2. Implementasi Algoritma *Apriori* Menggunakan *Tanagra 1.4.50*

Setelah data ditransformasi ke dalam bentuk *biner* pada *Microsoft Excel*, selanjutnya adalah melakukan uji data pada aplikasi *Tanagra 1.4.50* , adapun langkah-langkahnya sebagai berikut :
Buka *software Tanagra 1.4.50*, maka akan tampil tampilan awal *software.* Tampilan awal *software* terdiri dari menu file, diagram, window dan help. Tmpilan awal juga berisikan *components* yang merupakan komponen data mining yang bisa digunakan pada *Tanagra*.



Gambar 5.1 Hasil *Rule Algoritma Apriori*

ISSN 2548-9712

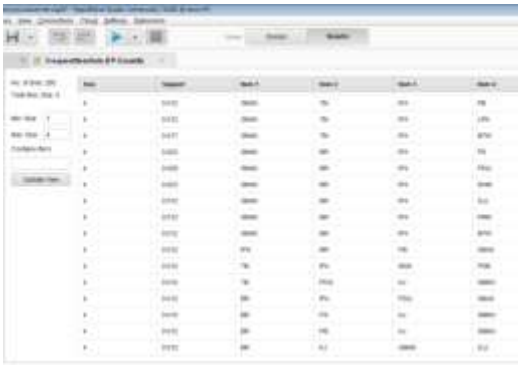
Dari Gambar 5.1. diketahui bahwa jumlah *rule* yang dihasilkan adalah 104 *rule* dan telah memetakan daerah beserta pemetaan sumber informasi pada daerah tersebut.

5.3. Implementasi Algoritma *FP-Growth* Menggunakan *RapidMiner*

Pengujian *algoritma FP-Growth* menggunakan *RapidMiner* dilakukan dengan nilai *Minimum Support* dan *Confidence* yang sama dengan pengujian *algoritma Apriori* menggunakan *Tanagra* yaitu 0.01 untuk *minimum Support* dan 1.0 untuk nilai *minimum Confidence.* Tahapan pertama adalah membuka aplikasi *RapidMiner* yang telah *diinstall*, kemudian melakukan tahapan proses pengambilan data sampai pada akhir proses pelaksanaan seperti pada gambar 5.2.

Gambar 5.2 Menjalankan Proses *Frequent Itemset RapidMiner*

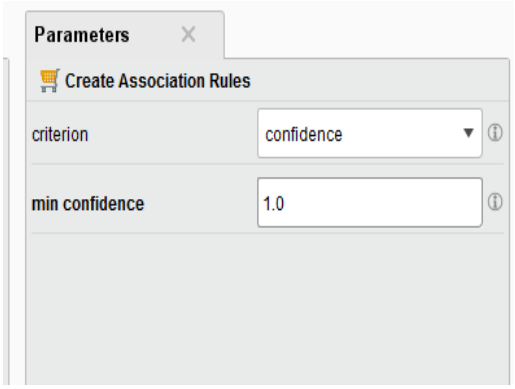
Setelah proses dijalankan , maka akan tampil hasil *frequent itemset* yang diinginkan seperti pada gambar 5.3.



Gambar 5.3 Hasil *Frequent Itemset FP-Growth*

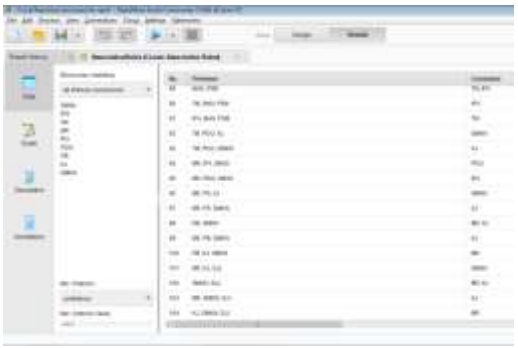
Jika dibandingkan dengan hasil *frequent itemset* yang dihasilkan algoritma *Apriori* menggunakan *Tanagra*, maka akan kita dapati kesamaan hasil *frequent itemset* yang dihasilkan, yaitu sama-sama menghasilkan 51 *itemset*.

Berikutnya adalah memasukkan operator *Create Association Rule* dengan melakukan pencarian pada bagian operator, sehingga di dalam area *process* telah terdapat 4 operator. Kemudian mengatur parameter pada operator *Create Association Rule*. Cara settingnya adalah dengan mengklik *create association rule* dan mengatur *min Confidence* menjadi 1.0 seperti pada gambar 5.4.



Gambar 5.4 Setting Parameter *Create Association Rule*

Setelah proses *setting* dilakukan pada *rapidminer* dan dijalankan maka hasilnya adalah:



Gambar 5.5 Hasil *Rule* Pada *RapidMiner* menggunakan *Algoritma FP-Growth*

Dari pengujian algoritma FP-Growth menggunakan *RapidMiner*, dihasilkan 104 *rule* asosiasi yang sama dengan pengujian algoritma *Apriori* menggunakan *Tanagra*.

ISSN 2548-9712

5.4. Hasil Uji Algoritma *Apriori* dan FP-Growth

Sesuai dengan batasan masalah pada Bab I, maka pengujian yang dilakukan adalah pengujian pada *rule* asosiasi yang dihasilkan dan waktu eksekusi yang diperlukan.

Agar lebih jelasnya hasil perbandingan antara kedua algoritma maka bisa dilihat pada tabel 7

Tabel 7 Hasil Perbandingan algoritma

Indikator Perbandingan	Algoritma *Apriori*	Algoritma FP-Growth
Jumlah *Rule* Asosiasi yang dihasilkan	104 *Rule*	104 *Rule*
Waktu Mendapatkan *Frequent Itemset*	*Error* (karena banyaknya jumlah *itemset* yang harus ditampilkan)	0 detik
Waktu Mendapatkan Rule Asosiasi	0 detik	0 detik

Indikator perbadingannya adalah pada jumlah *rule* asosiasi yang dihasilkan dan eksekusi waktu proses algoritma dalam menghasilkan *frequent itemset* maupun *rule* asosiasi.

5.4.1 Hasil Pengujian *Rule* yang Dihasilkan

Dari Hasil uji kedua algoritma di atas diketahui bahwa hasil perhitungan *rule* yaitu seperti tabel dibawah ini:

Tabel 8 Kesamaan *Rule* yang Dihasilkan kedua Algoritma

NO	IF	THEN	SUPPORT	CONFIDENCE
1	BR, PS, KJ	SMKN	0.0116279	1
2	BR, PS, SMKN	KJ	0.0116279	1
3	IPS, TM, PS	SMAN	0.0581395	1
4	BR, IPA, DHM	SMAN	0.0232558	1
5	IPS, TM, DHM	SMAN	0.0174419	1
6	TM, SLK	SMAN, IPS	0.0116279	1
7	SMAN, TM, SLK	IPS	0.0116279	1
8	IPS, TM, SLK	SMAN	0.0116279	1
9	SMS, JM	SMAN, IPS	0.0116279	1
10	SMAN, SMS, JM	IPS	0.0116279	1
11	IPS, SMS, JM	SMAN	0.0116279	1
12	BR, AG	SMAN, IPS	0.0174419	1
13	SMAN, BR, AG	IPS	0.0174419	1
14	IPS, BR, AG	SMAN	0.0174419	1
15	BR, IPA, SIJ	SMAN	0.0116279	1
16	BR, KJ, SIJ	SMKN	0.0116279	1
17	SMKN, SIJ	BR, KJ	0.0116279	1
18	BR, SMKN, SIJ	KJ	0.0116279	1
19	BR, PRM	SMAN, IPA	0.0116279	1
20	SMAN, BR, PRM	IPA	0.0116279	1
21	BR, IPA, PRM	SMAN	0.0116279	1
22	TM, IPA, LPK	SMAN	0.0116279	1
23	IPS, BR, SMAS	PB	0.0116279	1
24	PB, SMAS	IPS, BR	0.0116279	1
25	IPS, PB, SMAS	BR	0.0116279	1
26	BR, PB, SMAS	IPS	0.0116279	1
27	PB, SMKN	BR, KJ	0.0116279	1
28	BR, PB, SMKN	KJ	0.0116279	1
29	PB, KJ, SMKN	BR	0.0116279	1
30	TM, PDG, KJ	SMKN	0.0116279	1
31	TM, PDG, SMKN	KJ	0.0116279	1
32	BR, IPA, SMAS	PDG	0.0116279	1
33	BR, PDG, SMAS	IPA	0.0116279	1
34	MAN, PSM	TM, IPA	0.0116279	1
35	TM, MAN, PSM	IPA	0.0116279	1
36	IPA, MAN, PSM	TM	0.0116279	1
37	TM, MTW	SMAN, IPA	0.0174419	1
38	SMAN, TM, MTW	IPA	0.0174419	1
39	TM, IPA, MTW	SMAN	0.0174419	1
40	BR, MTW	SMAN, IPA	0.0116279	1

Hasil perhitungan *rule* yang didapat dari penerapan masing-masing algoritma ke dalam aplikasi menemukan kesamaan jumlah *rule* yang didapat yaitu 104 *rule* yang telah menyatakan pemetaan daerah dan pola strategisnya dengan nilai *Confidence* adalah 1.00 (*Strong Confidence*), berikut 42 *rule* kombinasi 4 item yang menyatakan pemetaan pola daerah strategis dari 104 *rule* yang dihasilkan oleh kedua algoritma:

Dari *rule* yang tertera pada tabel 5.3 dapat kita uraikan pola daerah strategis masing-masing daerah adalah:

1. Daerah Pesisir selatan adalah daerah mahasiswa paling banyak berasal dengan jumlah nilai *Support* pada *rule* tertinggi sebesar 0.058 % dan dari 3 *rule* kombinasi yang didapat pada *rule* tertinggi menyatakan bahwa jika mereka berasal dari Jurusan IPS serta mendapatkan informasi dari teman/orang lain maka ia berasal dari SMA Negeri. Disamping itu di daerah Pesisir Selatan yang berasal dari jurusan Kejuruan dan mendapatkan informasi dari brosur maka berasal dari SMK Negeri .
2. Daerah Dhamasraya mempunyai pola daerah strategis adalah jika ia berasal dari jurusan IPA dan mendapatkan info dari brosur, maka ia berasal dari SMA Negeri. Disamping Itu jika ia berasal dari jurusan IPS dan mendapatkan info dari teman maka ia berasal dari SMA Negeri.
3. Daerah solok mempunyai pola daerah strategis adalah jika ia berasal dari jurusan IPS dan berasal dari SMA negeri maka ia mendapatkan info dari SMS atau Teman/Orang lain
4. Daerah jambi mempunyai pola daerah strategis dalah jika ia berasal dari jurusan IPS dan dari SMA Negeri maka ia mendapatkan info dari SMS
5. Daerah Agam mempunyai pola dderah strategis adalah jika ia berasal dari SMA Negeri dan jurusan IPS, maka ia mwendapatkan informasi dari brosur.
6. Daerah sijunjung mempunyai pola daerah strategis adalah jika ia berasal dari SMA Negeri, dan jurusan IPA maka ia mendapatkan info dari Brosur. Disamping itu jika iaberasala dari SMK Negeri dan

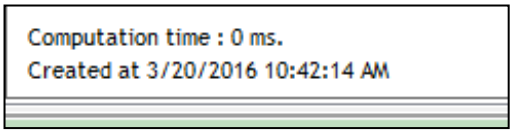
ISSN 2548-9712

jurusan Kejuruan, maka ia mendapatkan informasi dari brosur.

7. Daerah Pariaman mempunyai pola daerah strategis yaitu jika mahasiswa berasal dari SMA Negeri dan dari Jurusan IPA, maka ia mendapatkan informasi dari brosur
8. Daerah Kabupaten Lima Puluh Kota mempunyai pola daerah strategis yaitu jika mahasiswa berasal SMA Negeri dan dari jurusan IPA maka ia mendapatkan informasi dari Teman/Orang Lain.
9. Daerah Pasaman Barat mempunyai pola daerah strategis yaitu jika mahasiswa berasal dari SMA Swasta dan jurusan IPS maka ia mendapatkan informasi dari brosur. Di samping itu jika mahasiswa tersebut berasal dari SMK negeri dan jurusan kejuruan, maka ia mendapatkan informasi juga dari brosur.
10. Daerah Padang mempunyai pola daerah strategis yaitu jika ia berasal dari dari SMK Negeri dan jurusan kejuruan, maka ia mendapatkan informasi dari Orang lain / Teman. Di samping itu jika mahasiswa berasal dari SMA Swasta dan jurusan IPA maka ia mendapatkan informasi dari brosur.
11. Daerah Pasaman mempunyai pola daerah strategis adalah jika mahasiswa berasal dari MA Negeri dan jurusan IPA, maka ia mendapatkan informasi dari Teman/Orang Lain.
12. Daerah Kep.Mentawai mempunyai pola daerah strategis adalah jika ia berasal dari SMA Negeri dan jurusan IPA maka ia mendapatkan informasi dari Teman/Orang Lain atau Brosur.

5.4.2. Hasil Pengujian Waktu Eksekusi

a. Dalam menghasilkan *Frequent Itemset* pada *algoritma Apriori* menggunakan *Tanagra* untuk 4 digit kombinasi menghasilkan pesan error pada *software* dikarenakan banyaknya *frequent itemset* yang harus ditampilkan. Sedangkan pada *Algoritma FP-Growth* menggunakan *software RapidMiner* menghasilkan waktu 0 detik.
b. Dalam pengujian performa waktu dalam mengeksekusi data menggunakan *RapidMiner* maupun *Tanagra* dalam mendapatkan *rule* dari dua algoritma dengan 172 record data menghasilkan kesamaan waktu eksekusi yaitu 0 detik seperti pada gambar V.6 dan V.7 di bawah.



Gambar 5.6 Hasil Perhitungan Waktu Eksekusi *Tanagra*

Hasil pengujian waktu eksekusi pada *software RapidMiner* dengan hasil komputasi 0 detik.

Gambar 5.7 Hasil Perhitungan Waktu Eksekusi *RapidMiner*

Terjadinya kesamaan waktu dalam pengeksekusian ini dikareanakan data yang dieksekusi berjumlah sedikit sehingga tidak bisa melihat perbandingan performa waktu diantara kedua algoritma.

6. KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan hasil analisa dan pengujian yang telah dilakukan, maka dapat disimpulkan beberapa hal yang terkait dengan penelitian ini. Setelah dilakukan pengujian didapatkan beberapa hubungan antara jurusan SMA,kategori SMA, Kabupaten Asal dan SumberI nformasi dalam pemetaan daerah strategis menggunakan algoritma *apriori* dan *FP-Growth*. Adapun kesimpulan yang didapatkan sebagai berikut :

1. Pada penelitian ini dari pengujian dengan nilai minimum support sebesar 0.05 dan minimum confidence 0.7 terdapat 19 *association rule* dengan 8 association rule yang mempunyai kombinasi daerah yaitu hanya pada daerah Pesisir Selatan.
2. Dari pengujian data STMIK Adzkia tahun ajaran 2015/2016, maka diketahui bahwa siswa yang dari daerah kabupaten Pesisir Selatan, berasal dari SMA Negeri, dari jurusan IPS adalah daerah potensial yang mendapatkan informasi dari teman/orang lain merupakan sasaran strategis dalam pengenalan kampus dengan tingkat

ISSN 2548-9712

kepercayaan (*Nilai Confidence)* sebesar 100%.

3. Dalam perbandingan algoritma *apriori* dan *fp-growth*, penulis mendapatkan hasil yang lebih baik pada algoritma *fp-growth* dalam menghasilkan *frequent itemset* karena melakukan pencarian *frequent itemset* menggunakan sistem *FP-Tree* sehingga tidak perlu melakukan *scan database* yang berulang jika sudah ditemukan kombinasi *itemset* yang sudah ditemukan sebelumnya.
4. Waktu yang dihasilkan dengan 171 record data mahasiswa antara algoritma menggunakan *tanagra 1.4* dengan *algoritmafp-growth* menggunakan *rapidminer 7.0.001* menghasilkan waktu proses yang sama yaitu 0 detik.

6.2. Saran

Berdasarkan hasil analisa dan pengujian yang diperoleh, saran dari penulis untuk penelitian selanjutnya adalah :

1. Pada penelitian selanjutnya sebaiknya bisa menambahkan beberapa variabel lagi, sehingga dapat menampilkan lebih banyak lagi hubungan yang terbentuk dari *association rule* yang dihasilkan, sehingga dapat membentuk *knowledge* yang lebih detail dan bermanfaat.
2. Data yang digunakan dalam penelitian perbandingan algoritma *apriori* dan *fp-growth* seharusnya mempunyai nilai record yang besar di atas 3000 data agar lebih menampakkan kelebihan dan kelemahan dari masing-masing algoritma.
3. Pengujian perbandingan antara algoritma *apriori* dan *fp-Growth* sebaiknya dilakukan dalam satu aplikasi yang sama agar lebih memfalidkan hasil perbandingan .

DAFTAR PUSTAKA

[1] Anis Kurniawati., 2014.“Pemetaan Pola Hubungan Program Studi Dengan Algoritma Apriori – Studi Kasus Spmu Unnes.” *Edu Komputika Journal*. 1 (1). 51-58

[2] Azhari., dan Anshori., 2009. ”Pendekatan Aturan Asosiasi Untuk Analisis Pergerakan Saham.” *Seminar Nasional Informatika 2009 (semnasIF 2009) ISSN: 1979-2328 UPN Veteran*. E-183-189.

[3] Benni R Siburian.,2014.“Aplikasi Data Mining Untuk Menampilkan Tingkat Kelulusan Mahasiswa Dengan Algoritma Apriori.” *Pelita Informatika Budi Darma*. VII (2). 56-61.

[4] Brijesh Kumar Baradwaj.,2011. ”Mining Educational Data to Analize Students’performance”. *International Journal of Advanced Computer Science and Applications* vol.2, no.6

[5] Charanjit Bambrah., 2014.”Mining Association Rule in Student Assessment Data”. *International Journal of Advanced Research in Computer and Communication Engineering* ol.3, issue 3,

[6] Efori Buulolo.,2013. ”Implementasi Algoritma Apriori Pada Sistem Persediaan Obat (Studi Kasus : Apotik Rumah Sakit Estomihi Medan)”. *Pelita Informatika Budi Darma*. IV (1). 71-83.

[7] Erwin.,2009.”Analisis Market Basket Dengan Algoritma Apriori dan FP-Growth.” *Jurnal Generic ISSN 1907-4093*

[8] Goldie Gunadi., dan Dana Indra Sensuse.,2012.“ Penerapan Metode *Data Mining Market Basket Analysis* Terhadap Data Penjualan Produk BukuDengan Menggunakan Algoritma Apriori Dan *Frequent Pattern Growth* (Fp-Growth) :Studi Kasus Percetakan PT. Gramedia.” *Jurnal TELEMATIKA MKOM*. 1 (1). 118 132

[9] Kennedi Tampubolon, Hoga, S. dan Bobby, R.,2013. “Implementasi Data Mining Algoritma Apriori Pada Sistem Persediaan Alat-Alat Kesehatan.” *Informasi dan Teknologi Ilmiah (INTI)*. 1 (1). 93-106.

[10] Lokandra C. Reddy, Venkatadri.M 2011. “A Review on Data Mining from Past to the Future”. *International Journal of Computer Applications (0975-8887)*

[11] Ririanti.,2014.”Implementasi Algoritma Fp-Growth Pada Aplikasi Prediksi Persediaan Sepeda Motor (Studi Kasus Pt. Pilar Deli Labumas). *“ Pelita Informatika Budi Darma.* VI (1). 139-144.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 139 (2018) 275–279

Procedia Computer Science

www.elsevier.com/locate/procedia

The International Academy of Information Technology and Quantitative Management, the Peter Kiewit Institute, University of Nebraska

FP-Growth based Regular Behaviors Auditing in Electric Management Information System

Jiye Wang*, Zhihua Cheng

Department of Information and Communication Technology, State Grid Corporation of China, Beijing, 100000, China

Abstract

In Electric Management Information System (MIS), there are some users who do not comply with all operation/behavior regulations and make the similar mistakes continuously even though they are not on purpose. These behaviors are a huge threat to the system security. In this paper, we propose a method to detect these regular behaviors with association rules mining algorithm FP-Growth. First, the user log is separated into operation sets each of which contains user operation in a continuous period. Then we divide the operation sets of all users into two catalogs: normal and abnormal based on if a security problem has happened around the corresponding period of operation set. Next, we apply the FP-Growth algorithm in both normal and abnormal operation sets to generate the frequent patterns. Finally, the abnormal pattern is compared with normal ones to determine the regular behaviors that may be dangerous to the system. We test the proposed algorithm in the user log files generated from a simulated electric management information system. The experiment results indicate the proposed method can effectively detect the regular user behavior that could cause the system security problems.

© 2018 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer review under responsibility of the scientific committee of The International Academy of Information Technology and Quantitative Management, the Peter Kiewit Institute, University of Nebraska.

Keywords: behavior auditing; FP-growth; Electric Management Information System

1. Introduction

Currently, management information system security has become the main consideration in many companies since it influences basically every aspect of the management system from human resources management to financial affairs, from daily production to business dealing. Therefore user behavior auditing is essential to ensure

* Corresponding author. Tel.: +86-10-66597486; fax: +86-10-66597486.
E-mail address: jiyewang@sgcc.com.cn.

1877-0509 © 2018 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer review under responsibility of the scientific committee of The International Academy of Information Technology and Quantitative Management, the Peter Kiewit Institute, University of Nebraska.
10.1016/j.procs.2018.10.268

the stable running of MIS. One important aspect of behavior auditing is responsibility investigation. However, the existing responsibility investigations in MIS are mainly based on manually checking which is low efficiency and high cost. Only these huge errors or problems that course large damage are fully investigated. During that period, the investigation could also influence the regular operation of the whole system. While quite many system alarms or minor errors are usually ignored since the lack of manpower. Therefore the user behaviors that lead to these alarms are lack of supervision and correction, which may lead to larger damage in the future.

According to static, many alarms and errors in MIS are coursed unintentionally such as forgetting log out, modifying system configuration, password leakage and etc. These behaviors may happen again and again if no auditing is performed and the user is alerted. However, as discussed these minor system alerts or errors are usually ignored without further investigation considering the high cost.

To deal with that problem, we propose an association rule mining based algorithm to detect the relationship between user behavior and system alarm. The regular alarms or system security problems are recorded in a system issue log file, based on which we divide the user behavior into regular (no system alarm) and abnormal (system alarm happens) set. For each set, the frequent patterns are detected using association rule mining algorithm FP-growth. By comparing the mined patterns from regular set to abnormal set, we can detect the frequent behaviors that may lead to the system alarms. The rest of the paper will explain the proposed method in details. Section 2 gives the related work of behavior auditing and associate rules mining. The main framework and methodology are described in Section 3. Section 4 gives an experiment to illustrate the efficiency of the proposed algorithm using simulated user log files. Finally, Section 5 concludes the whole paper.

2. 2 Related work

User behaviors auditing is essential for information system security [1]. Steinbart et al. [2] examined how the quality of the relationship between the internal audit and the information security functions affects objective measures of the overall effectiveness of an organization's information security efforts. Zerbino et al. [3] presented a five-step Process Mining-based methodology for information system audits and validated it through a case in a freight export port process managed by a Port Community System (PCS), an open electronic platform enabling information exchange among port stakeholders.

Association rules mining methods can be used in behavior auditing to find out the reasons coursing the system problem. Borah and Nath et al. [4] introduced a technique to identify the symptoms and risk factors for three adverse diseases: cardiovascular disease, hepatitis and breast cancer, in terms of rare association rules. Son et al. [5] proposed a mining algorithm based on animal migration optimization (AMO), called ARM–AMO, to reduce the number of association rules. Wang et al. [6] constructed a frequent itemsets tree in parallel with mining frequent itemsets to improve the efficiency and interpretability of rule mining without generating candidate itemsets. Han et al. [7] presented the classic FP-growth algorithm for frequent pattern detection and association rule mining which has better performance especial for large dataset than Apriori algorithm [8] another classic association rule mining method.

3. Methodology

The workflow chart of the proposed method is shown in Figure 1. To identify the user behaviors leading to the system alarms/errors, we first divide the user behavior log into two sets: regular and abnormal based on whether system alarms/errors happened. Then frequent patterns are detected for both sets as exclude items and possible candidate. Finally the frequent behavior patterns in abnormal set are filtered by the regular patterns and the left patterns are considered as reason for the system alarms or errors.

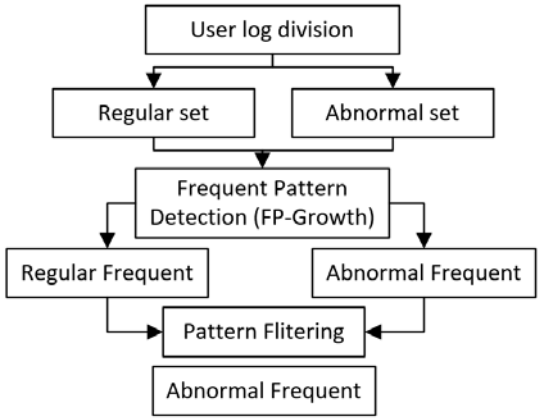


Fig. 1. Workflow of Frequent Abnormal Behavior Detection

3.1. User Operation Set Creation

In this paper we focus on the frequent behaviors that may damage the MIS. The basic idea to detect the frequent abnormal or harm behaviors is to separate the regular frequent patterns from the frequent patterns that related to the system alarms or errors.

In the daily operation, there are many warning and minor errors generated in MIS. To detect the reason coursing these alerts, we have to first collect the user operations related to the alerts or errors. In this paper, time is used as a main consideration to evaluate the relevant degree. If an operation is conducted before a system alert and their time difference is within a threshold, this operation is considered relevant to the system alert. Besides time, user group and accessed resources can also be used to determine the association between user behavior and system alert. In these identified related user behavior, the regular user operations are always exists, so we have to remove the normal ones from the related behavior set which left the candidates of behaviors leading to the system alert or errors.

Usually in MIS, we have two types of log files: user behavior log and system running log. The user log contains every operations that all user have done, while the system log includes the system running situation and system alert/error information. Both logs have timestamp for each records. Then for each alert/error in system log, we collect the user operation records related to the system alert or error happen before a certain time threshold. This threshold can be set to different value according to the feature of alert or error. The rest records in user log will be used as normal behavior.

3.2. Frequent Pattern Generation

To increase the scalability especially for the system without huge data volume, the system alert/error and user behavior are first merged into different classes, which can increase the chance to find the frequent pattern. Then the processed data are structured as follows:

$Issue_i$: [$behavior_j$] $j \in 1,m$
$behavior_j = \{a_1,a_2,\ldots a_k\}$

It means for every defined issue (a class of system alerts or errors), there is a corresponding set of behaviors. The behavior is composed by a serious continues operation from one user within a certain period for example 5 or 10 minutes.

Based on the data structure, FP-growth algorithm is applied to detect the frequent pattern for every defined system issue. The FP-Growth Algorithm is proposed by Jiawei Han in 2000. As a classic association mining algorithm, it is efficient and scalable for mining the complete set of frequent patterns by pattern fragment growth, using an extended prefix-tree structure for storing compressed and crucial information about frequent patterns named frequent-pattern tree (FP-tree). Compared with other algorithms such as Apriori Algorithm and the TreeProjection, FP-growth has better performance especially for large volume of data.

In this paper, we use FP-growth to detect the frequent pattern of user behavior for system issues to identify the regular user operations that may course the system failure. For each generated system issue, we find the operation sequence with minimum support and minimum confidence larger than the predefined thresholds. Then these operation sequence will be further filter for the reason that leads to the system issue.

3.3. Behavior Pattern Filtering

According to the pattern generation rule, regular operations are also recorded when a system issue happens. Therefore the detected frequent patterns include the regular behavior pattern. To remove these regular pattern, we define a normal class similar with system issue but including the user behaviors not related to any system issue. The frequent pattern for normal behaviors can be detected based on the normal behavior dataset. Since this normal behavior dataset contains huge volume of data, the minimum support should be reduced accordingly.

Since the system running is stable, we can pre-calculated the normal behavior patterns and directly apply it for abnormal behavior detection, which could reduce the computation load quite much. However, the normal behavior patterns should be updated regularly such as one month or on season to reflect the changes in the system.

If a new type of system issue (alerts or errors) are detected, the frequent patterns related to the issue are generated using the proposed method. Then for each detected frequent pattern, if it does not exist in the normal pattern list, the pattern is labeled as abnormal behavior for further analysis. The proposed abnormal behavior detection method can summarize the relationship behind the separate operations and identify the frequent user operations that lead to the certain system issues. Based on the pattern, we can further find out the user with bad habit and provide the corresponding suggestions to improve the user behavior.

4. Experiment

Considering the privacy protection, the experiment is conducted with simulated user and system log file. To demonstrate the proposed framework, a simulation platform is implemented to generate the user and system log of electric MIS using Simpy program that is a process-based discrete-event simulation framework using python. It can simulate the user activity and system responses by defining their logic behavior and interactions.

In this paper, we define 100 users working in an electric MIS with 10 types of resources and there are 12 types of system issues could be coursed by user incorrect operations. The simulation period lasts one month when the users are perform both normal and abnormal behaviors. During the simulation, more than one million user operations and 2000 system issue records are generated as user and system log file.

Then the log files will be processed as and tagged as regular or with certain system issue. For each system issue, the FP-growth algorithm is applied to detect the frequent behavior patterns. In this paper, the FP-growth algorithm is implemented according to https://github.com/evandempsey/fp-growth. According to the experimental results on simulation data, the accuracy for detecting regular incorrect behaviors leading to the system alert/error is over 83%. Even though the test data is different from the real user and system log files, the proposed algorithm can be helpful to find out the reasons for the system alerts or errors.

In another experiment of the simulated electric MIS, we create 10 users with bad habits that can course the system alerts or errors. Then the user operations are recorded in user log and system information are recorded in system log as described. We test the proposed abnormal pattern detection method based on the generated simulation data. According to the experiment results, 8 users are detected with abnormal behavior and 2 normal users are miss detected as abnormal pattern since they are mainly working during the system issue happens. The results suggest the proposed user profile based method can find out the frequent pattern related to the system alerts or errors, but the accuracy should be improved by associating the system issue with its related behavior more precisely.

5. Conclusions

In this paper we propose an algorithm to detect the association rules between user behavior and system issues (alerts or errors). The proposed method first classifies the user log according to its relationship with system issues, and then employs FP-growth to detect the frequent patterns that happen before the system issues. Next the detected behavior patterns are filtered by removing the regular patterns detected from the user behavior logs with no system issue happens. We test our algorithm on the simulated data, the experiment results indicate the efficiency of the proposed method by accuracy over 80%. However the proposed method has not yet applied on the real data which is still under testing. The further study can be focus on how to find the related user behaviors of a certain system issue more precisely which can reduce the computation complexity and improve the detection accuracy.

References

[1] Paul John Steinbart, Robyn L. Raschke, Graham Gal, William N. Dilla, The relationship between internal audit and information security: An exploratory investigation, International Journal of Accounting Information Systems, Volume 13, Issue 3, 2012, Pages 228-243,

[2] Paul John Steinbart, Robyn L. Raschke, Graham Gal, William N. Dilla, The influence of a good relationship between the internal audit and information security functions on information security outcomes, Accounting, Organizations and Society, 2018,

[3] Pierluigi Zerbino, Davide Aloini, Riccardo Dulmin, Valeria Mininno, Process-mining-enabled audit of information systems: Methodology and an application, Expert Systems with Applications, Volume 110, 2018, Pages 80-92,

[4] Anindita Borah, Bhabesh Nath, Identifying Risk Factors for Adverse Diseases using Dynamic Rare Association Rule Mining, Expert Systems with Applications, 2018,

[5] Le Hoang Son, Francisco Chiclana, Raghavendra Kumar, Mamta Mittal, Manju Khari, Jyotir Moy Chatterjee, Sung Wook Baik, ARM–AMO: An efficient association rule mining algorithm based on animal migration optimization, Knowledge-Based Systems, Volume 154, 2018, Pages 68-80,

[6] Ling Wang, Jianyao Meng, Peipei Xu, Kaixiang Peng, Mining temporal association rules with frequent itemsets tree, Applied Soft Computing, Volume 62, 2018, Pages 817-829,

[7] Agrawal, Rakesh; and Srikant, Ramakrishnan; Fast algorithms for mining association rules in large databases, in Bocca, Jorge B.; Jarke, Matthias; and Zaniolo, Carlo; editors, Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), Santiago, Chile, September 1994, pages 487-499

[8] Han (2000). "Mining Frequent Patterns Without Candidate Generation". Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. SIGMOD '00: 1–12.

Pemantauan Pola Pengunjung Perpustakaan Dengan Metode *FP-Growth* Guna Meningkatkan Fungsi Perpustakaan

Dudih Gustian [1,*], Kudin Rustaman [1], Aditya Nurfitri [1], Yunita Gusti Nurani [1], Neng Resti Noviansyah [1], Anggy Pradiftha Junfithrana [2]

[1] Sistem Informasi; Universitas Nusa Putra; Jl. Raya Cibolang Kaler No.21 Sukabumi 43155; Telp: 021-78839513; Fax: 021-78839421; e-mail: dudih@nusaputra.ac.id, kudin.rustaman@nusaputra.ac.id, aditya.nurfitri@nusaputra.ac.id, yunita.gusti@nusaputra.ac.id, neng.resti@nusaputra.ac.id

[2] Teknik Elektro; Universitas Nusa Putra; Jl. Raya Cibolang Kaler No.21 Sukabumi 43155; Telp: 021-78839513; Fax: 021-78839421; e-mail: mr.pradiftha@nusaputra.ac.id

* Korespondensi: e-mail: dudih@nusaputra.ac.id

Diterima: 6 Mei 2019 ; Review: 20 Mei 2019 ; Disetujui: 21 Juni 2019

Cara sitasi: Gustian D, Rustaman K, Nurfitri A, Nurani YG, Noviansyah NR, Junfithrana AP. 2019. Pemantauan Pola Pengunjung Perpustakaan Dengan Metode *FP-Growth* Guna Meningkatkan Fungsi Perpustakaan. Informatics For Educators and Professionals. 3 (2): 149 – 160.

Abstrak: Perpustakaan dilingkungan Universitas XYZ jadi kebutuhan wajib mahasiswa, dosen, dan pegawai untuk mencari informasi mutakhir ataupun untuk menyelesaikan tugas. Masalah yang terjadi ialah kurang berfungsinya perpustakaan sebagai sumber belajar dikarenakan kemudahan mendapatkan informasi melalui internet, fasilitas, kenyamanan dan pelayanan perpustakaan lalu keterbatasan jumlah selain itu koleksi literatur di perpustakaan yang kurang memadai juga membuat mahasiswa tidak mau baca dan pinjam buku di perpustakaan. Metode yang di gunakan dalam penelitian ini yaitu FP-Growth, karena bekerja lebih baik dan lebih cepat dibandingkan algoritma Apriori. Penelitian ini memberikan solusi dan manfaat yang di tawarkan untuk mengetahui pola pengunjung perpustakaan serta dapat membentuk dampak yang positif bagi peningkatan pengunjung perpustakaan sebagai sumber belajar. Pola pengunjung perpustakaan dengan nilai support dengan kombinasi J,H,K,E dengan nilai 0.76 – 0.795. Sementara untuk nilai confidence diantara 0.97 – 0.99 dengan kombinasi E,H,K. Terakhir untuk lift ratio dengan kombinasi pola E-M, E-H, E-J, J-H dengan nilai lift ratio 0.99 – 1.03. Aplikasi ini telah di uji ke beberapa user dan memperoleh nilai 82,25 sehingga dikatagorikan cukup membantu manajemen dalam memantau pola pengunjung.

Kata kunci: *FP-Growth*, Nilai Confidence, Perpustakaan

Abstract: *The library in the University of XYZ is a mandatory requirement for students, lecturers, and employees to find the latest information or to complete assignments. The problem that occurs is the lack of functioning of the library as a source of learning due to the ease of getting information through the internet, facilities and library services, and the limited number of books. The method used in this study is FP-Growth, because it works better and faster than the Apriori algorithm. This research provides solutions and benefits offered to know the pattern of library visitors and can form a positive impact on increasing library visitors as a learning resource. The pattern of library visitors with support values with a combination of J, H, K, E with a value of 0.76 - 0.795. While for the confidence value between 0.97 - 0.99 with a combination of E, H, K. Finally for lift ratio with a combination of E-M, E-H, E-J, J-H patterns with lift ratio 0.99 - 1.03. This application has been tested to several users and scored 82.25 so that it is categorized enough to help management in monitoring visitor patterns.*

Copyright@2019. P2M STMIK BINA INSANI

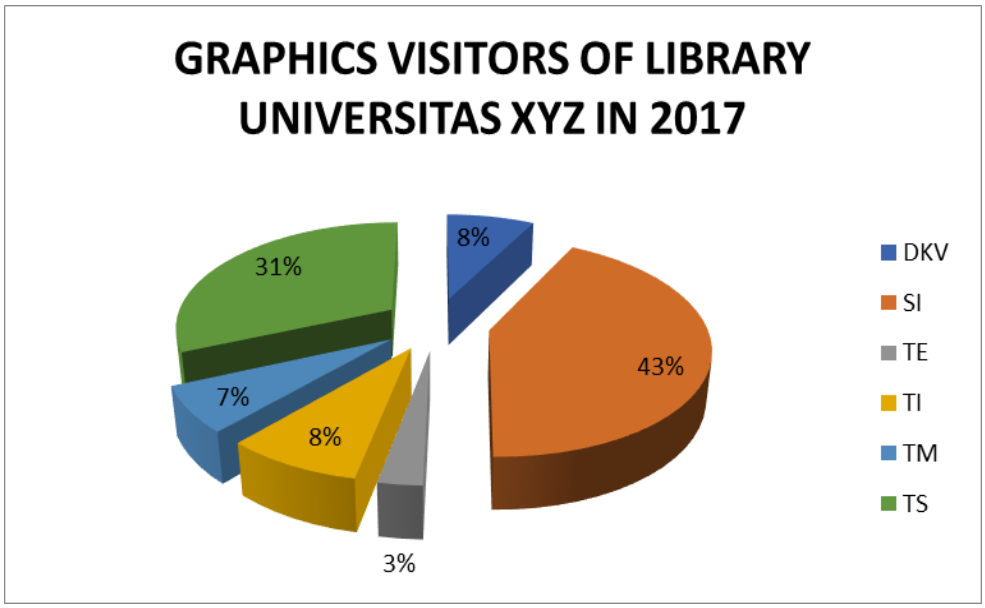
E-ISSN: 2548-3412 ; 149 – 160

Keywords: *FP-Growth, Library, Value of Confidence*

1. Pendahuluan

Perpustakaan jadi pilihan saat ini dengan variasi jenis pustaka yang beredar dan harga yang sulit dijangkau oleh sebagian besar masyarakat. Terkait hal tersebut, perpustakaan yang berada dilingkungan Universitas XYZ menjadi kebutuhan utama mahasiswa, dosen, dan pegawai untuk mencari informasi mutakhir ataupun guna menyelesaikan tugas.

Menurut data statistik pada tahun 2017 tingkat pengunjung perpustakaan dibeberapa prodi di Universitas XYZ seperti pada gambar 1 menunjukan bahwa untuk jurusan Desain Komunikasi Visual (DKV) sebanyak 8%, jurusan Sistem Informasi (SI) sebanyak 43%, jurusan Teknik Elektronika (TE) sebanyak 3%, jurusan Teknik Informatika (TI) sebanyak 8%, Teknik Mesin (TM) sebanyak 7 % dan Teknik Sipil (TS) Sebanyak 31%.

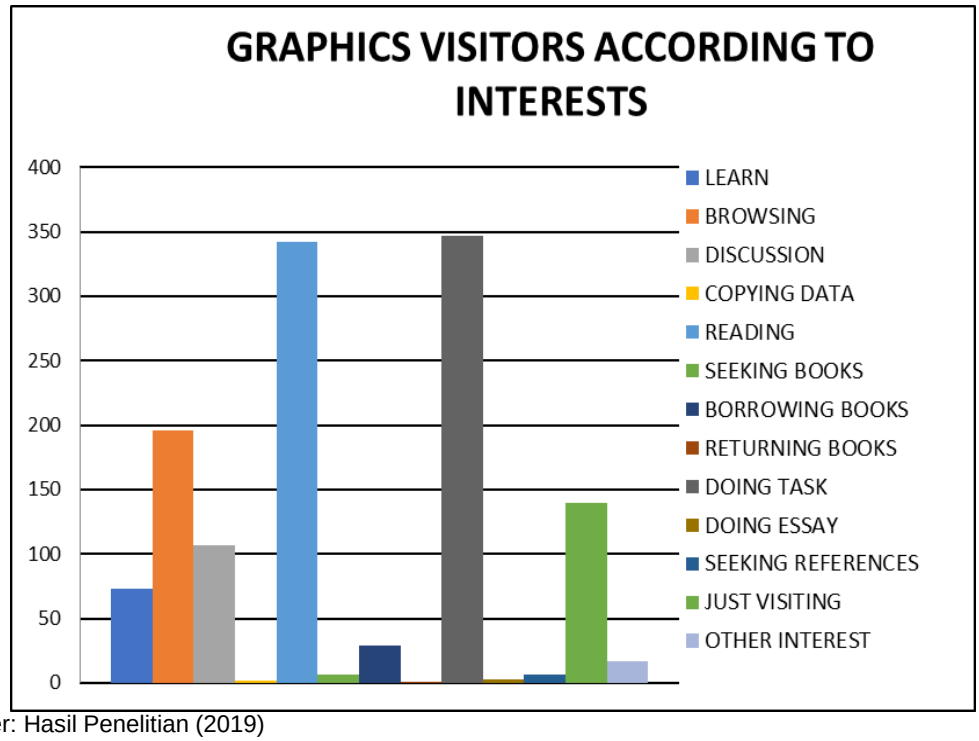


Sumber: Hasil Penelitian (2019)

Gambar 1. Grafik Pengunjung Perpustakaan

Masalah yang terjadi ialah kurang berfungsinya Perpustakaan sebagai sumber belajar dikarenakan beberapa faktor seperti, kemudahan mendapatkan informasi melalui internet sehingga membuat orang yang biasanya membaca diperpustakaan kini meninggalkan kebiasaannya untuk membaca diperpustakaan. Selain itu terbatasnya jumlah koleksi literature di perpustakaan yang kurang memadai juga membuat mahasiswa berat untuk baca dan pinjam di perpustakaan karena buku yang dibutuhkan itu tidak tersedia. Fasilitas, kenyamanan dan pelayanan perpustakaan juga menjadi beberapa penyebab minat mahasiswa untuk menggunakan perpustakaan sebagai penunjang perkuliahan mereka. Beberapa masalah tersebut disebabkan salah satu faktor karena minat baca dikalangan mahasiswa yang rendah pada umumnya. Hal ini diperoleh dengan hasil pengamatan pada tahun 2017 dari gambar 1 dari total sejumlah mahasiswa yang mengunjungi perpustakaan dan gambar kepentingan mahasiswa mengunjungi perpustakaan sebagai penunjang perkuliahan.

Pada gambar 2 menunjukkan pola pengunjung dapat dikelompokkan menjadi 4 bagian besar yang terdiri dari jumlah pola pengunjung dibawah 50 ialah mengerjakan skripsi, mengembalikan buku, menyalin data, mencari referensi, pinjam buku, melihat buku dan keperingan lainnya. Kemudian diatas 50 dan dibawah 100 ialah belajar dan diskusi, lalu kelompok selanjutnya diatas 100 dan dibawah 150 hanya ialah pola hanya berkunjung dan terakhir nilai diatas 300 dan dibawah 350 ialah belajar dan mengerjakan tugas. Hal ini memberikan gambaran bahwa rendahnya kebiasaan mahasiswa seperti pada pola pengunjung dibawah 50 yang memang merupakan salah satu fungsi perpustakaan sebagai sumber belajar.



Sumber: Hasil Penelitian (2019)

Gambar 2. Pola Pengujung Perpustakaan

Sebuah survei tentang association rule [Diah Larasati et al, 2015] membuktikan bahwa algoritma FP- *Growth* merupakan algoritma *association rule* tercepat di antara algoritma *association rule* yang lain, yaitu AIS dan Apriori. Apriori dibandingkan dengan algoritma yang lain dalam proses *association rule*, tetapi ada kekurangan dalam pencarian *frequent itemset*, dimana harus melakukan scanning database berulang-ulang untuk setiap kombinasi item, kondisi ini tentu menyebabkan lama proses scanning basis data. Selain itu dibutuhkan *generate candidate* besar untuk mendapatkan kombinasi item dari *database*. Sementara FP-*Growth* merupakan pengembangan dari algoritma Apriori, namun kekurangan dari Apriori diperbaiki oleh algoritma FP-*Growth*. Dalam Apriori diperlukan *generate candidate* untuk mendapatkan *frequent itemsets*, tetapi tidak dilakukan karena FP-*Growth* menggunakan konsep pembangunan *tree* dalam pencarian *frequent itemsets*. Penelitian tersebut memberikan hasil yang sama dengan survei yang dilakukan, yaitu algoritma FP-*Growth* bekerja lebih baik dan lebih cepat dibandingkan algoritma Apriori. Dari pertimbangan tersebut diputuskan dalam penelitian ini digunakan algoritma FP-*Growth* karena beberapa kelebihan yang ada.

Dalam penelitian ini digunakan adalah FP-Growth, berdasarkan studi literature beberapa penelitian sebelumnya bahwa didapatkan pola peminjaman buku, sehingga hasil penelitian pertama rule yang dihasilkan tiap bulan tidak selalu sama. Kedua rule hasil berdasarkan semester/tahun sebagian besar merupakan akumulasi dari rule yang dihasilkan perbulan [Miranda Nur Qolbi Aprilina et al, 2015]. Algoritma FP- Growth efisien dalam mencari frequent itemset dalam kumpulan data dengan membangkitkan struktur prefix-tree. Dari hasil pengumpulan data selanjutnya dilakukan analisis guna menentukan variabel yang akan dijadikan input pada RapidMiner, dimana variabel yang ada diolah sesuai jumlah permintaan dari setiap pelanggan. Pada tahap evaluasi akhir, hasil dari pengolahan data dengan bantuan Software Rapid Miner guna mendapatkan nilai statistik yang akan dibandingkan dengan hasil prediksi yang selama ini dilakukan dengan sistem rata-rata. Sehingga dapat ditarik kesimpulan terhadap hasil prediksi mana yang lebih mendekati atau dengan errornya yang lebih kecil [Hilda Herasmus,2017]. Agoritma *FP-Growth* digunakan juga dalam rekomendasi pemesanan sparepart dengan parameter data transaksi pembelian dengan jumlah 12 data yang diambil secara random dari transaksi dari PT Rosalia. Penelitian ini menggunakan untuk pendekatan nilai asosiasi data. Hasil yang dicapai dalam penelitian ini adalah adanya rule system rekomendasi pembelian sparepart dengan minimum support 0.15 dan minimum confidence 0.8

E-ISSN: 2548-3412 ; 149 – 160

[Nur Rohman Ardani & Nur Fitria, 2016]. Dalam buku Data Mining: A Book Recommender System Using Frequent Pattern Algorithm, dijelaskan bahwa sistem rekomendasi menggunakan algoritma asosiasi dengan beberapara parameter yang diperoeh dari data perpustakaan online dengan tujuan penelitian ini untuk melihat hubungan antara siswa dan buku-buku yang ada di perpustakaan, dengan demikian menemukan pola tersembunyi di dalamnya [Joshua J.V et al, 2016]. Menggunakan metode *FP-Growth* untuk memahami perilaku pengguna situs web melalui proses data mining data akses web. Dimana hasilnya yaitu dapat meningkatkan efektivitasnya. Kemudian dapat digunakan oleh administrator sistem atau perancang web dan dapat mengatur sistem dengan menentukan kesalahan sistem yang terjadi, tautan rusak dan rusak [K.Dharmarajan et al, 2016]. Melakukan penelitian dengan metode *FP-Growth* guna penelusuran pola konsumsi obat. Hasil penelitian ini bahwa Algoritma ini digunakan dalam menemukan pola obat berdasarkan penyakit yang didiagnosis. Hasilnya ialah pola kombinasi obat untuk sepuluh diagnosa penyakit. Pola-pola ini dapat digunakan untuk menyarankan dokter membuat resep medis. Juga, penelitian ini juga menemukan 21 pola kombinasi obat yang serupa untuk menangani beberapa penyakit [Rini Anggrainingsih et al, 2017].

Dari kondisi tersebut jelas bahwa dalam penelitian ini digunakan untuk mengetahui penyebab rendahnya pola dibawah 50 melalui pola kebiasaan mahasiswa dalam mengunjugi perpustakaan, sehingga pola pengunjung khususnya nilai dibawah 50 agar pola tersebut berubah menjadi minimal masuk ke pola diatasnya yakni belajar dan diskusi atau bahkan sampai ke pola selanjutnya yakni belajar dan mengerjakan tugas.

Penelitian ini memberikan solusi untuk mengetahui penyebab rendahnya pola dibawah 50 melalui kebiasaan mahasiswa dalam mengunjugi perpustakaan melalui hubungan dari beberapa pola kemungkinan, sehingga nilai dibawah 50 agar meningkat. Dimana hubungan ini nantinya dianalisis bagaimana langkah perventif dari pola yang ada guna menjadikan perpustakaan sebagai sumber belajar bagi para mahasiswa.

Hasil penelitian ini memberikan kontribusi bagi manajemen dalam melengkapi kekurangan yang ada khususnya infrastruktur. Sehingga sebagai bahan pertimbangan dan bahan evaluasi untuk memperbaiki sistem, prosedur dan kualitas perpustakaan Universitas XYZ. Hal ini tentu saja dapat membentuk dampak yang positif bagi peningkatan pengunjung perpustakaan serta mengoptimalkan fungsi perpustakaan sebagai sumber belajar.

2. Metode Penelitian

2.1. Tahap *FP-Growth*

Terdapat tiga tahapan dari *FP-Growth*, yaitu tahap pertama pembangkitan *conditional pattern base*, tahap kedua pembangkitan *conditional FP-Tree* dan tahapan ketiga pencarian *frequent item set*. Tahap pertama pembangkitan *conditional pattern base* adalah subdatabase yang berisi *prefix path* (himpunan item terurut yang mengawali k-*itemsets*), dan *suffix pattern* (k-itemsets). Contohnya, sebuah *itemset* yang telah tersusun berdasarkan *support descending order* {I6, I3, I1, I13, I16}, apabila I16 merupakan *suffix pattern* maka, I6, I3, I1, I13 ialah prefix path. Pembangkitan *conditional pattern base* didapatkan melalui FP-*tree* yang telah dibangun berdasarkan sebuah basis data transaksi. Tahapan kedua pembangkitan *conditional FP-Tree support count* bagi tiap item pada setiap *conditional pattern base* dijumlahkan, item yang memiliki jumlah *support count* lebih besar sama dengan min_sup akan dibangkitkan menjadi sebuah *tree* yang disebut dengan conditional FP-*tree*. Pada tahap ketiga pencarian *frequent item set* terjadi apabila *Conditional* Fp-*tree* merupakan single path, maka diperoleh frequent itemsets dengan melakukan kombinasi item untuk setiap *Conditional* Fp-*tree*. Jika bukan *single path* maka dilakukan pembangkitan FP *Growth* secara rekursif.

2.2. *Association Rule*

Association rule ialah bagian dari bidang data mining guna menentukan suatu aturan asosiatif agar memenuhi syarat minimum untuk *support* (minsup), *confidance* (minconf) pada

sebuah basis data [Witten et.al, 2017]. Pada tahap ini untuk menentukan nilai support dan confidence pada setiap item set dengan Persamaan (1).

$$\text{Support (A)} = \frac{Jumlah\ transaksi\ item\ A}{Total\ transaksi} \quad \ldots\ldots(1)$$

Lalu guna memperoleh nilai support dari dua item dapat diperoleh dengan persamaan (2).

$$\text{Support (A,B)} = P(A \cap B) = \frac{Jumlah\ transaksi\ A\ \&\ B}{Total\ transaksi} \quad \ldots\ldots(2)$$

Setelah semua frequent item dan large item set diperoleh lalu dicari syarat minimum confidence (mincof) dengan menggunakan Persamaan (3).

$$\text{Confidence } (A \rightarrow B) = P(A\ B) = \frac{\text{JumlahtransaksimengandungA\&B}}{\text{JumlahtransaksiyangmengandungA}} \quad \ldots\ldots(3)$$

Lalu dilakukan analisis dengan lift ratio, dimana bertujuan guna mengukur seberapa penting rule yang telah terbentuk berdasar nilai support dan confidence. Dimana lift ratio ialah perbandingan antara confidence dengan nilai benchmark confidence, dimana benchmark confidence merupakan perbandingan jumlah item consequent terhadap total jumlah transaksi.

$$\text{Benchmark- Confidence} = \frac{Nc}{N} \quad \ldots\ldots(4)$$

$$\text{LiftRatio} = \frac{Confidence\ (A,C)}{Benchmarkconfidence\ (A,C)} \quad \ldots\ldots(5)$$

Jika nilai lift ratio lebih besar dari 1, memberi arti adanya manfaat dari aturan itu lebih tinggi dari nilai lift ratio maka lebih besar kekuatan asosiasi tadi .

2.3. Uji Kehandalan GUI

Quisioner dengan Metode SQA (*Software Quality Assurance*). Dalam SQA terdapat tiga point pengertian dari kualitas software, diantaranya sebagai berikut ini: 1)Kebutuhan *software* ialah dasar darimana kualitas diukur. 2) Standar-standar spesifik yang menentukan kriteria pengembangan dalam menuntun pembangunan software. 3) Terdapat kebutuhan yang implisit yang sering tidak diperhatikan (contohnya: keinginan guna pemeliharaan yang terbaik [Ria K Handayani, 2012].

Skor tiap responden = <skor audiability> * 0.10 + <skor accurancy> *0.10 + skor completeness> * 0.15 + <skor error tolerance> * 0.10 + <skor excecution efficiency> * 0.10 + <skor operability> * 0.15 + <skor simplicity> * 0.15 + <skor training> * 0.15.

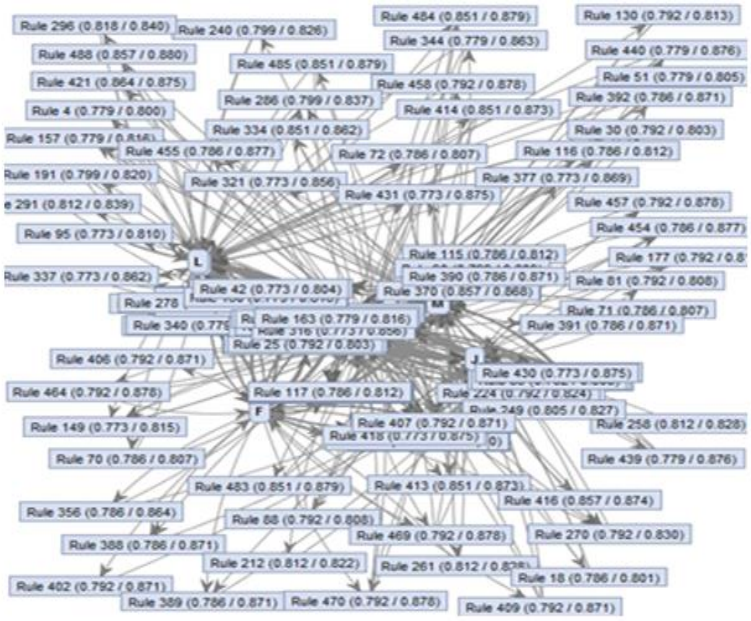
Dalam evaluasi sesuai kriteria, penelitian ini meggunakan rumus:

$$\text{Nilai rata – rata responden} : \frac{\text{Jumlah Skor Responden}}{\text{Jumlah Responden}}$$

3. Hasil dan Pembahasan

Dari hasil analisa data mining dengan Rapid Miner, yang mana jumlah rules yang dihasilkan yaitu 18 rules. Kita bisa ambil beberapa dari rules ini guna dibentuk dasar kuat dalam mengetahui kegiatan apa saja yang lebih banyak dilakukan oleh pengunjung perpustakaan. Hal ini diperlihatkan pada gambar 3:

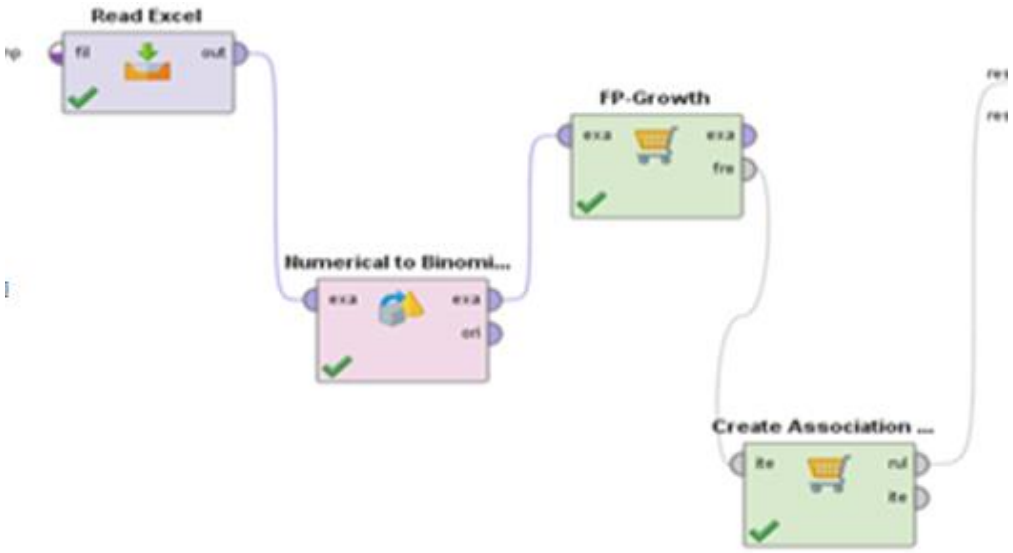
E-ISSN: 2548-3412 ; 149 – 160



Sumber: Hasil Penelitian (2019)

Gambar 3. Hasil Bentuk *Graph View*

Dalam penelitian ini data diambil dari data pengunjung perpustakaan Universitas XYZ tahun 2017, yaitu untuk jurusan Desain Komunikasi Visual (DKV) sebanyak 98 pengunjung, jurusan Sistem Informasi (SI) sebanyak 541 pengunjung, jurusan Teknik Elektronika (TE) sebanyak 35 pengunjung, jurusan Teknik Informatika (TI) sebanyak 108 pengunjung, Teknik Mesin (TM) sebanyak 87 pengunjung dan Teknik Sipil (TS) Sebanyak 400 pengunjung. Nantinya data ini digunakan di framework Rapid Miner untuk menghasilkan nilai minimum support , nilai minimum confidence dan suatu model treenya.

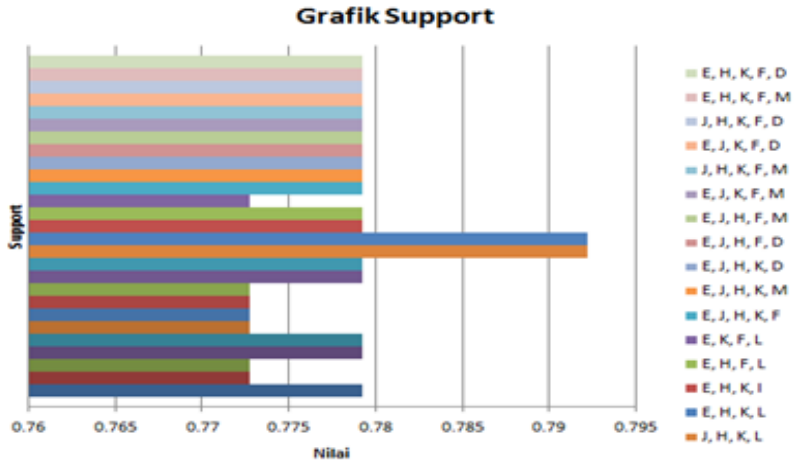


Sumber: Hasil Penelitian (2019)

Gambar 4. Konfigurasi Dalam *Rapid Miner*

3.1. Nilai *Support*

Dari hasil pengolahan dengan *rapid miner* diperoleh jumlah nilai *support* sekitar 1689, namun dipilih nilai support yang memiliki nilai diatas 70% dengan kombinasi premis diatas 4.



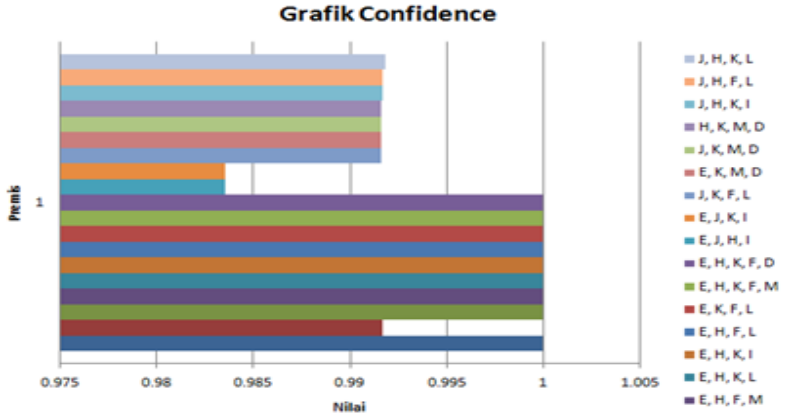
Sumber: Hasil Penelitian (2019)

Gambar 5. Nilai *Support*

Dalam gambar 5 terlihat bahwa nilai support untuk kombinasi premis 4 dan 5 berada diantara 0,76 sampai 0,795. Namun ada kombinasi premis yang menempati urutan teratas yakni J,H,K,L (mencari buku, mengerjakan skripsi, mencari referensi, hanya berkunjung) dengan E,H,K,L (mengembalikan buku, mengerjakan skripsi, mencari referensi, hanya berkunjung). Kondisi bisa dibuat keputusan yakni jika mahasiswa itu hanya berkunjung, mengerjakan skripsi, mencari referensi maka besar kemungkinan dia akan mencari buku. Lalu jika mahasiswa itu hanya berkunjung, mengerjakan skripsi, mencari referensi maka besar kemungkinan dia akan mengembalikan buku.

3.2. Nilai *Confident*

Jumlah nilai *confindence* yang dihasilkan sama dengan jumlah nilai *support*, namun peneliti memilih nilai *confidence* diatas 0,975 – 1.



Sumber: Hasil Penelitian (2019)

Gambar 6. Nilai *Confidence*

Gambar 6 menunjukkan bahwa terdapat 9 nilai *confidence* yang mendapat nilai 1, sisanya ada diantara 0.97 – 0,99. Terdapat hal unik dalam pola tersebut baik dengan jumlah 3

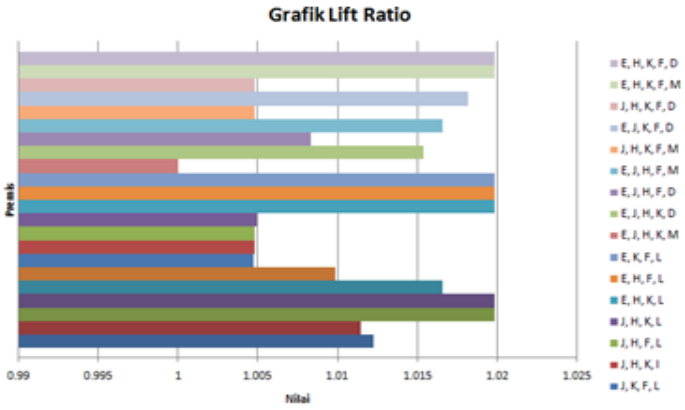
E-ISSN: 2548-3412 ; 149 – 160

maupun 4. Ada 2 kebiasaan mahasiswa dalam jumlah pola 3 yakni jika mahasisa mengerjakan skripsi, mencari referensi/belajar maka disebabkan karna hanya berkunjung. Sementara untuk jumlah pola 4 yakni diakibatkan keterkaitan dengan pola mengembalikan buku dan mengerjakan skripsi bisa membuat mahasiswa mengerjakan hal lainnya.

3.3. *Lift Ratio*

Jumlah nilai lift ratio yang dihasilkan sama dengan jumlah nilai *support*, namun peneliti memilih nilai *confidence* diatas 0,99 – 1.03.

Pada gambar 7 tersebut, bisa dilihat bahwa terdapat hubungan dari pola yang terbentuk baik untuk jumlah pola 3, 4 dan 5. Pola E,M (mengembalikan buku, kepentingan lain), E,H (mengembalikan buku, mengerjakan skripsi), E,J (mengembalikan buku, mencari buku) dan J,H (mencari buku, mengerjakan skripsi) mempunyai kecenderungan hubungan dengan pola lainnya.



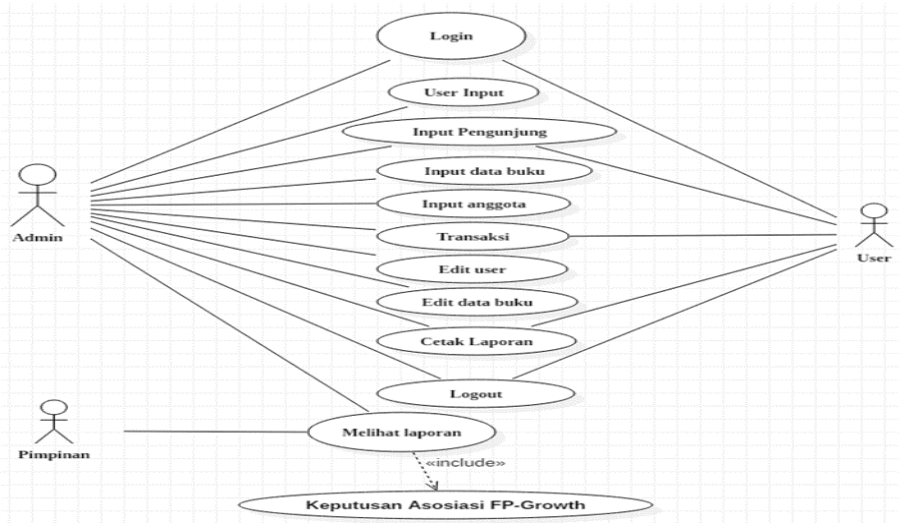
Sumber: Hasil Penelitian (2019)

Gambar 7. Nilia *Lift Ratio*

3.4. Hasil Implementasi

1. *Use Case Diagram*

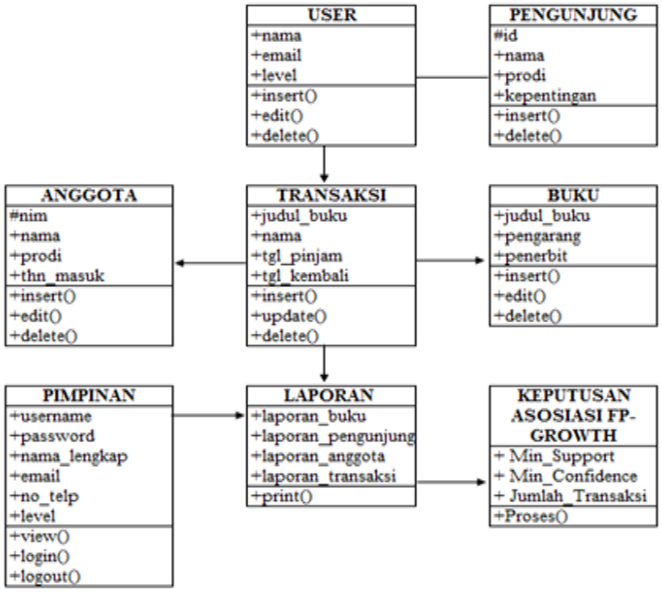
Gambar 8 memperlihatkan admin bisa melakukan semua fungsi yang ada pada sistem. Dimana admin harus login terlebih dahulu ke sistem. Sedangkan *actor* pimpinan dapat melihat hasil laporan yang telah diolah admin.



Sumber: Hasil Penelitian (2019)

Gambar 8. *Use case* yang dibuat

2. *Class Diagram*



Sumber: Hasil Penelitian (2019)

Gambar 9. *Class* diagram yang dibuat

Pada gambar 9 terdiri dari enam obyek diantaranya *user*, pengunjung, anggota, transaksi, buku, laporan dan pimpinan. Dimana untuk obyek transaksi dapat diakses oleh anggota, *user* yang terdiri dari pengunjung. Sementara obyek laporan dapat diakses oleh obyek pimpinan.

3. Implementasi Sistem

Berikut hasil implementasi sistem untuk tampilan antarmuka pada aplikasi perpustakaan, meliputi tampilan halaman utama, *form* halaman buku dan *form* halaman laporan.

Sumber: Hasil Penelitian (2019)

Gambar 10. Form Halaman Utama

E-ISSN: 2548-3412 ; 149 – 160

Pada gambar 10 menunjukkan hasil tampilan antarmuka sistem untuk halaman utama. Pada halaman utama terdapat beberapa menu utama diantaranya data buku, pengunjung, anggota, transaksi dan laporan.

Sumber: Hasil Penelitian (2019)

Gambar 11. Form Halaman Buku

Pada gambar 11 menunjukkan tampilan antarmuka untuk *form* halaman buku. Pada halaman buku ditampilkan informasi mengenai data buku yang telah dimasukkan data buku. Adapun informasi yang dapat dilihat pada halaman buku, yaitu judul buku, pengarang, penerbit, tahun terbit, jumlah buku, lokasi dan tanggal *input*.

Sumber: Hasil Penelitian (2019)

Gambar 12. Form Halaman Laporan

Pada gambar 12 menggambarkan hasil implementasi sistem untuk tampilan halaman laporan. Pada halaman laporan terdapat beberapa jenis laporan yang dapat ditampilkan, diantaranya laporan pengunjung, laporan buku, laporan anggota dan laporan transaksi.

Hasil perhitungan pola pengunjung dengan algoritma *Fp-Growth* tersedia seperti gambar 13 dibawah ini.

No	Pola pengunjung Perpustakaan	Nilai %
1	Jika pola mahasiswa mengerjakan skripsi, mencari referensi dan hanya berkunjung	100
2	Jika pola mahasiswa mengembalikan buku,belajar dan diskusi	99
3	Jika pola mahasiswa mengerjakan skripsi, belajar dan diskusi	100
4	Jika pola mahasiswa mengembalikan buku, mengerjakan skripsi, belajar dan kepentingan lain	100
5	Jika pola mahasiswa mengembalikan buku, mengerjakan skripsi, mencari referensi dan hanya berkunjung	100
6	Jika pola mahasiswa mengembalikan buku, mengerjakan skripsi, mencari referensi dan diskusi	100
7	Jika pola mahasiswa mengembalikan buku, mengerjakan skripsi, belajar dan hanya berkunjung	100
8	Jika pola mahasiswa mengembalikan buku, mencari referensi, belajar dan hanya berkunjung	100
9	Jika pola mahasiswa mengembalikan buku, mengerjakan skripsi, mencari referensi, belajar, dan kepentingan lain	100
10	Jika pola mahasiswa mengembalikan buku, mengerjakan skripsi, mengerjakan referensi, belajar dan meminjam buku	100
11	Jika pola mahasiswa mengembalikan buku, mencari buku, mengerjakan skripsi dan diskusi	98
12	Jika pola mahasiswa mengembalikan buku, mencari buku, mencari referensi dan diskusi	98
13	Jika pola mahasiswa mencari buku, mencari referensi, belajar dan hanya berkunjung	99
14	Jika pola mahasiswa mengembalikan buku, mencari referensi, kepentingan lain dan meminjam buku	99
15	Jika pola mahasiswa mencari buku, mencari referensi, kepentingan lain dan meminjam buku	99
16	Jika pola mahasiswa mengerjakan skripsi, mencari referensi, kepentingan lain dan meminjam buku	99

Sumber: Hasil Penelitian (2019)

Gambar 13. Hasil *Fp-growth*

4. Uji Aplikasi

Untuk melihat manfaat aplikasi yang dibangun, dilakukan proses pengujian dengan metode SQA. Hasilnya untuk tiap responden = <78,4> * 0.125 + <80> *0.125 + 79,6> * 0.125 + <80> * 0.125 + <82> * 0.125 + <80> * 0.125 + <80> * 0.125 + <80> * 0.15 = 80.

4. Kesimpulan

Jika pola mahasiswa yang terbentuk menyalin data, meminjam buku dan kepentingan lainnya agar terjadi peningkatan pola maka hal ini ada keterkaitan dengan pola mahasiswa mencari buku (J), mengerjakan skripsi (H), mencari referensi (K), mengembalikan buku (E) dengan range nilai support 0.76 – 0.795. Jika ingin pola mahasiswa yang terbentuk menyalin data, meminjam buku dan kepentingan lainnya agar terjadi peningkatan maka maka hal ini ada keterkaitan dengan pola mengembalikan buku (E), mengerjakan skripsi (H) dan mencari referensi (K). dengan nilai confidence diantara 0.97 – 0,99. Secara kombinasi pola yang terbentuk bahwa kebiasaan mahasiswa jika mengujungi perpustakaan terdiri dari pola E-M, E-H, E-J, J-H. Sehingga hampir dapat dipastikan kombinasi pola tersebut sangat dominan jika dibandingkan dengan pola kombinasi lainnya dengan nilai *lift ratio* diatas 0,99 – 1.03. Aplikasi ini telah di uji ke beberapa *user* dan memperoleh nilai 82,25 sehingga dapat di kategorikan termasuk baik dan dapat di terima oleh user.

E-ISSN: 2548-3412 ; 149 – 160

Referensi

Anggrainingsih R et al. 2017. Discovering Drugs Combination Pattern Using FPGrowth Algorithm", 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI). Yogyakarta, Indonesia. Hal. 702.

Aprilina M N Q. et al. 2016. Analisa Konsistensi Pola Peminjaman Buku Menggunakan Algoritma Fp-Growth. Undergraduate skripsi. Universitas Sebelas Maret.

Ardani N R, Fitrina N. 2016. Sistem Rekomendasi Pemesanan Sparepart dengan Algoritma Fp-Growth (Studi Kasus PT. Rosalia Surakarta). Seminar Nasional Teknologi Informasi dan Multimedia. Yogyakarta. Hal. 97.

Fitria R et al. 2017. Implementasi Algoritma Fp-Growth dalam Penentuan Pola Hubungan Kecelakaan Lalu Lintas. Jurnal Sistem Informasi (Journal of Information Systems). 2(13) : 118-124.

Handayani R.K. 2012. Review of Use of Software Super decisions in the Process of Recurring Decision Making: A Case Study of Determination of High School Teachers. ISSN 2085-725X 2(2).

Herasmus H. 2017. Analisa Customer Service System Menggunakan Metode Data Mining Dengan Algoritma FP-Growth. Jurnal Teknik Ibnu Sina (JT-IBSI): ISSN : 2541-2647. 2 (2): 31.

Joshua J.V et al. 2016. Data Mining: A Book Recommender System using Frequent Pattern Algorithm. Journal of Software Engineering and Simulation. 3(3): 01-13.

K.Dharmarajan, M.A.Dorairangaswamy. 2016. Web Usage Mining: Improve the User Navigation Pattern Using Fp-Growth Algorithm. Elysium Journal,Engineering Research & Management. 3: 32-35.

Larasati D P. et al. 2015. Analisis dan implementasi algoritma Fp- Growth pada aplikasi smart untuk menentukan market basket analysis pada usaha retail (studi kasus : PT.X)", e-Proceeding of Engineering. ISSN : 2355-9365. 2(1): 749.

Witten, Ian H, Frank, E. 2005. Data Mining – Practical Machine Learning Tools and Techniques 2nd Edition, Morgan Kaufmann Publisher.

ISSN
2548-9712

ANALISA DAN PERBANDINGAN METODE ALGORITMA APRIORI DAN FP-GROWTH UNTUK MENCARI POLA DAERAH STRATEGIS PENGENALAN KAMPUS STUDI KASUS DI STKIP ADZKIA PADANG

Domi Sepri[1], M. Afdal[2]

[1]Pascasarjana Universitas Putra Indonesia "YPTK" Padang
[2]Program Studi Sistem Informasi UIN Suska Riau
[1]domisepri@gmail.com,[2]afdal.afatih@gmail.com

Abstrak

Sekolah Tinggi Keguruan dan Ilmu Pendidikan (STKIP) ADZKIA merupakan salah satu institusi pendidikan formal di kota Padang yang disahkan oleh pemerintah. Persaingan di dalam dunia bisnis, khususnya dalam bidang pendidikan membuat pihak perguruan tinggi harus mencari pola sasaran daerah yang strategis dalam pengenalan sekolah. Dengan semakin banyaknya STKIP di Kota Padang, membuat setiap sekolah berusaha mencari calon siswa baru kedaerah-daerah yang potensial. Salah satu cara yang dapat dilakukan untuk penentuan daerah strategis adalah dengan menggunakan teknik *DataMining*. Dari data-data mahasiswa yang ada disekolah dapat diolah mengunakan algoritma *Apriori* dan *FP-Growth* yang menjadi informasi baru untuk dimanfaatkan oleh dalam menentukan daerah yang strategis. Dalam penelitian ini penulis mencoba membandingkan hasil dari algoritma *Apriori* dan *FP-Growth* yang menggunakan data mahasiswa angkatan 2015/2016 dengan nilai *minsupport* = 0.05% dan nilai *minconfidence* = 0.7% telah diperoleh 19 *Association Rule* dan 2 rule tertinggi yang dapat dijadikan sebagai pengetahuan baru serta acuan berharga pada lingkup penelitian ini.

Kata Kunci: *Data Mining,Association Rule, Apriori, FP-Growth.*

1. PENDAHULUAN

1.1 Latar Belakang

Salah satu masalah yang sering dihadapi oleh instansi Perguruan tinggi swasta yang baru berdiri adalah masih belum maksimalnya jumlah mahasiswa yang diharapkan sesuai dari target instansi tersebut. Ini disebabkan oleh kurang maksimalnya strategi promosi dalam memperkenalkan kampusnya ke daerah potensial calon mahasiswa.

Pada Beberapa penelitian tentang *Data Mining*, telah dijelaskan pentingnya data mining dalam dunia pendidikan terutama dalam prediksi-prediksi kemajuan dunia pendidikan. Literatur Internasional terdapat contoh adanya penelitian tentang prediksi siswa yang harus mendapatkan perhatian khusus dari data – data siswa yang putus sekolah (*Baradwaj, 2010).*

Pada penelitian ini penulis mencoba membandingkan algoritma *Apriori* dan *FP-Growth* yang merupakan teknik Assosiation Rules dalam menentukan pola terbaik yang akan dijadikan rujukan dalam medapatkan pola-pola yang tersedia sebagai bahan mempromosikan kampus dikarenakan *FP-Growth* dan *Apriori*merupakan dua algoritma yang sama-sama mempunyai struktur datayang kuat, sehingga menghasilkan peluang *generate rule* yang baik (Yıldız Barış, 2010).

1.2. Perumusan Masalah

Berdasarkan latar belakang di atas, maka dirumuskan permasalahannya sebagai berikut :

1. Bagaimana menerapkan algoritma *Apriori* dan *FP-Growth* di dalam promosi pendidikan sehingga dalam proses tersebut dapat menentukan pola strategis untuk mempromosikan kampus?
2. Bagaimana melakukan pengujian Metode *Apriori* dengan *Tanagra* dan *FP-Growth* dengan *RapidMiner* sehingga menghasilkan penyelesaian masalah dalam mencari strategi promosi yang tepat?
3. Bagaimana perbandingan algoritma *Apriori* dan *FP-Growth* dalam menentukan pola daerah strategis pengenalan kampus ?

ISSN 2548-9712

1.3. Tujuan Penelitian

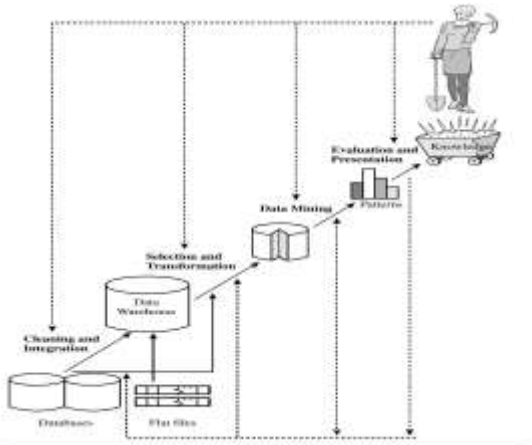
Penelitian ini bertujuan untuk memecahkan permasalahan yang telah dirumuskan di atas yaitu :

1. Untuk mengetahui strategi yang cocok untuk promosi pendidikan di STKIP ADZKIA Padang.
2. Untuk menganalisa keefektifan metode *Apriori* dan *FP- Growth* di dalam strategi promosi pendidikan di STKIP ADZKIA Padang.
3. Untuk melakukan pengujian metode *Apriori* mrnggunakan *Tanagra 1.4.50* dan *FP-Growth* menggunakan *tools Rapid-Miner 7.0.001* sehingga menghasilkan strategi promosi pendidikan yang terbaik.
4. Menghasilkan sebuah hubungan antara variabel atau item berdasarkan implementasi dari algoritma *Apriori & FP-Growth* .
5. Menguji hubungan variable atau item tersebut untuk mendapatkan pengetahuan baru dari proses ekstraksi menggunakan *Tools* yang ada.

2. LANDASAN TEORI

2.1. Knowledge Discovery in Database (KDD)

Menurut Tampubolon, *et al* (2013) *Knowledge Discovery in Database* (KDD) didefinisikan sebagai ekstraksi informasi potensial, implisit dan tidak dikenal dari sekumpulan data. Proses *Knowlegde Discovery in Database* melibatkan hasil proses *Data Mining* (proses pengekstrak kecenderungan suatu pola data), kemudian mengubah hasilnya secara akurat menjadi informasi yang mudah dipahami.



Gambar 2.1. *Data Mining* Sebagai Sebuah Langkah Dalam Proses *Knowledge Discovery*

2.2. Data Mining

Data Mining adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual (Benni R Siburian, 2013).

2.3. Association Rule

Menurut Tampubolon, *et al* (2013) Analisis asosiasi atau *association rule mining* adalah teknik *Data Mining* untuk menemukan aturan assosiatif antara suatu kombinasi item. Algoritma aturan asosiasi akan menggunakan data latihan, sesuai dengan pengertian *Data Mining*, untuk menghasilkan pengetahuan. Pengetahuan untuk mengetahui *item-item* belanja yang sering dibeli secara bersamaan dalam suatu waktu. Aturan asosiasi yang berbentuk "*if…then…*" atau "jika…maka…" merupakan pengetahuan yang dihasilkan dari fungsi aturan asosiasi. Aturan ini dihitung dari data yang sifatnya probabilistik.

Metodologi dasar analisis asosiasi terbagi menjadi dua tahap:

1. Analisa pola frekuensi tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai support dalam *database*. Nilai *support* sebuah *item* diperoleh dengan rumus berikut:

$$Support(A) = \frac{JumlahTransaksiMengandungA}{TotalTransaksi} x\ 100\%$$

2. Pembentukan aturan assosiatif

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan assosiatif yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan assosiatif A _B Nilai *confidence* dari aturan A _B diperoleh dari rumus berikut:

$$Confidence(A \rightarrow B) = \frac{Jumlah\ TransaksiMengandungA\ \&\ B}{Jumlah\ TransaksiMengandung\ A} x100\%$$

2.4. Algoritma Apriori

Apriori adalah suatu algoritma yang sudah sangat dikenal dalam melakukan pencarian frequent itemset dengan menggunakan teknik association rule. Algoritma *Apriori* menggunakan knowledge mengenai frequent itemset yang telah diketahui

ISSN 2548-9712

sebelumnya, untuk memproses informasi selanjutnya. Pada algoritma *Apriori* untuk menentukan kandidat-kandidat yang mungkin muncul dengan cara memperhatikan minimum support (Erwin, 2009).

Proses utama yang dilakukan dalam algoritma Apriori untuk mendapat *frequent itemset* yaitu (Erwin, 2009) :

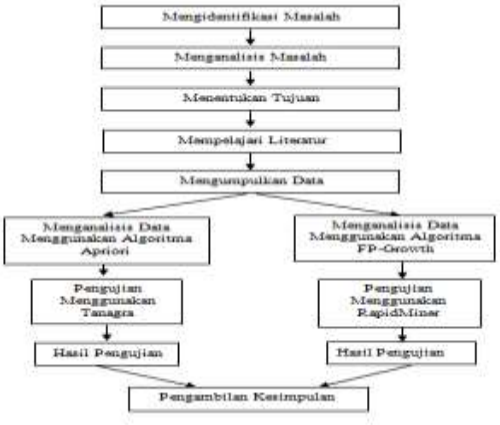
1. *Join* (penggabungan)
   Proses ini dilakukan dengan cara pengkombinasian item dengan item yang lainnya hingga tidak dapat terbentuk kombinasi lagi.
2. *Prune* (pemangkasan)
   Proses pemangkasan yaitu hasil dari item yang telah dikombinasikan kemudian dipangkas dengan menggunakan minimum *support* yang telah ditentukan oleh *user*.

2.5. Algoritma FP-Growth

Menurut Goldie (2012), *FP-growth* adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. *FP-growth* menggunakan pendekatan yang berbeda dari paradigma yang digunakan pada algoritma Apriori.

3. METODOLOGI PENELITIAN

Dalam metodologi penelitian ada urutan kerangka kerja yang harus diikuti, urutan kerangka kerja ini merupakan gambaran dari langkah–langkah yang harus dilalui agar penelitian ini bisa berjalan dengan baik. Kerangka kerja yang harus diikuti bisa dilihat pada gambar 3.1.



Gambar III.1 Kerangka Kerja

Berdasarkan gambar 3.1 dijabarkan urutan–urutan langkah kerja sebagai berikut :

1, Mengidentifikasi Masalah

Masalah yang diidentifikasi dalam penelitian ini untuk mendapatkan strategi daerah untuk promosi yang efektif dan efisien dalam mendapatkan calon mahasiswa baru ditengah ketatnya persaingan antar Perguruan Tinggi.

2. Menganalisis Masalah

Analisis masalah pada penelitian ini dilakukan dengan dua metode yaitu metode deskriptif dan metode komparatif.

a. Metode Deskriptif
   Pada metode ini data yang ada dikumpulkan, disusun, dikelompokkan dan dianalisis sehingga diperoleh beberapa gambaran yang jelas pada masalah yang akan dibahas.
b. Metode Komperatif
   Pada metode ini analisis dilakukan dengan cara membandingkan teori dan praktek sehingga diperoleh gambaran yang jelas tentang persamaan dan perbedaan di antara algoritma Apriori dan FP-Growth.

3. Menentukan Tujuan

Menetukan tujuan akhir dari penelitian ini.

4. Mempelajari Literatur

Literatur-literatur yang dipakai sebagai bahan referensi dalam penelitian ini adalah dari jurnal-jurnal ilmiah, modul pembelajaran dan buku tentang *Data Mining*. literatur-literatur ini akan menjadi pedoman untuk melakukan penelitian agar memudahkan proses penelitian.

5. Pengujian Menggunakan *Tanagra 1.4.50* dan *Rapidminer 7.0.001*

Pada tahap ini, *rule* di tes kembali atau diuji lagi menggunakan sistem *data mining* yang sudah ada. *Tools* yang digunakan sebagai pengujian sistem adalah *Rapidminer* dan Tanagra 1.4.50.

6. Pengambilan Kesimpulan

Setelah diuji, hasil analisis antara cara manual dan pengujian memanfaatkan *tools* akan terlihat perbandingannya. Langkah berikutnya adalah mengambil kesimpulan dari analisa penerapan pada algoritma FP-Growth maupun Apriori terhadap *knowledge* yang baru didapat yaitu berupa penentuan dalam menentukan strategi dalam mempromosikan pendidikan.

ISSN 2548-9712

4. ANALISA DAN PERANCANGAN

Pada penelitian ini proses analisa dibagi dalam 2 jenis, yaitu analisa pada menggunakan algoritma *Apriori* dan analisa data menggunakan algoritma *FP-Growth*. Karena terdapat perbedaan langkah – langkah pengolahan data pada kedua algoritma tersebut sebagai bahan dalam membandingkan kedua algoritma tersebut.

4.1. Pemilihan Variabel

Data sampel yang diuji dalam penelitian ini sebanyak 20 sampel data mahasiswa baru di kampus STKIP ADZKIA TA. 2015 format seperti berikut :

1. Asal Sekolah (SMA Negeri, SMA Swasta, SMK Negeri, SMK Swasta, MA Negeri, MA Swasta)
2. Jurusan di SMA (IPA, IPS, Kejuruan)
3. Alamat Asal (Kabupaten Asal)
4. Sumber Informasi STKIP

Tabel 1 Data Sampel 20 mahasiswa

No	Kabupaten	SMA Asal	Jurusan	Informasi STKIP
1	[illegible]	[illegible]	IPS	Brosur
2	[illegible]	[illegible]	IPA	Teman/ Orang Lain
3	Padang	[illegible]	[illegible]	Teman/ Orang Lain
4	[illegible]	[illegible]	IPS	Brosur
5	[illegible]	[illegible]	[illegible]	Brosur
6	[illegible]	[illegible]	IPS	Teman/ Orang Lain
7	Padang	[illegible]	IPA	Brosur
8	[illegible]	[illegible]	IPS	Brosur
9	[illegible]	[illegible]	IPS	Brosur
10	[illegible]	[illegible]	IPS	Brosur
11	[illegible]	[illegible]	IPA	Teman/ Orang Lain
12	[illegible]	[illegible]	Akuntansi	Teman/ Orang Lain
13	Padang	[illegible]	IPS	Brosur
14	[illegible]	[illegible]	IPS	Teman/ Orang Lain
15	[illegible]	[illegible]	[illegible]	Teman/ Orang Lain
16	[illegible]	[illegible]	IPA	Teman/ Orang Lain
17	[illegible]	[illegible]	IPS	Teman/ Orang Lain
18	[illegible]	[illegible]	IPS	Brosur
19	Padang	[illegible]	Akuntansi	Brosur
20	[illegible]	[illegible]	IPS	Teman/ Orang Lain

4.2. Analisa dengan Algoritma Apriori

Proses berikutnya pada algoritma *Apriori* adalah menseleksi item yang tidak berada di bawah *Support* yang ditentukan. Dalam hal ini *minimal support* yang ditentukan adalah 30% sehingga akan menghasilkan seleksi item

Tabel IV.2 item Minimum Support 30%

Item	Frekwensi	*Support* (Transaksi/ Total Transaksi)	Persen *Support*
PS	14	14/20 = 0.7	70%
SMAN	15	15/20 = 0.75	75%
IPS	11	11/20 = 0.55	55%
BR	11	11/20 = 0.55	55%
TM	9	9/20 = 0.45	45%

Setelah dilakukan kombinasi 2 item, maka akan dihasilkan kombinasi item yang berada di bawah nilai *Support*

Tabel 3 Item Minimum Support 30%

Item	Frekwensi	*Support* (Transaksi/ Total Transaksi)	Persen *Support*
PS, SMAN	12	12/20 = 0.6	60%
PS, IPS	9	15/20 = 0.75	75%
PS, BR	7	11/20 = 0.55	55%
PS, TM	7	11/20 = 0.55	55%
SMAN, IPS	11	9/20 = 0.45	45%
SMAN, BR	8	8/20 = 0.4	40 %
SMAN, TM	7	7/20 = 0.35	35 %
IPS, BR	7	7/20 = 0.35	35%

Setelah itu akan dilakukan kombinasi 3 itemset akan dihasilkan kombinasi di atas *Support* 30% yaitu :

Tabel 4 Item Minimum Support 30%

Item	Frekwensi	*Support* (Transaksi/ Total Transaksi)	Persen *Support*
PS, SMAN, IPS	9	9/20 = 0.45	45%
PS, SMAN, BR	6	6/20 = 0.3	30%
PS, SMAN, TM	6	6/20 = 0.3	30%
PS, IPS, BR	6	6/20 = 0.3	30%
SMAN, IPS, BR	7	7/20 = 0.35	35 %

Setelah itu akan dilakukan kombinasi 4 itemset akan dihasilkan kombinasi di atas *Support* 30% yaitu :

Tabel 5 Item Minimum Support 30%

Item	Frekwensi	*Support* (Transaksi/ Total Transaksi)	Persen *Support*
PS, SMAN, IPS, BR	6	6/20 = 0.3	30%

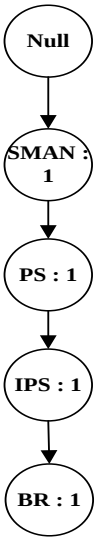
4.3. Analisa Data dengan Algoritma *FP-Growth*

TID 1 : SMAN, PS, IPS, BR

Proses pembentukan *tree* pada tahap *TID 1* adalah dengan membuat cabang *null* yang

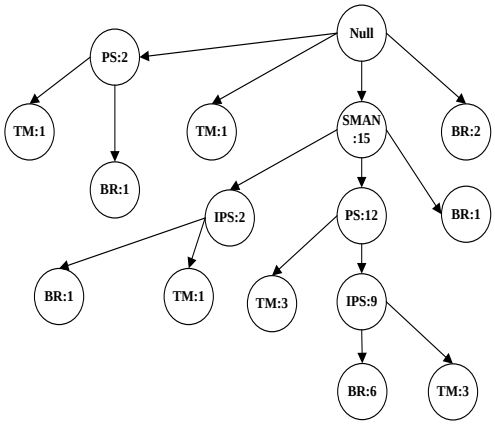
ISSN 2548-9712

merupakan pangkal terbentuknya pohon pada algoritma *FP-Growth*, dari null maka akan dibuat cabang berturut-turut mulai dari SMAN, PS, IPS, hingga BR.



Gambar 4.1 Proses FP-Tree TID 1

TID 20 : SMAN, IPS, TM



Gambar 4.2 Proses FP-Tree TID *20*

Setalah proses dari *FP-Tree* selesai dan dihasilkan *Frequent itemset* sebagai berikut:

Tabel 6 Kombinasi Itemset Algoritma FP-Growth

Suffix	*Frequent Itemset*
TM	{TM, PS}, {TM, SMAN}, {TM, PS, SMAN}
BR	{BR, IPS}, {BR, PS }, {BR, SMAN}, {BR, PS, IPS}, {BR, PS, SMAN}, {BR, IPS, SMAN}, {BR, PS, IPS, SMAN}
IPS	{IPS, PS}, {IPS, SMAN}, {IPS, PS, SMAN}
PS	{PS, SMAN }
SMAN	-

5. HASIL DAN PEMBAHASAN

5.1. Implementasi Sistem

Implementasi yang dilakukan adalah melakukan penerapan semua data mahasiswa baru STKIP ADZKIA 2015/2016 untuk mendapatkan hasil pemetaan pola daerah strategis untuk pengenalan kampus menggunakan algoritma *Apriori* dan *FP-Growth* ke dalam 2 aplikasi data mining, yaitu :

1. Algoritma *Apriori* menggunakan *Tanagra 1.4.50.*
2. Algoritma FP-Growth menggunakan *RapidMiner 7.0.001.*

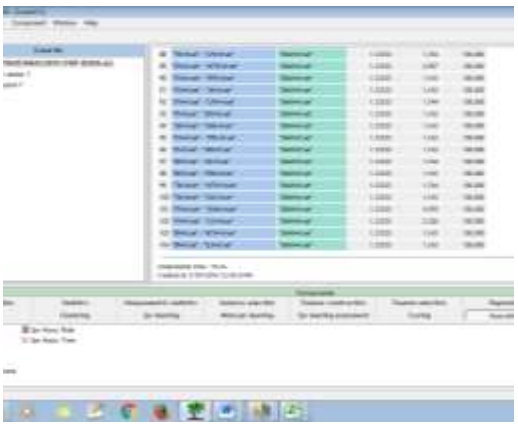
Spesifikasi inputan nilai *Support* dan *Confidence* yang dimasukkan dalam pengujian mengggunakan *Tanagra* dan *RapidMiner* adalah :

1. Nilai Min *Support* adalah : 0.01
2. Nilai Min *Confidence* : 1.0

Nilai ini disesuaikan dengan kebutuhan untuk mendapatkan pola strategis dari masing-masing daerah.

5.2. Implementasi Algoritma *Apriori* Menggunakan *Tanagra 1.4.50*

Setelah data ditransformasi ke dalam bentuk *biner* pada *Microsoft Excel*, selanjutnya adalah melakukan uji data pada aplikasi *Tanagra 1.4.50* , adapun langkah-langkahnya sebagai berikut :
Buka *software Tanagra 1.4.50*, maka akan tampil tampilan awal *software.* Tampilan awal *software* terdiri dari menu file, diagram, window dan help. Tmpilan awal juga berisikan *components* yang merupakan komponen data mining yang bisa digunakan pada *Tanagra*.



Gambar 5.1 Hasil *Rule Algoritma Apriori*

ISSN 2548-9712

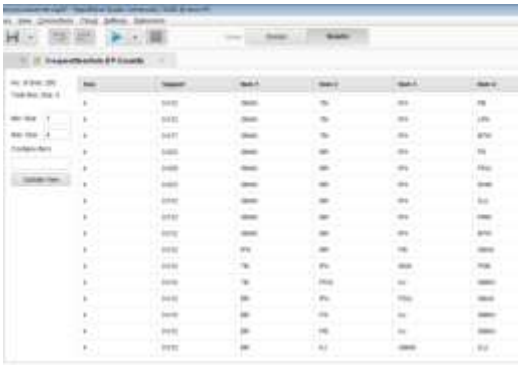
Dari Gambar 5.1. diketahui bahwa jumlah *rule* yang dihasilkan adalah 104 *rule* dan telah memetakan daerah beserta pemetaan sumber informasi pada daerah tersebut.

5.3. Implementasi Algoritma *FP-Growth* Menggunakan *RapidMiner*

Pengujian *algoritma FP-Growth* menggunakan *RapidMiner* dilakukan dengan nilai *Minimum Support* dan *Confidence* yang sama dengan pengujian *algoritma Apriori* menggunakan *Tanagra* yaitu 0.01 untuk *minimum Support* dan 1.0 untuk nilai *minimum Confidence.* Tahapan pertama adalah membuka aplikasi *RapidMiner* yang telah *diinstall*, kemudian melakukan tahapan proses pengambilan data sampai pada akhir proses pelaksanaan seperti pada gambar 5.2.

Gambar 5.2 Menjalankan Proses *Frequent Itemset RapidMiner*

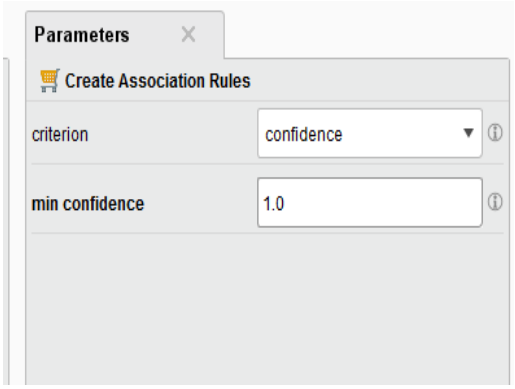
Setelah proses dijalankan , maka akan tampil hasil *frequent itemset* yang diinginkan seperti pada gambar 5.3.



Gambar 5.3 Hasil *Frequent Itemset FP-Growth*

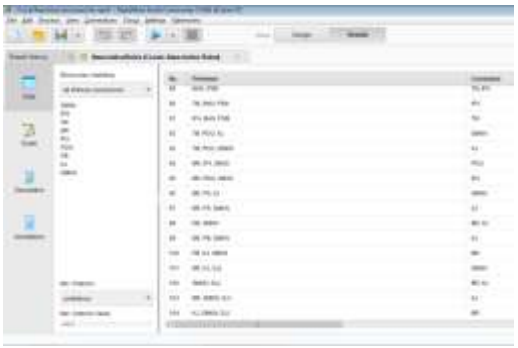
Jika dibandingkan dengan hasil *frequent itemset* yang dihasilkan algoritma *Apriori* menggunakan *Tanagra*, maka akan kita dapati kesamaan hasil *frequent itemset* yang dihasilkan, yaitu sama-sama menghasilkan 51 *itemset*.

Berikutnya adalah memasukkan operator *Create Association Rule* dengan melakukan pencarian pada bagian operator, sehingga di dalam area *process* telah terdapat 4 operator. Kemudian mengatur parameter pada operator *Create Association Rule*. Cara settingnya adalah dengan mengklik *create association rule* dan mengatur *min Confidence* menjadi 1.0 seperti pada gambar 5.4.



Gambar 5.4 Setting Parameter *Create Association Rule*

Setelah proses *setting* dilakukan pada *rapidminer* dan dijalankan maka hasilnya adalah:



Gambar 5.5 Hasil *Rule* Pada *RapidMiner* menggunakan *Algoritma FP-Growth*

Dari pengujian algoritma FP-Growth menggunakan *RapidMiner*, dihasilkan 104 *rule* asosiasi yang sama dengan pengujian algoritma *Apriori* menggunakan *Tanagra*.

ISSN 2548-9712

5.4. Hasil Uji Algoritma *Apriori* dan FP-Growth

Sesuai dengan batasan masalah pada Bab I, maka pengujian yang dilakukan adalah pengujian pada *rule* asosiasi yang dihasilkan dan waktu eksekusi yang diperlukan.

Agar lebih jelasnya hasil perbandingan antara kedua algoritma maka bisa dilihat pada tabel 7

Tabel 7 Hasil Perbandingan algoritma

Indikator Perbandingan	Algoritma *Apriori*	Algoritma FP-Growth
Jumlah *Rule* Asosiasi yang dihasilkan	104 *Rule*	104 *Rule*
Waktu Mendapatkan *Frequent Itemset*	*Error* (karena banyaknya jumlah *itemset* yang harus ditampilkan)	0 detik
Waktu Mendapatkan Rule Asosiasi	0 detik	0 detik

Indikator perbadingannya adalah pada jumlah *rule* asosiasi yang dihasilkan dan eksekusi waktu proses algoritma dalam menghasilkan *frequent itemset* maupun *rule* asosiasi.

5.4.1 Hasil Pengujian *Rule* yang Dihasilkan

Dari Hasil uji kedua algoritma di atas diketahui bahwa hasil perhitungan *rule* yaitu seperti tabel dibawah ini:

Tabel 8 Kesamaan *Rule* yang Dihasilkan kedua Algoritma

NO	IF	THEN	SUPPORT	CONFIDENCE
1	BR, PS, KJ	SMKN	0.0116279	1
2	BR, PS, SMKN	KJ	0.0116279	1
3	IPS, TM, PS	SMAN	0.0581395	1
4	BR, IPA, DHM	SMAN	0.0232558	1
5	IPS, TM, DHM	SMAN	0.0174419	1
6	TM, SLK	SMAN, IPS	0.0116279	1
7	SMAN, TM, SLK	IPS	0.0116279	1
8	IPS, TM, SLK	SMAN	0.0116279	1
9	SMS, JM	SMAN, IPS	0.0116279	1
10	SMAN, SMS, JM	IPS	0.0116279	1
11	IPS, SMS, JM	SMAN	0.0116279	1
12	BR, AG	SMAN, IPS	0.0174419	1
13	SMAN, BR, AG	IPS	0.0174419	1
14	IPS, BR, AG	SMAN	0.0174419	1
15	BR, IPA, SIJ	SMAN	0.0116279	1
16	BR, KJ, SIJ	SMKN	0.0116279	1
17	SMKN, SIJ	BR, KJ	0.0116279	1
18	BR, SMKN, SIJ	KJ	0.0116279	1
19	BR, PRM	SMAN, IPA	0.0116279	1
20	SMAN, BR, PRM	IPA	0.0116279	1
21	BR, IPA, PRM	SMAN	0.0116279	1
22	TM, IPA, LPK	SMAN	0.0116279	1
23	IPS, BR, SMAS	PB	0.0116279	1
24	PB, SMAS	IPS, BR	0.0116279	1
25	IPS, PB, SMAS	BR	0.0116279	1
26	BR, PB, SMAS	IPS	0.0116279	1
27	PB, SMKN	BR, KJ	0.0116279	1
28	BR, PB, SMKN	KJ	0.0116279	1
29	PB, KJ, SMKN	BR	0.0116279	1
30	TM, PDG, KJ	SMKN	0.0116279	1
31	TM, PDG, SMKN	KJ	0.0116279	1
32	BR, IPA, SMAS	PDG	0.0116279	1
33	BR, PDG, SMAS	IPA	0.0116279	1
34	MAN, PSM	TM, IPA	0.0116279	1
35	TM, MAN, PSM	IPA	0.0116279	1
36	IPA, MAN, PSM	TM	0.0116279	1
37	TM, MTW	SMAN, IPA	0.0174419	1
38	SMAN, TM, MTW	IPA	0.0174419	1
39	TM, IPA, MTW	SMAN	0.0174419	1
40	BR, MTW	SMAN, IPA	0.0116279	1

Hasil perhitungan *rule* yang didapat dari penerapan masing-masing algoritma ke dalam aplikasi menemukan kesamaan jumlah *rule* yang didapat yaitu 104 *rule* yang telah menyatakan pemetaan daerah dan pola strategisnya dengan nilai *Confidence* adalah 1.00 (*Strong Confidence*), berikut 42 *rule* kombinasi 4 item yang menyatakan pemetaan pola daerah strategis dari 104 *rule* yang dihasilkan oleh kedua algoritma:

Dari *rule* yang tertera pada tabel 5.3 dapat kita uraikan pola daerah strategis masing-masing daerah adalah:

1. Daerah Pesisir selatan adalah daerah mahasiswa paling banyak berasal dengan jumlah nilai *Support* pada *rule* tertinggi sebesar 0.058 % dan dari 3 *rule* kombinasi yang didapat pada *rule* tertinggi menyatakan bahwa jika mereka berasal dari Jurusan IPS serta mendapatkan informasi dari teman/orang lain maka ia berasal dari SMA Negeri. Disamping itu di daerah Pesisir Selatan yang berasal dari jurusan Kejuruan dan mendapatkan informasi dari brosur maka berasal dari SMK Negeri .
2. Daerah Dhamasraya mempunyai pola daerah strategis adalah jika ia berasal dari jurusan IPA dan mendapatkan info dari brosur, maka ia berasal dari SMA Negeri. Disamping Itu jika ia berasal dari jurusan IPS dan mendapatkan info dari teman maka ia berasal dari SMA Negeri.
3. Daerah solok mempunyai pola daerah strategis adalah jika ia berasal dari jurusan IPS dan berasal dari SMA negeri maka ia mendapatkan info dari SMS atau Teman/Orang lain
4. Daerah jambi mempunyai pola daerah strategis dalah jika ia berasal dari jurusan IPS dan dari SMA Negeri maka ia mendapatkan info dari SMS
5. Daerah Agam mempunyai pola dderah strategis adalah jika ia berasal dari SMA Negeri dan jurusan IPS, maka ia mwendapatkan informasi dari brosur.
6. Daerah sijunjung mempunyai pola daerah strategis adalah jika ia berasal dari SMA Negeri, dan jurusan IPA maka ia mendapatkan info dari Brosur. Disamping itu jika iaberasala dari SMK Negeri dan

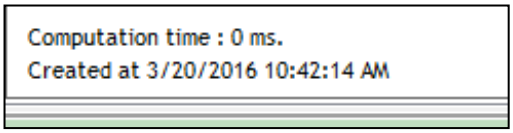
ISSN 2548-9712

jurusan Kejuruan, maka ia mendapatkan informasi dari brosur.

7. Daerah Pariaman mempunyai pola daerah strategis yaitu jika mahasiswa berasal dari SMA Negeri dan dari Jurusan IPA, maka ia mendapatkan informasi dari brosur
8. Daerah Kabupaten Lima Puluh Kota mempunyai pola daerah strategis yaitu jika mahasiswa berasal SMA Negeri dan dari jurusan IPA maka ia mendapatkan informasi dari Teman/Orang Lain.
9. Daerah Pasaman Barat mempunyai pola daerah strategis yaitu jika mahasiswa berasal dari SMA Swasta dan jurusan IPS maka ia mendapatkan informasi dari brosur. Di samping itu jika mahasiswa tersebut berasal dari SMK negeri dan jurusan kejuruan, maka ia mendapatkan informasi juga dari brosur.
10. Daerah Padang mempunyai pola daerah strategis yaitu jika ia berasal dari dari SMK Negeri dan jurusan kejuruan, maka ia mendapatkan informasi dari Orang lain / Teman. Di samping itu jika mahasiswa berasal dari SMA Swasta dan jurusan IPA maka ia mendapatkan informasi dari brosur.
11. Daerah Pasaman mempunyai pola daerah strategis adalah jika mahasiswa berasal dari MA Negeri dan jurusan IPA, maka ia mendapatkan informasi dari Teman/Orang Lain.
12. Daerah Kep.Mentawai mempunyai pola daerah strategis adalah jika ia berasal dari SMA Negeri dan jurusan IPA maka ia mendapatkan informasi dari Teman/Orang Lain atau Brosur.

5.4.2. Hasil Pengujian Waktu Eksekusi

a. Dalam menghasilkan *Frequent Itemset* pada *algoritma Apriori* menggunakan *Tanagra* untuk 4 digit kombinasi menghasilkan pesan error pada *software* dikarenakan banyaknya *frequent itemset* yang harus ditampilkan. Sedangkan pada *Algoritma FP-Growth* menggunakan *software RapidMiner* menghasilkan waktu 0 detik.
b. Dalam pengujian performa waktu dalam mengeksekusi data menggunakan *RapidMiner* maupun *Tanagra* dalam mendapatkan *rule* dari dua algoritma dengan 172 record data menghasilkan kesamaan waktu eksekusi yaitu 0 detik seperti pada gambar V.6 dan V.7 di bawah.



Gambar 5.6 Hasil Perhitungan Waktu Eksekusi *Tanagra*

Hasil pengujian waktu eksekusi pada *software RapidMiner* dengan hasil komputasi 0 detik.

Gambar 5.7 Hasil Perhitungan Waktu Eksekusi *RapidMiner*

Terjadinya kesamaan waktu dalam pengeksekusian ini dikareanakan data yang dieksekusi berjumlah sedikit sehingga tidak bisa melihat perbandingan performa waktu diantara kedua algoritma.

6. KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan hasil analisa dan pengujian yang telah dilakukan, maka dapat disimpulkan beberapa hal yang terkait dengan penelitian ini. Setelah dilakukan pengujian didapatkan beberapa hubungan antara jurusan SMA,kategori SMA, Kabupaten Asal dan SumberI nformasi dalam pemetaan daerah strategis menggunakan algoritma *apriori* dan *FP-Growth*. Adapun kesimpulan yang didapatkan sebagai berikut :

1. Pada penelitian ini dari pengujian dengan nilai minimum support sebesar 0.05 dan minimum confidence 0.7 terdapat 19 *association rule* dengan 8 association rule yang mempunyai kombinasi daerah yaitu hanya pada daerah Pesisir Selatan.
2. Dari pengujian data STMIK Adzkia tahun ajaran 2015/2016, maka diketahui bahwa siswa yang dari daerah kabupaten Pesisir Selatan, berasal dari SMA Negeri, dari jurusan IPS adalah daerah potensial yang mendapatkan informasi dari teman/orang lain merupakan sasaran strategis dalam pengenalan kampus dengan tingkat

ISSN 2548-9712

kepercayaan (*Nilai Confidence)* sebesar 100%.

3. Dalam perbandingan algoritma *apriori* dan *fp-growth*, penulis mendapatkan hasil yang lebih baik pada algoritma *fp-growth* dalam menghasilkan *frequent itemset* karena melakukan pencarian *frequent itemset* menggunakan sistem *FP-Tree* sehingga tidak perlu melakukan *scan database* yang berulang jika sudah ditemukan kombinasi *itemset* yang sudah ditemukan sebelumnya.
4. Waktu yang dihasilkan dengan 171 record data mahasiswa antara algoritma menggunakan *tanagra 1.4* dengan *algoritmafp-growth* menggunakan *rapidminer 7.0.001* menghasilkan waktu proses yang sama yaitu 0 detik.

6.2. Saran

Berdasarkan hasil analisa dan pengujian yang diperoleh, saran dari penulis untuk penelitian selanjutnya adalah :

1. Pada penelitian selanjutnya sebaiknya bisa menambahkan beberapa variabel lagi, sehingga dapat menampilkan lebih banyak lagi hubungan yang terbentuk dari *association rule* yang dihasilkan, sehingga dapat membentuk *knowledge* yang lebih detail dan bermanfaat.
2. Data yang digunakan dalam penelitian perbandingan algoritma *apriori* dan *fp-growth* seharusnya mempunyai nilai record yang besar di atas 3000 data agar lebih menampakkan kelebihan dan kelemahan dari masing-masing algoritma.
3. Pengujian perbandingan antara algoritma *apriori* dan *fp-Growth* sebaiknya dilakukan dalam satu aplikasi yang sama agar lebih memfalidkan hasil perbandingan .

DAFTAR PUSTAKA

[1] Anis Kurniawati., 2014.“Pemetaan Pola Hubungan Program Studi Dengan Algoritma Apriori – Studi Kasus Spmu Unnes.” *Edu Komputika Journal*. 1 (1). 51-58

[2] Azhari., dan Anshori., 2009. ”Pendekatan Aturan Asosiasi Untuk Analisis Pergerakan Saham.” *Seminar Nasional Informatika 2009 (semnasIF 2009) ISSN: 1979-2328 UPN Veteran*. E-183-189.

[3] Benni R Siburian.,2014.“Aplikasi Data Mining Untuk Menampilkan Tingkat Kelulusan Mahasiswa Dengan Algoritma Apriori.” *Pelita Informatika Budi Darma*. VII (2). 56-61.

[4] Brijesh Kumar Baradwaj.,2011. ”Mining Educational Data to Analize Students’performance”. *International Journal of Advanced Computer Science and Applications* vol.2, no.6

[5] Charanjit Bambrah., 2014.”Mining Association Rule in Student Assessment Data”. *International Journal of Advanced Research in Computer and Communication Engineering* ol.3, issue 3,

[6] Efori Buulolo.,2013. ”Implementasi Algoritma Apriori Pada Sistem Persediaan Obat (Studi Kasus : Apotik Rumah Sakit Estomihi Medan)”. *Pelita Informatika Budi Darma*. IV (1). 71-83.

[7] Erwin.,2009.”Analisis Market Basket Dengan Algoritma Apriori dan FP-Growth.” *Jurnal Generic ISSN 1907-4093*

[8] Goldie Gunadi., dan Dana Indra Sensuse.,2012.“ Penerapan Metode *Data Mining Market Basket Analysis* Terhadap Data Penjualan Produk BukuDengan Menggunakan Algoritma Apriori Dan *Frequent Pattern Growth* (Fp-Growth) :Studi Kasus Percetakan PT. Gramedia.” *Jurnal TELEMATIKA MKOM*. 1 (1). 118 132

[9] Kennedi Tampubolon, Hoga, S. dan Bobby, R.,2013. “Implementasi Data Mining Algoritma Apriori Pada Sistem Persediaan Alat-Alat Kesehatan.” *Informasi dan Teknologi Ilmiah (INTI)*. 1 (1). 93-106.

[10] Lokandra C. Reddy, Venkatadri.M 2011. “A Review on Data Mining from Past to the Future”. *International Journal of Computer Applications (0975-8887)*

[11] Ririanti.,2014.”Implementasi Algoritma Fp-Growth Pada Aplikasi Prediksi Persediaan Sepeda Motor (Studi Kasus Pt. Pilar Deli Labumas). *“ Pelita Informatika Budi Darma.* VI (1). 139-144.

Penerapan Algoritma FP-Growth Dalam Penentuan Pola Pembelian Konsumen Pada Kain Tenun Medali Mas

Icca Astrina✉, Muhammad Zainal Arifin, Utomo Pujianto
Jurusan Teknik Elektro, Universitas Negeri Malang
✉icca29astrina@gmail.com

Abstrak: Di kota Kediri terdapat toko kain tenun Medali Mas yang sangat populer. Medali Mas mempunyai aktivitas transaksi penjualan tinggi sehingga menghasilkan tumpukan data besar. Tumpukan data ini diteliti menjadi sebuah informasi pola pembelian konsumen menggunakan teknik data mining association rule dan algoritma FP-Growth. Algoritma FP-Growth menggunakan konsep *tree development* dalam pencarian jenis barang yang sering dibeli (*frequent itemsets*). Data yang digunakan yaitu 26 jenis barang kain tenun dan 200 data transaksi dengan ketentuan 2 atau 3 jenis barang dalam 1 transaksi. Pada penelitian ini ditentukan nilai minimum support sebesar 20% dan nilai minimum confidence sebesar 10%. Pengujian Chi-Square juga digunakan untuk mengetahui seberapa besar korelasi hubungan antar variabel dari hasil *frequent itemsets*. Hasil akhir pola pembelian konsumen yaitu (m→no) apabila membeli kain semi sutra Lusi=abu, Pakan=biru Bunga, maka kemungkinan konsumen akan membeli produk Sarung Lusi=hitam, Pakan=hijau Lurik, Katun Lusi=kuning, Pakan=tosca Bambu dengan hasil korelasi antar variabel sebesar 19,1397274913.

Kata kunci: *Chi-square, FP-Growth, Medali Mas, pola pembelian konsumen, tree development.*

Abstract: *In Kediri City there is a very popular woven fabric shop called Medali Mas. It has high sales transaction activity resulting in a large stack of data purchases. This data stack is examined as an information pattern for consumer purchases using data mining association rule techniques and FP-Growth algorithms. The FP-Growth algorithm uses the concept of development tree in searching for frequent item sets. The data used are, 26 types of woven fabric items and 200 transaction data provided that 2 or 3 types of items in 1 transaction. In this research, the minimum support value of 20% and minimum confidence value of 10% are determined. It also used Chi-Square testing to find out how much correlation between variables from the results of frequent itemsets that have been calculated. The final result of the consumer purchasing pattern is obtained (m→no) when buying Semi sutra Lusi = grey, Pakan = Blue Flowers, then the consumer might buy Sarong Lusi = black, Pakan= green Lurik and Cotton Lusi= yellow, Pakan = Tosca Bamboo with the results of the correlation between variables at 19.1397274913.*

Keywords: *Chi-square, FP-Growth, Medali Mas, consumer buying pattern, tree development.*

I. PENDAHULUAN

Kain Tenun merupakan salah satu warisan yang dimiliki budaya Indonesia. Kain tenun ikat merupakan salah satu jenis kain yang banyak diminati oleh masyarakat Indonesia. Di kota Kediri terdapat produksi kain tenun ikat khas yang berada di desa Bandar Kidul dengan nama toko Medali Mas. Kain tenun ikat Medali Mas diproduksi dengan cara manual, yaitu dengan menggunakan tenaga manusia. Produksi kain tenun ikat oleh Medali Mas dilakukan sesuai pesanan konsumen. Namun demikian, ada juga pembuatan kain tenun ikat siap jadi untuk stok penjualan. Proses penjualan selain dilakukan di toko Medali Mas juga dilakukan melalui pameran agar dapat meningkatkan pembelian.

Medali Mas mempunyai aktivitas transaksi penjualan dan pelayanan konsumen yang setiap hari semakin tinggi sehingga menghasilkan tumpukan data pembelian yang semakin besar. Dengan data mining atau lebih tepatnya *Market Basket Analysis* (MBA), data yang banyak itu dapat diolah sehingga diperoleh informasi tersembunyi yang akan menghasilkan *knowledge* yang berguna untuk pengambilan keputusan oleh produsen. Dalam penelitian ini penulis mengelola tumpukan data pembelian menjadi sebuah informasi penting untuk kelangsungan penjualan pada Medali Mas. Dibutuhkan sebuah teknologi untuk mempercepat pemberian perhitungan analisis apakah produk yang dijual sudah memenuhi keinginan konsumen atau belum, produk apa saja yang dibeli oleh konsumen, dan kebiasaan konsumen membeli produk yang satu dengan yang lainnya. Apabila produk yang tidak sesuai keinginan konsumen masih diproduksi dan dijual, maka akan menyebabkan kerugian pada bisnis tersebut karena tidak laku. Dengan demikian, Medali Mas hanya menyediakan produk yang biasa dibeli oleh konsumen untuk menghindari kerugian. Oleh karena itu, diharapkan adanya sistem penentuan pola pembelian konsumen yang dapat membantu Medali Mas dalam proses produksi dan jual beli.

Proses penggalian data dengan jumlah yang banyak membutuhkan algoritma yang tepat. Algoritma FP-Growth merupakan salah satu algoritma yang dapat digunakan untuk menentukan himpunan data yang sering muncul (*frequent itemset*) dalam suatu dataset [1]. Telah banyak masalah yang penyelesaiannya menggunakan algoritma FP-Growth namun dengan kasus yang berbeda, di antaranya adalah untuk mengatasi masalah pada data penjualan produk buku pada PT. Gramedia [2], MBA pada suatu supermarket [3], menentukan MBA pada usaha *retail*

Copyright © Matrix 2019

[4], analisa *customer service system* [5], sistem rekomendasi buku di perpustakaan [6].

Pada penelitian ini diterapkan algoritma FP-Growth untuk penentuan pola pembelian konsumen pada kain tenun Medali Mas.

II. METODE PENELITIAN

A. *Market Basket Analysis (MBA)*

MBA adalah suatu metodologi untuk melakukan analisis *buying habit* konsumen dengan menemukan asosiasi antar beberapa jenis barang yang berbeda, yang diletakkan konsumen dalam *shopping basket* yang dibeli pada suatu transaksi tertentu. Tujuan dari MBA adalah untuk mengetahui produk-produk mana yang mungkin akan dibeli secara bersamaan. Analisis data transaksi dapat menghasilkan pola pembelian produk yang sering terjadi. Teknik ini telah banyak digunakan oleh toko grosir maupun *retail* [7].

MBA memanfaatkan data transaksi penjualan untuk dianalisis kemudian menemukan pola item-item yang secara bersamaan dalam suatu transaksi. Salah satu manfaat dari MBA adalah untuk merancang strategi penjualan atau pemasaran dengan memanfaatkan data penjualan yang ada di perusahaan [8], yaitu dengan mengubah tata letak toko, menempatkan jenis barang secara berdekatan yang sering dibeli secara bersamaan oleh konsumen dan memberikan diskon kepada barang yang jarang dibeli dan mahal.

B. *Association Rule*

Analisis asosiasi atau *association rule mining* adalah teknik *data mining* untuk menemukan aturan asosiasi antara suatu kombinasi barang. Contoh aturan asosoasi dari analisis pembelian di suatu pasar swalayan adalah dapat diketahuinya berapa besar kemungkinan seorang pelanggan membeli roti bersamaan dengan susu. Dengan pengetahuan tersebut, pemilik pasar swalayan dapat mengatur penempatan barangnya atau merancang kampanye pemasaran dengan memakai kupon diskon untuk kombinasi barang tertentu. Analisis asosiasi menjadi terkenal karena aplikasinya untuk menganalisis isi keranjang belanja di pasar swalayan. Analisis asosiasi juga sering disebut *Market Basket Analysi*s [7]. Menurut [9], ada dua ukuran parameter dalam analisa asosiasi yaitu *Support* dan *Confidence*.

Support adalah pengukuran untuk menunjukan seberapa besar tingkat dominasi suatu item dari keseluruhan transaksi.

$$Support(A) = \frac{Jumla \quad transaksi \ mengandung \ A}{Total \ transaksi} \quad (1)$$

Confidence adalah pengukuran untuk menunjukkan hubungan antar dua item berdasarkan suatu kondisi tertentu.

$$Confidence = \frac{\sum transaksi \ mengandung \ A \ dan \ B}{\sum transaksi \ mengandung \ A} \quad (2)$$

C. *Kain Tenun Medali Mas*

Kota Kediri memiliki ciri khas kain tenun ikat yang dapat dibanggakan. Medali Mas merupakan salah satu dari pengerajin kain tenun ikat yang berada di Jalan KH. Agus Salim No.103 Bandar Kidul kecamatan Mojoroto kota Kediri. Banyaknya pengerajin kain tenun di daerah Bandar Kidul membuat Medali Mas bersaing sehat untuk mendapatkan hati konsumen. Medali Mas melakukan penjualan dengan cara manual, yaitu dengan dilihat dari segi jenis kain, Medali Mas memiliki 5 jenis kain tenun ikat yaitu, sutra, semi sutra, katun, sarung, dan sal. Untuk warna kain, Medali Mas menggunakan 9 warna seperti hijau, kuning, merah muda, abu, merah, biru, putih, hitam, dan oranye. Sedangkan untuk motif pada kain tenunnya terdapat 9 jenis seperti wajik, lurik, tirto, bunga, matahari, ceplok, polos, penari, dan bambu.

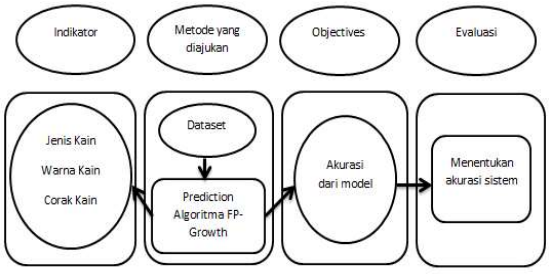
D. *Algoritma FP-Growth*

Frequent Pattern Growth (FP-Growth) adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. Karakteristik algoritma FP-Growth adalah struktur data yang digunakan adalah tree yang disebut dengan FP-Tree. Dengan menggunakan FP-Tree, algoritma FP-Growth dapat langsung mengekstrak frequent Itemset dari FP-Tree. Penggalian itemset yang frequent dengan menggunakan algoritma FP-Growth akan dilakukan dengan cara membangkitkan struktur *data tree* atau disebut dengan FP-Tree [9].

Karakteristik algoritma FP-Growth adalah struktur data yang digunakan adalah *tree* yang disebut dengan FP-Tree. Dengan menggunakan FP-Tree, algoritma FP-Growth dapat langsung mengekstrak *frequent itemset* dari FP-Tree. Penggalian frequent itemset dengan menggunakan algoritma FP-Growth dilakukan dengan cara membangkitkan struktur data *tree* atau disebut dengan FP-Tree. Metode FP-Growth dapat dibagi menjadi 3 tahapan utama [10] yaitu tahap pembangkitan conditional pattern base, tahap pembangkitan *conditional* FP-Tree, dan tahap pencarian *frequent itemset*.

E. *Kerangka Pemikiran*

Adapun kerangka pemikiran yang diajukan adalah sebagai berikut.



Gambar 1. Kerangka pemikiran.

Indikator dalam proses penentuan pola pembelian adalah jenis kain, warna kain, dan motif kain. Sebagian data tersebut akan dijadikan *data training*, sedangkan metode yang akan mengolah dalam penentuan pola pembelian adalah algoritma FP-Growth dari hasil tersebut maka akan digunakan akurasi uji Chi-Square untuk menentukan apakah frekuensi observasi dengan frekuensi harapan saling memengaruhi atau tidak.

III. HASIL DAN PEMBAHASAN

A. *Rancangan Penelitian*

Rancangan penelitian yang digunakan dalam penelitian "Penerapan Algoritma FP-Growth dalam Penentuan Pola Pembelian Konsumen pada Kain Tenun Medali Mas" meliputi studi literatur, pengambilan data, perhitungan dengan metode FP-Growth, pengujian, dan optimasi yang diberikan. Alur penelitian dan cara kerja sistem yang akan dibangun dapat dijelaskan pada Gambar 2.

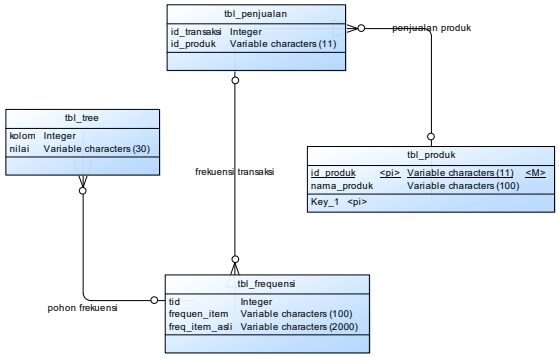
Gambar 2. Diagram alir penelitian.

B. *Pengumpulan Data*

Data yang digunakan dalam penelitian adalah data primer yang diperoleh langsung dari toko kain tenun Medali Mas. Setiap data penjualan menggambarkan pola yang akan dijadikan sebagai masukan yang akan diproses oleh algoritma FP-Growth dan pengujian Chi-Square. Data yang di perlukan adalah Data Barang yang berisi 26 jenis barang dengan *id* dan keterangan jenis kain, warna kain, dan corak kain dalam setiap produk serta Data Transaksi Penjualan berisi produk yang dibeli dalam setiap transaksi. Dengan demikian, hanya membutuhkan jenis produk yang berbeda dengan 2 dan 3 jenis item tanpa melihat jumlah produk yang dibeli dalam setiap transaksi. Dibutuhkan 10 data transaksi untuk pengujian sementara dan 200 data transaksi sebagai data yang diolah sebenarnya.

C. *Relasi Database*

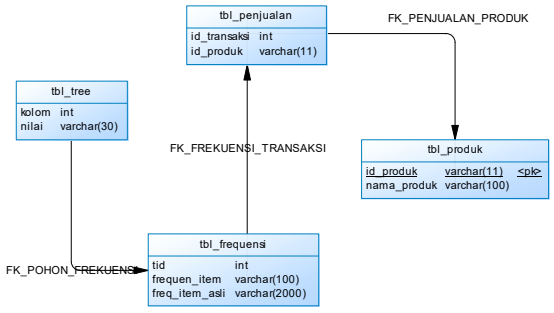
Perancangan *database* merupakan tahapan perancangan selanjutnya yang saling berkaitan. Perancangan *database* akan menjelaskan tentang data apa saja yang harus disimpan yang meliputi tabel dan relasinya. Relasi antar tabel dan jenis relasinya tbl_produk berelasi dengan tbl_penjualan berdasarkan id_produk dengan jenis relasi *one to many*, tbl_penjualan berelasi dengan tbl_frequensi berdasarkan transaksi *id* dengan jenis relasi *one to many* dan tbl_frequensi berelasi dengan tbl_tree berdasarkan transaksi *id* dengan jenis relasi *one to many*. Relasi *database* ditunjukkan pada Gambar 3.



Gambar 3. Relasi *database.*

D. *Physical Data Model (PDM)*

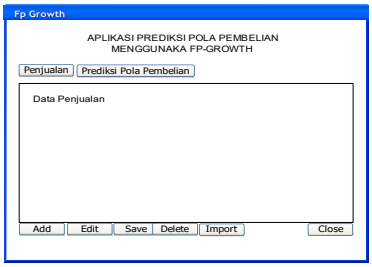
PDM adalah diagram yang akan menjelaskan tentang diagram *database* mengenai *primary key* dan *foreign key*. Pada tabel penjualan, id_produk merupakan *foreign key* dari tbl_produk, sedangkan pada tbl_frequensi, tid merupakan *foreign key* dari tbl_penjualan, dan pada tbl_tree kolom merupakan *forein key* dari tbl_frequensi. PDM ditunjukkan pada Gambar 4.



Gambar 4. *Physical data model.*

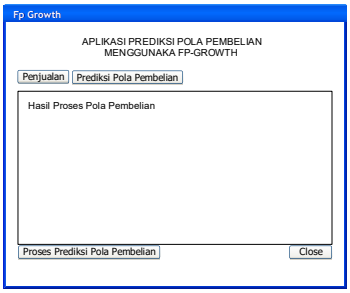
E. *Antar Muka*

Perancangan antar muka pengguna bertujuan untuk memudahkan pengguna menggunakan sistem yang akan dibangun. Pada Gambar 5 di bawah ini merupakan *interface* dari proses penjualan yang akan menampilkan data transaksi penjualan toko Medali Mas.



Gambar 5. Rancangan *interface* proses penjualan.

Pada Gambar 6 ditunjukkan *interface* dari proses pola pembelian yang akan menampilkan hasil prediksi pola pembelian konsumen pada toko Medali Mas dengan menggunakan algoritma FP-Growth.



Gambar 6. Rancangan *interface* proses pola pembelian.

F. Implementasi Sistem

Dalam proses implementasi algoritma FP-Growth terdapat beberapa tahap yang harus dilakukan. Tabel 1 merupakan 10 contoh data produk dan Tabel 2 merupakan 10 data transaksi yang digunakan untuk mencari *frequent itemset*.

Tabel 1. Data produk.

Kode Produk	*Nama Produk*
a	Katun Lusi=oren, Pakan=merah Bambu
b	Katun Lusi=hijau, Pakan=oranye Matahari
c	Semi Sutra Lusi=hitam, Pakan=merah Penari
d	Sutra Lusi=merah, Pakan=putih Polos
e	Sutra Lusi=merah, Pakan=biru Bunga
f	Sutra Lusi=putiih, Pakan=merah Wajik
g	Katun Lusi=merah, Pakan=hijau Ceplok
h	Semi Sutra Lusi=merah, Pakan=hitam Matahari
i	Sal Lusi=biru, Pakan=merah Bunga
j	Sarung Lusi=merah, Pakan=merah Tirto

Tabel 2. Data Transaksi.

TID	*Item*
1	{a,b}
2	{a, b, c}
3	{d, e, j}
4	{b, e, f}
5	{c, d, f}
6	{a, b, g}
7	{b, d, h}
8	{b, c, e}
9	{c, g}
10	{d, c}

Data transaksi yang diambil merupakan 2 dan 3 jenis produk. Setelah data transaksi diketahui selanjutnya menghitung frekuensi kemunculan setiap item dan mencari nilai *support* pada setiap barang. Tabel 3 merupakan tabel frekuensi dan nilai *support*.

Tabel 3. Tabel frekuensi dan nilai s*upport*

Barang	*Frekuensi*	*Nilai Support*
a	3	30%
b	6	60%
c	5	50%
d	4	40%
e	3	30%
f	2	20%
g	2	20%
h	1	10%
i	0	0%
j	1	10%

Setelah perhitungan frekuensi kemunculan setiap *item* diketahui barang yang berada di atas nilai *support count* ξ=20% atau *item* dengan frekuensi di atas 2 adalah a, b, c, d, e, f, dan g. Ketujuh barang inilah yang akan berpengaruh dan akan dimasukkan ke dalam FP-Tree, selebihnya (h, i, dan j) tidak digunakan karena tidak berpengaruh signifikan seperti ditunjukkan pada Tabel 4.

Tabel 4. Transaksi yang disesuaikan dengan *frequent list.*

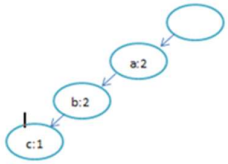
TID	*Barang*
1	{a,b}
2	{a, b, c}
3	{d, e}
4	{b, e, f}
5	{c, d, f}
6	{a, b, g}
7	{b, d}
8	{b, c, e}
9	{c, g}
10	{d, c}

Setelah data transaksi disesuaikan dengan *minimum support,* maka langkah selanjutnya adalah membentuk *tree* yang merupakan langkah selanjutnya dalam proses algoritma FP-Growth. Gambar 7 menunjukkan pembuatan FP-Tree yang diawali dari TID 1 yaitu {a,b}.

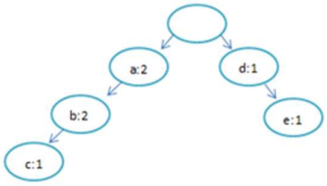
Gambar 7. Pembentukan FP-Tree setelah pembacaan TID 1.

Setelah pembacaan TID 1, selanjutnya dilakukan TID 2 yaitu {a, b, c}. Karena item a dan b ada pada lintasan sebelumnya yaitu TID 1, maka lintasan item {a, b, c} dapat menggunakan lintasan yang sama. Namun untuk lintasan yang dilintasi untuk kedua kalinya, maka akan diberi keterangan nomor seperti a:2 yang artinya *item* a sudah dilewati sebanyak 2 kali. Gambar 8 dan Gambar 9 menunjukkan langkah TID 2 hingga TID 3 dengan

pembentukan lintasan a→b→c yang digabung dengan TID 1.

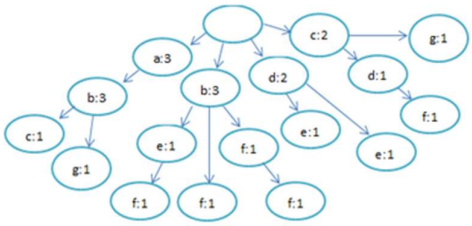


Gambar 8. Pembentukan FP-Tree setelah pembacaan TID 2.



Gambar 9. Pembentukan FP-Tree setelah pembacaan TID 3.

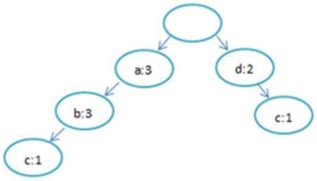
Setelah dilakukan pembacaan TID 1, TID 2, dan TID 3 pembacaan dilakukan dengan cara yang sama sampai dengan pembacaan TID 10 seperti Gambar 10.



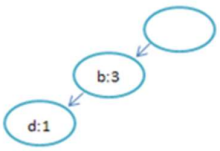
Gambar 10. Pembentukan FP-Tree setelah pembacaan TID 10.

Jika pembacaan dari TID 1 sampai TID 10 sudah dilakukan, maka akan dilanjutkan dengan mencari lintasan yang berakhiran dengan *support count* (b), (c), (d), (e), (f), dan (g). Contoh proses pembentukan masing-masing *node* dapat dilihat pada Gambar 11 sampai Gambar 16.

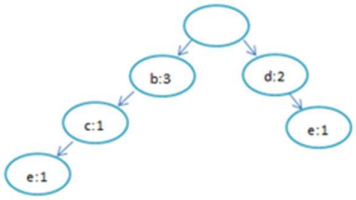
Gambar 11. Lintasan yang mengandung simpul b.



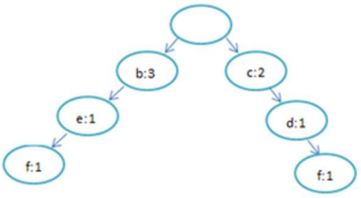
Gambar 12. Lintasan yang mengandung simpul c.



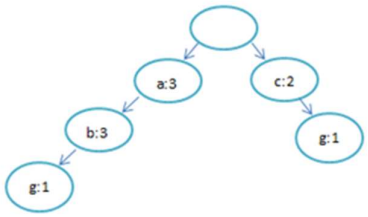
Gambar 13. Lintasan yang mengandung simpul d



Gambar 14. Lintasan yang mengandung simpul e



Gambar 15. Lintasan yang mengandung simpul f.



Gambar 16. Lintasan yang mengandung simpul g.

Setelah memeriksa *frequent itemsets* untuk beberapa akhiran (*suffix*), hasilnya dirangkum pada Tabel 5.

Tabel 5. Hasil *frequent itemset.*

Suffix (Akhiran)	***Frequent Itemset***
b	{a, b}
c	{a, b, c}, {d, c}
d	{b, d}
e	{d, e}, {b, c, e}
f	{b, e, f}, {c, d, f}
g	{a, b, g}, {c, g}

Setelah diketahui hasil dari *frequent itemsets* pada Tabel 5, maka dapat dihitung nilai *confidence* dengan ketentuan minimum 10%. Tabel 6 merupakan 10 perhitungan nilai *confidence* dari sebagian hasil *frequent items*.

Tabel 6. Hasil nilai *confidence* pada *frequent itemset.*

Frequent Itemset	*Nilai Confidence*
a → b	100%
d → c	25%
b → d	16,6%
d → e	25%
c → g	20%
a → bc	30%
b → ac	16,6%
c → ab	20%
b → a	50%
ab → c	30%

Perhitungan nilai *confidence* pada sebagian hasil frequent items diatas menunjukkan sudah memenuhi persyaratan minimum nilai *confidence* yaitu 10%, begitu juga dengan perhitungan yang lainnya. Jadi dapat disimpulkan apabila data sudah memenuhi nilai *minimum support* dan nilai *minimum confidence,* berarti dapat digunakan sebagai referensi dan rekomendasi pembelian pada toko kain Medali Mas. Namun apabila ingin mendapatkan hasil yang lebih akurat, proses pengujian dapat dilakukan dengan menggunakan Chi-square.

Pengujian Chi-Square

Uji Chi-Square adalah pengujian hipotesis mengenai perbandingan antara frekuensi observasi/ yang benar-benar terjadi/ aktual (F_o) dengan frekuensi harapan/ ekspektasi (F_e) yang didasarkan atas hipotesis tertentu. Untuk menguji Chi-square ini digunakan hipotesa sebagai berikut: [9]

H_o: konsumen tidak setuju adanya pengaruh variabel Independen terhadap variabel dependen.
H_a: konsumen setuju adanya pengaruh variabel Independen terhadap variable dependen.

Selanjutnya merupakan hasil frequent item yang ditemulan setelah pengolahan menggunakan algoritma FP-Growth dengan 3 barang {b,e,f} dengan 6 kemungkinan:

b → e,f, jika membeli b maka akan membeli e dan f
e,f → b, jika membeli e dan f maka akan membeli b
e → b,f, jika membeli e maka akan membeli b dan f
b,f → e, jika membeli b dan f maka akan membeli e
f → b,e jika membeli f maka akan membeli b dan e
b,e →f, jika membeli b dan e maka akan membeli f

Tabel 7 merupakan contoh perhitungan dengan menggunakan kemungkinan jika membeli f, maka akan membeli b dan e.

Tabel 7. Data implementasi *frequent item* {b,e,f}.

Variabel 1	Variabel 2		Jumlah
	a	$\bar{a}$	
Bc	1	1	2
$(bc)^{-}$	2	6	8
Jumlah	3	7	10

$$X^2 = \frac{N(ad-b)^2}{(a+c)(b+d)(a+b)(c+d)}$$

$$X^2 = \frac{10(8-0)^2}{(1+1)(0+8)(1+0)(2+8)}$$

$$X^2 = \frac{10 \text{ x } 64}{(2 \text{ x } 8 \text{ x } 1 \text{ x } 10)}$$

$$X^2 = \frac{640}{160}$$

$$X^2 = 4{,}44444444$$

Setelah nilai diketahui, maka tabel Chi-Square pada Tabel 8 selanjutnya dapat digunakan sebagai pengambil keputusan untuk menentukan hasil yang paling signifikan. Pada pengujian kali ini menggunakan nilai signifikan 0,05 (standar signifikan) dengan *degrees of freedom* sebesar 1.

Df = (jumlah kolom – 1) (jumlah baris – 1)
Df = (2 – 1) (2 – 1)
Df = 1

Dengan demikian, diketahui nilai patokan untuk pengujian Chi-Square sebesar 3,84. Apabila nilai kurang dari 3,84, maka H_o diterima atau tidak ada hubungan antar variabel yang diuji. Sebaliknya, apabila nilai lebih besar dari 3,84, maka Ha diterima atau ada hubungan antar variabel yang diuji.

Tabel 8. Uji Chi-Square.

DF	*Probability of a larger value of* X^2					
	0,75	**0,50**	**0,25**	**0,10**	**0,05**	**0,01**
1	0.102	0,455	1,32	2,71	3,84	6,63
2	0,575	1,384	2,77	4,61	5,99	9,21
3	1,212	2,366	4,11	6,25	7,81	11,34
4	1,923	3,357	5,39	7,78	9,49	13,28
5	2,675	43,51	6,63	9,24	11,07	15,09
6	3,455	5,348	7,84	10,64	12,59	16,81
7	4,255	6,346	9,04	12,02	14,07	18,48
8	5,071	7,344	10,22	13,36	15,51	20,09
9	5,899	8,343	11,39	14,68	16,92	21,67
10	6,737	9,342	12,55	15,99	18,31	23,21
11	7,584	10,431	13,70	17,28	19,68	24,72
12	8,438	11,340	14,85	18,55	21,03	26,22
13	9,299	12,340	15,98	19,81	22,36	27,69
14	10,156	13,339	17,12	21,06	23,68	29,14
15	11,037	14,339	18,25	22,31	25,00	30,58
16	11,912	15,338	19,37	23,54	26,30	32,00
17	12,792	16,338	20,49	24,77	27,59	33,41
18	13,675	17,338	21,60	25,99	28,87	34,80
19	14,562	18,338	22,72	27,20	30,14	36,19
20	15,452	19,337	23,83	28,41	31,41	37,57

Dengan menggunakan tabel Chi-Square 2×2 maka memiliki nilai df=1 dan nilai signifikan (α) 0,05 sehingga tabel nilai kritis distribusi Chi-Square sebesar 3,841459. Dengan demikian, variabel akan diterima apabila nilai *X* lebih besar daripada nilai tabel.

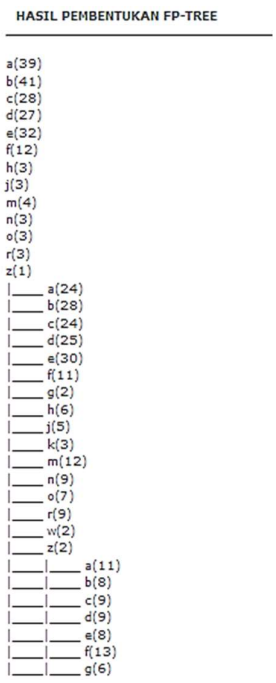
G. Hasil

Data pada penelitian ini menggunakan 200 data transaksi yang ada pada toko kain tenun Medali Mas. Terdapat persyaratan dalam pengambilan data transaksi, yaitu dipilih hanya yang mengandung 2 atau 3 jenis *item* dalam 1 transaksi tanpa harus melihat jumlah dari *item* yang dibeli. Keputusan ini diambil dengan pertimbangan bahwa lebih dari 60 persen transaksi pembelian melibatkan

variasi *item* yang kecil, yaitu satu sampai dengan tiga item berbeda. Dengan kata lain, cukup jarang pembeli yang membeli lebih dari tiga *item* berbeda dalam satu struk pembelian. Argumen lain yang mendasari keputusan ini adalah bahwa hal tersebut akan dapat membantu menyederhanakan pembentukan pola kaidah asosiasi (*association rule*) yang akan dihasilkan oleh sistem sebagai bagian dari rekomendasi untuk pembeli. Medali Mas mempunyai 26 jenis item yang dijual dengan ketentuan warna kain, jenis kain, dan corak kain yang berbeda. Data transaksi tersebut akan diolah dengan menggunakan metode *association rule* dan algoritma FP-Growth untuk mendapatkan informasi tentang *frequent itemset* yang dihasilkan. Kemudian hasil dari *frequent itemset* akan diuji hubungan antar variabel dengan menggunakan Chi-Square.

Implementasi *user interface* terdiri dari beberapa tampilan pada menu yang ada pada aplikasi. Desain dari *user interface* yang baik pada suatu sistem dapat mempermudah user untuk menggunakan sistem tersebut. Berikut *user interface* pada sistem yang sudah dibangun. Terdapat lima menu di dalam sistem Penerapan Algoritma FP-Growth dalam Penentuan Pola Pembelian Konsumen , yaitu menu "Home", " Produk", "Penjualan", "Frekuensi", "Penentuan pola pembelian", dan "About".

Gambar 17 merupakan tampilan dari halaman penentuan pola pembelian bagian pembentukan FP-Tree. Sebuah FP-Tree adalah kompak struktur data yang mewakili *set data* dalam bentuk pohon yang memiliki cabang. Setiap transaksi dibaca kemudian dipetakan ke jalur di FP-Tree. Kegiatan tersebut dilakukan terus menerus sampai semua transaksi dibaca.


HASIL PEMBENTUKAN FP-TREE

a(39)
b(41)
c(28)
d(27)
e(32)
f(12)
h(3)
j(3)
m(4)
n(3)
o(3)
r(3)
z(1)
|___ a(24)
|___ b(28)
|___ c(24)
|___ d(25)
|___ e(30)
|___ f(11)
|___ g(2)
|___ h(6)
|___ j(5)
|___ k(3)
|___ m(12)
|___ n(9)
|___ o(7)
|___ r(9)
|___ w(2)
|___ z(2)
|___|___ a(11)
|___|___ b(8)
|___|___ c(9)
|___|___ d(9)
|___|___ e(8)
|___|___ f(13)
|___|___ g(6)

Gambar 17. *User interface* FP-Tree.

Gambar 18. *User interface* menu utama.

Untuk tampilan dari menu "Home" dapat dilihat seperti pada Gambar 18.

Pada perhitungan Chi-Square digunakan untuk menguji hubungan atau pengaruh dua buah variabel nominal dan mengukur kuatnya hubungan antara variabel yang satu dengan variabel nominal yang lainnya. Perhitungan Chi-Square ditampilkan pada Pengujian Chi-Square (Jumlah Kemunculan) pada halaman Penentuan Pola Pembelian yang ditampilkan dalam bentuk tabel. Setiap tabel mewakili perhitungan Chi-Square dalam 1 *rule* untuk mencari nilai korelasinya berupa "X^2". Untuk 2 *items,* terdapat 2 *rules* yang dimiliki sedangkan untuk 3 *items* memiliki 6 *rules*. Dengan demikian, dapat disimpulkan *frequent itemsets* mana yang memiliki korelasi paling positif. Gambar 19 merupakan tampilan salah satu perhitungan chi-square dengan 2 *item* yang mempunyai 2 *rules* yaitu:

a → m, apabila membeli barang a maka membeli barang m
m →a, apabila membeli barang m maka membeli barang a

Perhitungan Chi-square pada transaksi (a, m) menghasilkan nilai X^2 = 1,09706060691 dengan ketentuan apabila nilai X^2 lebih kecil daripada nilai tabel Chi-Square yaitu 3,841459 maka mempunyai hubungan yang negatif atau tidak ada hubungan antara kedua item begitu pula sebaliknya, apabila nilai X^2 lebih besar dari nilai tabel Chi-Square maka hubungan antara barang adalah positif atau memiliki hubungan.

Frequen Item { a, m, } -->jika membeli a maka akan membeli m

	a	not a	JUMLAH
m	2	17	19
not m	37	143	180
	39	160	199

X=1.09706060691

Gambar 19. *User interface* Chi-Square 2 *item.*

Perhitungan Chi-Square dengan 3 *items* dapat dilihat dalam Gambar 20. Pada contoh merupakan perhitungan 3 *items* yaitu (m, n, o) yang mempunyai 6 *rules*, yaitu:

m → no, apabila membeli barang m maka akan membeli barang no
no → m, apabila membeli barang no maka akan membeli barang m
n → mo, apabila membeli barang n maka akan membeli barang mo
mo → n, apabila membeli barang mo maka akan membeli barang n
o→ mn, apabila membeli barang o maka akan membeli barang mn
mn → o, apabila membeli barang mn maka akan membeli barang 0

Perhitungan Chi-Square dengan *item* (m, n, o) dengan *rule* jika membeli m maka akan membeli no menghasilkan nilai X^2 sebesar 19,1397274913 berarti memiliki nilai positif karena lebih besar dari nilai tabel Chi-Square yaitu 3,841459 yang artinya hubungan antar *item* memiliki korelasi yang sangat tinggi.

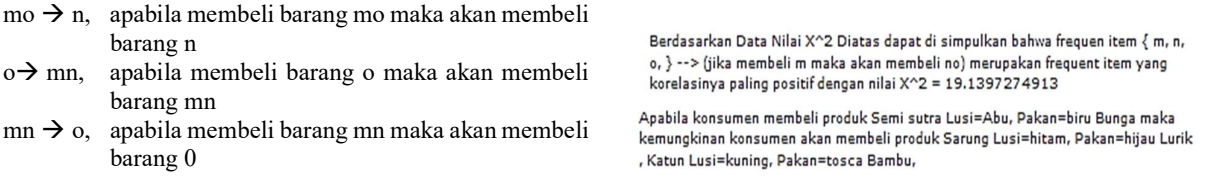
Gambar 20. *User Interface* Chi-Square 3 *items.*

Penentuan nilai Chi-Square yang paling positif atau yang memiliki hubungan antara *item* yang paling bersangkutan dapat dilihat berdasarkan nilai X^2 yang paling kecil. Tabel 9 merupakan 10 nilai X^2 yang paling positif berdasarkan 26 jenis barang dan 200 transaksi yang dilakukan.

Tabel 9. Sepuluh nilai *X* paling positif.

Frequent Item	**Nilai X^2**
{m,n,o} → jika membeli m maka akan membeli no	19,1397274913
{m, n} → jika membeli m maka akan membeli n	19,048313637
{d,o,w} → jika membeli w maka akan membeli do	16,204858541
{b, a, r} → jika membeli r maka akan membeli ba	17,1244863428
{a, k ,j } → jika membeli j maka akan membeli ak	14,37995338
{j, m} → jika membeli j maka akan membeli m	11,5069016788
{d,f,z} → jika membeli z maka akan membeli df	10,9538313609
{c, b} → jika membeli c maka akan membeli b	10,4215091063
{f,d,w} → jika membeli w maka akan membeli fd	9,57468159976
{ b, e } → jika membeli b maka akan membeli e	8,88330980589

Dengan demikian, dapat disimpulkan bahwa data transaksi yang mempunyai hubungan antar *item* paling positif adalah (m,→no) yang mempunyai kemungkinan yaitu "jika membeli m maka akan membeli no" dengan nilai X^2 sebesar 19,1397274913. Gambar 21 merupakan tampilan kesimpulan dari sistem.



Gambar 21. *User interface* kesimpulan.

IV. KESIMPULAN

Hasil eksperimen menunjukan indikasi yang sesuai dengan dugaan awal penelitian bahwa ada pola-pola kaidah asosiasi yang kuat dalam pembelian konsumen kain tenun Medali Mas. Hal ini dibuktikan dengan munculnya sejumlah kaidah asosiasi yang dihasilkan dari implementasi sistem menggunakan algoritma FP-Growth. Tidak hanya itu, hasil uji korelasi dengan *Chi-Square test* juga memperlihatkan bahwa kaidah-kaidah asosiasi dari pola pembelian konsumen pada kain tenun Medali Mas juga memiliki landasan obyektif yang kuat untuk dapat disajikan sebagai rekomendasi pembelian bagi konsumen.

Penelitian ini juga menghasilkan peluang penelitian pengembangan sistem lebih lanjut, misalnya dengan penggunaan algoritma-algoritma klasifikasi dalam ranah *data mining* untuk mengidentifikasi kelas-kelas konsumen dan item pembelian, maupun perluasan jumlah variabel yang digunakan untuk mengidentifikasi pola pembelian yang lebih kompleks.

DAFTAR PUSTAKA

[1] Ikhwan, D. Nofriansyah and Sriani, "Penerapan data mining dengan algoritma Fp-Growth untuk mendukung strategi promosi pendidikan (studi kasus kampus STMIK Triguna Dharma)," *Jurnal Saintikom*, vol. 14, no. 3, pp 211-226, Sep. 2015.

[2] G. Gunadi and D. I. Sensuse, "Penerapan metode data mining market basket analysis terhadap data penjualan produk buku dengan menggunakan algoritma apriori dan frequent pattern growth (FP-Growth): studi kasus percetakan PT.Gramedia," *Jurnal Telematika Mkom*, vol. 4, no.1, pp. 118-132, Mar. 2012.

[3] F. Fathimah, A. Setiawan and R. Rosadi, "Assosiasi Data Mining Menggunakan Algoritma FP-Growth unuk Market Basket Analysis," *academia.edu*. [Online]. Available: https://www.academia.edu/4758451/ASOSIASI_DATA_MINING_MENGGUNAKAN_ALGORITMA_FP-GROWTH_UNTUK_MARKET_BASKET_ANALYSIS. [Accessed: Mar. 2, 2019].

[4] D. P. Larasati, M. Nasrun and U. A. Ahmad, "Analisis dan implementasi algoritma fp-growth pada aplikasi smart untuk menentukan market basket analysis pada usaha retail (studi kasus: PT.X)," in *e-Proc. of Eng.*, 2015, vol. 2, no. 1, pp. 749 – 755.

[5] H. Herasmus. "Analisa customer service system menggunakan metode data mining dengan algoritma fp-growth," *Jurnal Teknik Ibnu Sina*, vol. 2, no. 2, 2017.

[6] R. M. Miraldi, A. Rachmat and B. Santoso, "Implementasi algoritma fp-growth untuk sistem rekomendasi buku di perpustakaan UKDW," *Jurnal Informatika*, vol. 10, no. 1, pp. 29-39, 2014.

[7] Kusrini and E. T. Luthfi, *Algoritma Data Mining*. Yogyakarta, Indonesia: Andi, 2009.

[8] D. Chaerunnissa, "Metode association rule dalam menganalisa pola belanja konsumen pada data transaksi penjualan menggunakan algoritma fp-growth," Bachelor thesis, Universitas Dian Nuswantoro, Semarang, Indonesia, 2015.

[9] Erwin, "Analisis market basket dengan algoritma apriori dan fp-growth," *Jurnal Generic*, vol. 4, no. 2, pp. 26 – 30, July 2009.

[10] A. Rochmah, "Perancangan fitur rekomendasi film di website solo movie dengan menggunakan metode algoritma apriori," Bachelor thesis, Universitas Sebelas Maret Surakarta, Surakarta, Indonesia, 2010.

Implementasi Algoritma FP-GROWTH untuk Sistem Rekomendasi Buku di Perpustakaan UKDW

Rama Novta Miraldi
r.miraldi@gmail.com

Antonius Rachmat
anton@ti.ukdw.ac.id

Budi Susanto
budsus@ti.ukdw.ac.id

Abstract

The number of collections in a library will be increase along with the growth of the library. The Library of Universitas Kristen Duta Wacana (UKDW) also has a large number of books collection in order to provide literature and resources service to students. When the number of collection is very large, we need a retrieval system that will help anyone to find the needed resources. This system has a recommendation feature to provide related books or resources. This feature will help users to find similar literature/resources or related topic with what they need. This paper will describe about the main design and implementation of the book recommendation based on FP-Growth algorithm. Based on the experiments, this system can provide 60.78% of accuracy for related books recommendation

Kata Kunci : *perpustakaan, buku, data mining, fp-growth, fp-tree.*

1. Pendahuluan

Perpustakaan adalah sebuah ruangan, bagian sebuah gedung, ataupun gedung itu sendiri yang digunakan untuk menyimpan buku dan terbitan lainnya menurut tata susunan tertentu untuk digunakan pembaca, bukan untuk dijual (Sulistyo, 1991). Seiring perkembangan zaman koleksi buku suatu perpustakaan bertambah banyak, tidak terkecuali perpustakaan Universitas Kristen Duta Wacana (UKDW). Bertambahnya koleksi buku akan menyulitkan pengunjung perpustakaan untuk mencari buku yang diinginkan. Oleh karena itu dibutuhkan alat bantu berupa sistem pencarian untuk mempermudah pengunjung perpustakaan mencari buku.

Peminjam buku pada perpustakaan membutuhkan alternatif informasi buku lain ketika buku yang hendak dipinjam sedang dipinjam. Referensi peminjam buku dapat menjadi lebih luas jika sistem pencarian ditambahkan fitur rekomendasi buku. Fitur rekomendasi buku ini dapat dilakukan dengan cara mengimplementasikan salah satu algoritma *Association Analysis* yaitu Algoritma *Frequent Pattern Growth* (*FP-Growth*). Algoritma *FP-Growth* ini adalah algoritma yang membutuhkan pelatihan sistem sebelum dapat dipakai untuk memberikan rekomendasi. Dengan menggunakan data transaksi sebagai data untuk pelatihan, sistem dapat menghasilkan rekomendasi.

Oleh karena itu dengan mengimplementasikan algoritma *FP-Growth* untuk fitur rekomendasi pada perpustakaan Universitas Kristen Duta Wacana dapat referensi yang akurat terkait dengan buku yang akan dipinjam. Selain itu juga dapat membantu perpustakaan Universitas Kristen Duta Wacana untuk mengembangkan sistem rekomendasi untuk lebih baik dikemudian hari.

2. Teori Pendukung

2.1 Association Rule

Association Rule merupakan tugas *data mining* yang mendasar (L. Bing, 2007). Menurut "*Data Preparation for Data Mining*" dalam *Applied Artificial Intelligence* (Zhang, Zhang, & Yang, 2003) *Association Rule* merupakan salah satu metode yang bertujuan untuk mencari pola yang sering muncul pada banyak transaksi, dimana setiap transaksi terdiri dari beberapa *item.*

Salah satu pengaplikasian dari *Association Rule* adalah *Market Basket Analysis*, dimana bertujuan untuk menemukan bagaimana *item* yang dibeli oleh pelanggan dalam supermarket atau toko saling berhubungan. Pencarian *Association Rules* dilakukan melalui dua tahap yaitu pencarian *frequent itemset* dan penyusunan *rules.* Penting tidaknya suatu *Association Rules* dapat diketahui dengan dua parameter, yaitu *support* (nilai penunjang) dan *confidence* (nilai kepastian). Support adalah ukuran yang menunjukan tingkat dominasi itemset dari kesuluruhan transaksi. Persamaan 1 untuk menentukan nilai suatu support

$$Support = \frac{(X\ U\ Y)count}{n} \quad [1]$$

dimana (X U Y) count adalah jumlah transaksi yang mengandung x dan y, dan n adalah total transaksi. Parameter lain nya adalah confidence yaitu nilai ukuran seberapa besar valid tidaknya suatu *Association Rules.* Sebuah *Association Rules* dengan *confidence* sama atau lebih besar dari minimum *confidence* dapat dikatakan sebagai *valid association rule.* Persamaan 2 untuk menentukan nilai confidence, dimana X count adalah jumlah kemunculan item X pada seluruh transaksi.

$$Confidence = \frac{(X\ U\ Y).count}{X.count} \quad [2]$$

2.2 Lift

Korelasi pada suatu *rule* tidak hanya diukur pada nilai *support* dan *confidence* tetapi juga oleh korelasi antara kedua *itemsets*. Ada banyak ukuran korelasi yang dapat dipilih, salah satu nya adalah *lift. Lift* adalah salah satu contoh ukuran korelasi sederhana. Nilai *lift* antara A dan B dapat diukur dengan menggunakan komputasi sebagai berikut (Tan, Steinbach, & Kumar, 2004)

$$Lift(A,B) = \frac{P(A \cup B)}{P(A).P(B)} \quad [3]$$

dimana nilai P (A U B) adalah nilai *confidence* dari itemset A dengan B. Dan P(A) adalah jumlah transaksi yang mengandung A. Jika hasil nilai dari perhitungan pada rumus [2.3] kurang dari 1 maka korelasi dari A adalah *negative correlated* dengan itemset B, yang berarti bahwa tidak adanya keterkaitan antar satu sama lain. Dan jika hasil yang didapat adalah lebih dari 1, maka antara A dan B adalah *positively correlated.* Sedangkan bila hasil yang didapat adalah sama dengan 1 maka A dan B adalah *independent.*

2.3 FP-Tree

Frequent Patern Tree (FP-Tree) adalah representasi pemasukan data yang di padatkan (Kumar, 2004). *FP-Tree* dibentuk dengan membaca kumpulan data pada suatu transaksi dalam suatu waktu dan memetakan transaksi tersebut ke dalam lintasan *FP-Tree.* Transaksi yang berbeda dapat memiliki *items* yang sama, sehingga memungkinkan lintasannya saling menimpa. Semakin banyak lintasan yang menimpa satu sama lain, dengan menggunakan struktur *FP-Tree* maka proses pemadatan akan semakin baik.

Dalam "*Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach*" (Han, Pei, Yin, & Mao, 2004) menjelaskan misal I = $\{a_1, a_2, \ldots\}$.

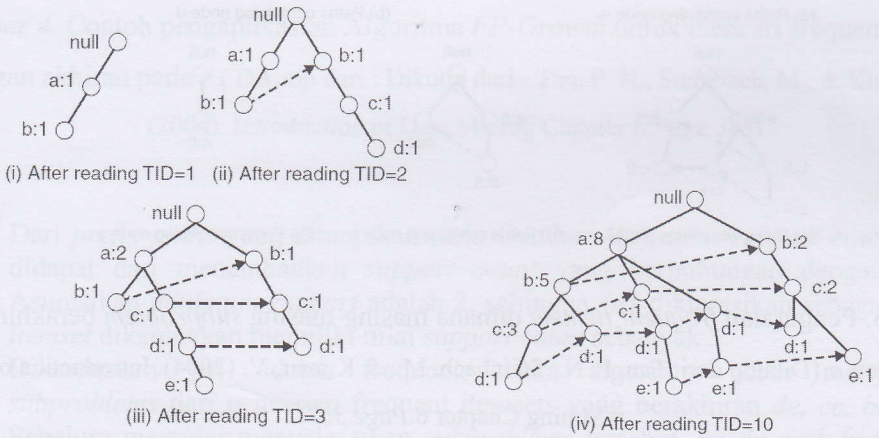
. ., a_n} adalah kumpulan dari item, dan database transaksi DB = {T_1, T_2, ….., T_n}, dimana T_i (i € [1..n]) adalah sekumpulan transaksi yang mengandung item di I. Sedangkan *support* adalah penghitung (*counter*) frekuensi kemunculan transaksi yang mengandung suatu pola. Suatu pola dikatakan sering muncul (*frequent pattern*) apabila *support* dari pola tersebut tidak kurang dari suatu konstanta minimum *support* (batas ambang minimum *support*) yang telah di definisikan sebelumnya. Permasalahan mencari pola frequent dengan batas ambang minimum *support count,* inilah yang dicoba untuk dipecahkan oleh FP-Growth dengan bantuan struktur FP-tree.

Berikut salah satu contoh kasus dengan menggunakan 10 transaksi dengan minimum support count = 2

Tabel 1.
Tabel Data Transaksi Mentah
(Diambil dari : Tan, P. N., Steinbach, M., & Kumar, V. (2004). Introduction of Data Mining Chapter 6.*Page 364)*

TID	Items
1	{a,b}
2	{b,c,d}
3	{a,c,d,e}
4	{a,d,e}
5	{a,b,c}
6	{a,b,c,d}
7	{a}
8	{a,b,c}
9	{a,b,d}
10	{b,c,e}

Gambar di bawah ini memberikan ilustrasi mengenai langkah pembentukan FP-tree.



Gambar 1. Pembentukan *FP-tree*

(Diambil Dari: Tan, P. N., Steinbach, M., & Kumar, V. (2004). Introduction of Data Mining Chapter 6.***Page 364)***

2.4 FP-Growth

FP-Growth adalah algoritma pencarian frequent itemsets yang didapat dari FP-tree dengan menjelajahi *tree* dari bawah menuju ke atas (Tan, Steinbach, & Kumar, 2004) Algoritma *FP*-Growth merupakan pengembangan dari algoritma *Apriori.* Sehingga kekurangan dari algoritma *Apriori* diperbaiki di algoritma *FP-Growth.* Algoritma ini menentukan *frequent itemset* yang berakhiran *suffix* tertentu dengan menggunakan metode *devide and conquer* untuk memecah *problem* menjadi *subproblem* yang lebih kecil (Han, Kamber, & Pei, Data Mining : Concepts and Techniques, 2011). FP-Growth menggunakan

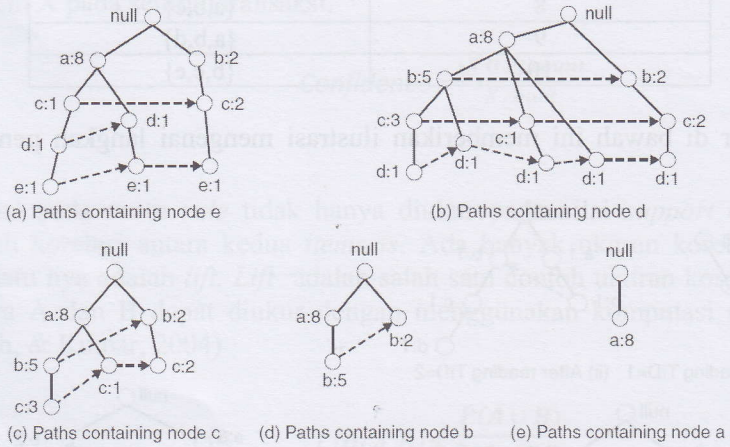
konsep pembangunan *tree* dalam pencarian *frequent itemset*, tidak menggunakan *generate candidate* seperti Algoritma *Apriori*. Hal ini lah yang menyebabkan Algoritma *FP-Growth* lebih cepat dari Algoritma *Apriori*.

Adapun *pseudocode* dari algoritma FP-Growth sebagai berikut

```
procedure FP_growth(Tree, α)
(1)  if Tree contains a single path P then
(2)    for each combination (denoted as β) of the nodes in the path P
(3)      generate pattern β ∪ α with support_count = minimum support count of nodes in β;
(4)  else for each a_i in the header of Tree {
(5)    generate pattern β = a_i ∪ α with support_count = a_i.support_count;
(6)    construct β's conditional pattern base and then β's conditional FP_tree Tree_β;
(7)    if Tree_β ≠ ∅ then
(8)      call FP_growth(Tree_β, β); }
```

Gambar 2. FP-Growth.
(Dikutip dari: Han, J., Kamber, M., Pei, J. (2011). *Data Mining: Concepts and techniques)*

Untuk menemukan *Frequent itemset* dari tabel 1, maka perlu di tentukan terlebih dahulu lintasan yang berakhir dengan *support count* terkecil, yaitu *e* yang di ikuti dengan *d, c, b,* dan di akhiri *a*. Proses pembentukan masing-masing node dapat dilihat pada gambar berikut:



Gambar 3. Penguraian *frequent itemset* dimana masing-masing *subproblem* berakhiran di *e, d, c, b* dan *a*.(Dikutip dari: Tan, P. N., Steinbach, M., & Kumar, V. (2004). Introduction of Data Mining Chapter 6.*Page 367*)

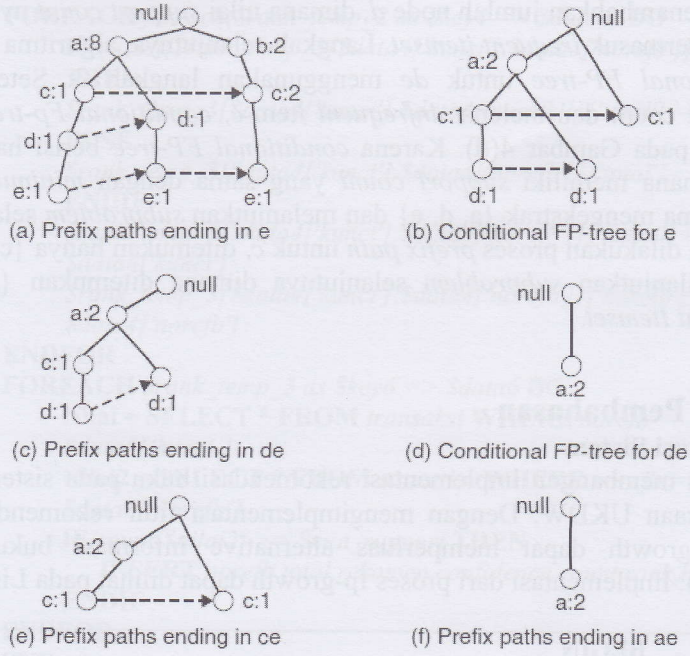
Setelah memeriksa *frequent itemset* untuk beberapa akhiran (*suffix*), maka didapat hasil yang dirangkum dalam tabel berikut:

Tabel 3.
Daftar *Frequent Itemsets* diurutkan berdasarkan hubungan akhiran
(Dikutip dari : Tan, P. N., Steinbach, M., & Kumar, V. (2004). Introduction of Data Mining Chapter 6.*Page 367*)

Suffix	*Frequent Itemset*
e	{e},{d,e},{a,d,e},{c,e},{a,e}
d	{d},{c,d},{b,c,d},{a,c,d},{b,d},{a,b,d},{a,d}
c	{c},{b,c},{a,b,c},{a,c}
b	{b},{a,b}
a	{a}

Berikut contoh yang lebih detail tentang langkah-langkah menemukan frequent itemset, dimana itemsets yang dicari adalah itemsets berakhiran *e,* berdasarkan *Introduction of Data Mining.*

1. Langkah pertama adalah mengumpulkan semua lintasan yang mengandung *node e. Initial* paths ini disebut *Prefix paths*. Berikut gambar dari lintasan yang mengandung *node e* beserta tahapan-tahapan lainnya



Gambar 4. Contoh pengaplikasian Algoritma *FP-Growth* untuk mencari frequent itemset dengan akhiran pada *e.*(Dikutip dari : Dikutip dari : Tan, P. N., Steinbach, M., & Kumar, V. (2004). Introduction of Data Mining Chapter 6.*Page 368)*

2. Dari *prefix paths* yang ditunjukan pada Gambar 4(a), nilai *support count* untuk *e* didapat dari menambahkan *support count* yang berhubungan dengan node *e.* Asumsikan *minimum support* adalah 2, sehingga {*e*} diklarasikan sebagai *frequent itemset* dikarenakan memiliki nilai *support count* sebanyak 3.
3. Dikarenakan {e} adalah frequent, maka algoritma harus menyelesaikan *subproblems* dari pencarian frequent itemsets yang berakhiran *de, ce, be,* dan *ae.* Sebelum memulai menyelesaikan *subproblems* tersebut, *prefix path* harus diubah terlebih dahulu menjadi *conditional FP-tree. Conditional FP-tree* dapat diperoleh dengan langkah berikut :
    a) *Support count* beserta *prefix path* harus diperbaharui terlebih dahulu karena beberapa transaksi tidak mengandung item *e.* Sebagai contoh pada gambar 2.7(a), null → b: 2 → c: 2 → e: 1, pada {b, c} mengandung transaksi yang tidak memiliki *e.*
    b) *Prefix paths* dipotong , dengan cara melepas node *e.* Node ini dapat di lepas karena support counts berhubungan dengan prefix paths telah diperbaharui untuk mencerminkan hanya transaksi yang mengandung *e* dan subproblems dari pencarian *frequent itemsets* yang berakhiran *de, ce, be, dan ae* tidak lagi membutuhkan informasi node *e.*
    c) Setelah pembaharuan *support count* bersama dengan *prefix path,* beberapa item tidak lagi frequent. Sebagai contoh, node *b* hanya muncul sekali dan

memiliki nilai *support count* 1, yang berarti hanya ada 1 transaksi yang mengandung *b* dan *e*. Maka *b* dapat diabaikan dari *subsequent analysis* karene semua itemset yang berakhiran *be* pasti tidak *infrequent*. Gambar 2.7(b) menunjukan Conditional FP-tree untuk *e*, dimana nilai *b* dan *e* telah dieliminasi.

4. FP-Growth menggunakan *conditional FP-tree* dari *e* untuk menyelesaikan subproblem dari pencarian *frequent itemset* yang berakhiran *de, ce,* dan *ae*. Untuk menentukan {d, e} termasuk *frequent itemset* atau tidak, dapat diketahui dengan cara menambahkan jumlah node *d*, dimana nilai *support count* nya sebesar 2. Maka {d, e} termasuk *frequent itemset*. Langkah selanjutnya, algoritma akan membangun *conditional FP-tree* untuk *de* menggunakan langkah 3. Setelah memperbarui support count dan melepas *infrequent* item *c, conditional Fp-tree* untuk *de* dapat dilihat pada Gambar 4(d). Karena *conditional FP-tree* berisi hanya satu item, *a,* yang mana memiliki *support count* yang sama dengan *minimum support*, maka algoritma mengekstrak {a, d, e} dan melanjutkan *subproblem* selanjutnya, yaitu *ce*. Setelah dilakukan proses *prefix path* untuk *c,* ditemukan hanya {c,e} yang *frequent*. Dan dilanjutkan *subproblem* selanjutnya dimana ditemukan {a,e} sebagai sisa *frequent Itemset.*

3. Hasil dan Pembahasan

3.1 Implementasi Sistem

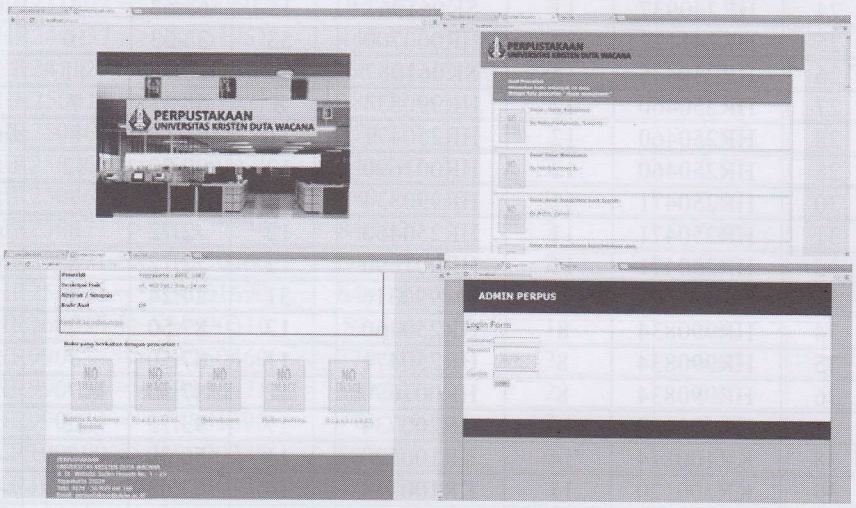
Penulis membangun implementasi rekomendasi buku pada sistem pencarian buku pada perpustakaan UKDW. Dengan mengimplementasi fitur rekomendasi menggunakan algoritma FP-growth dapat memperluas alternative informasi buku terhadap calon peminjam buku. Implementasi dari proses fp-growth dapat dilihat pada List 1.

```
1   BEGIN
2   SET min support
3   Truncate Table Rank
4   utik = SELECT norefb, count (norefb) as total FROM transaksi
5   FOREACH utik as row1 DO
        IF total ≥ min support THEN
            Insert norefb, total to rank
        ENDIF
    ENDFOR
6   query_test = SELECT nim,tgl_pinjam,petugas,norefb FROM `transaksi`
    ORDER BY `transaksi`.`nim` ASC'
7   last_query = query_test->result_array()
8   FOREACH last_query as key => data DO
        transaksi[data['nim'].date('dmy',strtotime(data['tgl_pinjam'])).data['
        petugas']][$i] = $data;
        $i++;
9   Total_trans = count(transaksi)
10  rank = SELECT norefb, total FROM rank
11  Truncate table muncul and rank
12  FOREACH rank as row2 =>data2 DO
        Nampung = SELECT nim,petugas,norefb,tgl_pinjam
        FROM transaksi WHERE norefb = rank → norefb, petugas =
        nampung → petugas, tgl_pinjam = nampung → tgl_pinjam
        Rank_temp[data→norefb] → total = $data → total
        FOREACH nampung as row3 => data2 DO
            Tgl = date('dmy',strtotime($data2->tgl_pinjam))
            petugas = data2→petugas
            IF (!empty($transaksi[$data2->nim.$tgl.$petugas])) THEN
                rank_temp[data→norefb]→hasil_utik[$data2→nim.$tgl.
                $petugas] = $transaksi[$data2->nim.$tgl.$petugas]
                FOREACH $transaksi[$data2→nim.$tgl.$petugas] as
```

```
                    $row3 => $data3) DO
                        IF($data3['norefb'] != $data->norefb) THEN
                            $rank_temp_2[$j]['kunci']=$data→norefb;
                            $rank_temp_2[$j]['norefb']= $data3['norefb'];
                        ENDIF
                    ENDFOR
                ENDIF
            ENDFOR
        ENDFOR
13      FOREACH foreach($rank_temp_2 as $key4 => $data4) DO
            IF (!isset($rank_temp_3[$data4['kunci'].$data4['norefb']]['total'])) 
            THEN
            $rank_temp_3[$data4['kunci'].$data4['norefb']]['total'] = 1
            ELSE
            $rank_temp_3[$data4['kunci'].$data4['norefb']]['total']++
            ENDIF
            $rank_temp_3[$data4['kunci'].$data4['norefb']]['kunci'] =
            $data4['kunci'];
            $rank_temp_3[$data4['kunci'].$data4['norefb']]['norefb'] =
            $data4['norefb'];
        ENDFOR
14      FOREACH $rank_temp_3 as $key6 => $data6 DO
            Nilai = SELECT * FROM transaksi WHERE norefb =
            $data6['kunci']
            Nilai2 = SELECT * FROM transaksi WHERE norefb =
            $data6['norefb']
            IF count($nilai2) >= $min_support THEN
                INSERT norefb,total,rekomen,confidence,count,rank,lift TO rank
            ENDIF
        ENDFOR
15      END
```

List 1. *Pseudo-code* Kalkulasi FP-Growth

Gambar 5 merupakan hasil akhir dari antarmuka implementasi fitur rekomendasi pada sistem pencarian di perpustakaan UKDW. Dimana *form* utama terdiri dari *form* menu pencarian, *form* hasil pencarian, *form* detail buku, dan *form* admin. Pada *form* detail buku bagian bawah terdapat hasil rekomendasi buku yang dihasilkan oleh algoritma fp-growth yang di batasi jumlah buku yang ditampilkan paling banyak 5 buah buku rekomendasi.



Gambar 5. Form Menu pencarian, Hasil Pencarian, Detail buku, dan Admin

3.2 Metode Pengujian

Pengujian program dilakukan untuk mengetahui seberapa kuat rekomendasi yang dihasilkan oleh sistem. Kekuatan suatu rekomendasi diukur menggunakan nilai *lift ratio*. Pengujian dilakukan dengan menghitung nilai lift ratio dari rule yang dihasilkan. Data yang digunakan dalam pengujian ini adalah seluruh buku yang memiliki rekomendasi yang dihasilkan oleh sistem dengan min support = 7. Data transaksi yang digunakan untuk data training sebanyak 80% data transaksi peminjaman buku perpustakaan yang di ambil secara acak. Sedangkan sisa nya, 20% data digunakan untuk pengujian selanjutnya.

Tabel 4.1.

Hasil Lift Ratio

Rule	A	Count (A)	C	Count (C)	Conf (%)	Count (A U C)	Lift Ratio
1	HR000607	33	HR231502	54	48.48	16	94.842
2	HR000700	53	HR241122	30	18.86	10	66.434
3	HR001203	38	HR001208	49	18.42	7	39.711
4	HR001208	49	HR001203	38	14.28	7	39.711
5	HR001690	12	HR990834	8	58.33	7	770.219
6	HR001690	12	HR250460	12	99.99	12	880.250
7	HR001690	12	HR250471	11	66.66	8	640.182
8	HR200734	17	KR100109	16	88.23	15	582.518
9	HR200934	14	PS005859	14	50.00	7	377.250
10	HR200934	14	KR100433	15	50.00	7	352.100
11	HR200934	14	HR230852	14	50.00	7	377.250
12	HR200934	14	PS015247	13	50.00	7	406.269
13	HR201076	8	SK061095	22	87.50	7	420.119
14	HR201761	15	PS010565	22	86.66	13	416.118
15	HR201835	13	PS000717	14	76.92	10	580.385
16	HR201835	13	PS000625	14	76.92	10	580.385
17	HR210671	12	PS007571	10	58.33	7	616.175
18	HR220141	17	HR220142	56	41.17	7	77.669
19	HR220142	56	HR220141	17	12.50	7	77.669
20	HR221056	27	PS009067	17	33.33	9	207.118
21	HR230852	14	HR200934	14	50.00	7	377.250
22	HR230852	14	PS015247	13	50.00	7	406.269
23	HR231502	54	HR000607	33	29.63	16	94.842
24	HR240937	11	SK061961	15	63.63	7	448.127
25	HR241122	30	HR000700	53	33.33	10	66.434
26	HR250425	16	SK061057	19	68.75	11	382.214
27	HR250460	12	HR990834	8	58.33	7	770.219
28	HR250460	12	HR250471	11	66.66	8	640.182
28	HR250460	12	HR001690	12	99.99	12	880.250
30	HR250471	11	HR990834	8	63.63	7	840.239
31	HR250471	11	HR250460	12	72.72	8	640.182
32	HR250471	11	HR001690	12	72.72	8	640.182
33	HR260036	33	SM990316	31	24.24	8	82.604
34	HR990834	8	HR250460	12	87.50	7	770.219
35	HR990834	8	HR250471	11	87.50	7	840.239
36	HR990834	8	HR001690	12	87.50	7	770.219
37	KR100109	16	HR200734	17	93.75	15	582.518
38	KR100114	14	KR100120	11	50.00	7	480.136
39	KR100120	11	KR100114	14	63.63	7	480.136
40	KR100433	15	HR200934	14	46.66	7	352.100
41	KR100433	15	PS005859	14	46.66	7	352.100

Tabel 4.1 (sambungan)

42	KR100433	15	HR230852	14	46.66	7	352.100
43	KR100433	15	PS015247	13	46.66	7	379.185
44	PS000625	14	PS000717	14	71.42	10	538.929
45	PS000625	14	HR201835	13	71.42	10	580.385
46	PS000717	14	HR201835	13	71.42	10	580.385
47	PS000717	14	PS000625	14	71.42	10	538.929
48	PS002891	9	PS003022	12	77.77	7	684.639
49	PS003022	12	PS014922	16	58.33	7	855.815
50	PS003022	12	PS002891	9	58.33	7	481.396
51	PS005854	8	HR210697	10	99.99	9	855.815

No	Rule	Lift	Kemunculan pada data 20%	Support	Memenuhi
1	HR000607 => HR231502	94.842	5	0.18	Ya
2	HR000700 => HR241122	66.434	0	0	Tidak
3	HR001203 => HR001208	39.711	1	0.03	Ya
4	HR001208 => HR001203	39.711	1	0.03	Ya
5	HR001690 => HR990834	770.219	2	0.07	Ya
6	HR001690 => HR250460	880.250	2	0.07	Ya
7	HR001690 => HR250471	640.182	2	0.07	Ya
8	HR200734 => KR100109	582.518	0	0	Tidak
9	HR200934 => PS005859	377.250	0	0	Tidak
10	HR200934 => KR100433	352.100	0	0	Tidak
11	HR200934 => HR230852	377.250	0	0	Tidak
12	HR200934 => PS015247	406.269	0	0	Tidak
13	HR201076 => SK061095	420.119	0	0	Tidak
14	HR201761 => PS010565	416.118	4	0.15	Ya
15	HR201835 => PS000717	580.385	1	0.03	Ya
16	HR201835 => PS000625	580.385	1	0.03	Ya
17	HR210671 => PS007571	616.175	4	0.15	Ya
18	HR220141 => HR220142	77.669	1	0.03	Ya
19	HR220142 => HR220141	77.669	1	0.03	Ya
20	HR221056 => PS009067	207.118	3	0.11	Ya
21	HR230852 => HR200934	377.250	0	0	Tidak
22	HR230852 => PS015247	406.269	0	0	Tidak
23	HR231502 => HR000607	94.842	5	0.18	Ya
24	HR240937 => SK061961	448.127	0	0	Tidak
25	HR241122 => HR000700	66.434	0	0	Tidak
26	HR250425 => SK061057	382.214	3	0.11	Ya
27	HR250460 => HR990834	770.219	2	0.07	Ya
28	HR250460 => HR250471	640.182	2	0.07	Ya
29	HR250460 => HR001690	880.250	2	0.07	Ya
30	HR250471 => HR990834	840.239	2	0.07	Ya
31	HR250471 => HR250460	640.182	2	0.07	Ya
32	HR250471 => HR001690	640.182	2	0.07	Ya
33	HR260036 => SM990316	82.604	1	0.03	Ya
34	HR990834 => HR250460	770.219	2	0.07	Ya
35	HR990834 => HR250471	840.239	2	0.07	Ya
36	HR990834 => HR001690	770.219	2	0.07	Ya
37	KR100109 => HR200734	582.518	0	0	Tidak
38	KR100114 => KR100120	480.136	2	0.07	Ya
39	KR100120 => KR100114	480.136	2	0.07	Ya
40	KR100433 => HR200934	352.100	0	0	Tidak

41	KR100433 => PS005859	352.100	0	0	Tidak
42	KR100433 => HR230852	352.100	0	0	Tidak
43	KR100433 => PS015247	379.185	0	0	Tidak
44	PS000625 => PS000717	538.929	1	0.03	Ya
45	PS000625 => HR201835	580.385	2	0.07	Ya
46	PS000717 => HR201835	580.385	1	0.03	Ya
47	PS000717 => PS000625	538.929	1	0.03	Ya
48	PS002891 => PS003022	684.639	0	0	Tidak
49	PS003022 => PS002891	684.639	0	0	Tidak
50	PS003022 => PS014922	385.109	0	0	Tidak
51	PS005859 => HR200934	377.250	0	0	Tidak

Tabel diatas adalah tabel hasil lift ratio dari rule yang dihasilkan oleh algoritma FP-Growth. Setelah nilai lift dari masing-masing rule telah didapat, di filter terlebih dahulu dengan nilai lift lebih besar daripada 1. Dari 51 rule pada data pada tabel diatas, tidak ada nilai lift yang dibawah dari 1. Hal tersebut dikarenakan perbandingan antara pembagi dan penyebut yang terlampau jauh, dan nilai penyebut yang sangat besar (22570), sehingga mengakibatkan nilai lift melebihi dari 1. Dengan melihat hasil nilai lift ratio diatas dapat disimpulkan, rule yang di hasilkan oleh algoritma *FP-Growth* terhadap 80% data transaksi peminjaman menghasilkan rule yang *positif correlated.* Hal selanjut nya yaitu menguji rule pada tabel diatas dengan 20% data transaksi peminjaman yang berbeda dari data 80% transaksi peminjaman yang digunakan untuk menghasilkan rule. Berikut Tabel pengujian rule terhadap data transaksi sebanyak 2641 data yang sebelumnya tidak dimasukkan ke dalam sistem.

Tabel 4.2.
Tabel Pengujian

Dari tabel diatas dapat dilihat bahwa rule yang memiliki nilai support tertinggi adalah rule yang memiliki jumlah kemunculan pada data 20% sebanyak 5. Sedangkan rule yang memiliki nilai kemunculan terendah adalah 1. Dan terdapat 20 rule yang tidak memenuhi kemunculan pada transaksi 20%. Pengujian dilakukan dengan melihat adakah transaksi pada 20% data yang memenuhi masing – masing rule. Rule yang memenuhi 20% data transaksi sebanyak 31 rule. Dengan diketahuinya banyak nya rule yang memenuhi pada 20% data transaksi yang telah ditentukan, dapat di hitung tingkat keakuratan algoritma, dengan cara menghitung rata-rata keberhasilan.

$$\text{Tingkat keakuratan algoritma} = \frac{Jumlah\ Rule\ yang\ berhasil}{Total\ Rule} \times 100\% \quad [4]$$

Sehingga nilai tingkat keakuratan algoritma sebesar = 0.6078 x 100% = 60.78 %

4. Kesimpulan

1. Algoritma *FP-Growth* dapat digunakan didalam kasus perekomendasian buku dan mendapatkan hasil keluaran berupa rekomendasi buku yang terkait.
2. Tingkat keakuratan FP-Growth dalam memberikan rekomendasi buku adalah sebesar 60, 78%.

DAFTAR PUSTAKA

Alamsyah, N., Khotimah, B, K., & Cahyani, A, D. (2012). *Penetepan Pola Konsumen Menggunakan Algoritma Frequent Pattern Growth Pada Cross Market Analysis.* Madura : Universitas Trunojoyo Madura.

Basuki, S.(1991). *Pengantar Ilmu Perpustakaan.* Jakarta : Gramedia.

Fowler, M. (2000). *UML Distilled (2nd Edition).* Canada : Addison Wesley.

Han, J., Kamber, M., & Pei, J. (2011). *Data Mining : Concepts and techniques (3rd Edition).* Morgan Kaufmann.

Han, J., Pei, J., & Yin, Yiwen. (2004). *Mining Frequent patterns without candidate generation.* Simon Fraser University. Netherlands: Kluwer Academic.

Schneider, G., Winter, J, P. (2001). *Applying Use Case: A Practical Use Case (2nd Edition).* California : Addison-Wesley Profesional.

Tan, Pang-Ning., Steinbach, Michael., & Kumar, Vipin.(2004). *Introduction of Data Mining.* http://www-users.cs.umn.edu/~kumar/dmbook/ch6.pdf(diakses November 2012).

Yongmei, L., Yong ,G. (2009). Application in Market Basket Analysis Based on FP-Growth Algorithm. *Dalam World Congress on Computer Science and Information Enginering.* China: Capital Normal University.

Zhang, S., Zhang, C., & Yang, Q. (2003). *Data Preparation For Data Mining.* Applied Artificial Intelligence , 375-381.

Zhang, W., Liao, Hongzhi., & Zhao, N. (2008). Research on the FP Growth Algorithm about Association Rule Mining. Dalam *International Seminar on Business and Information Management.*

Volume 2, Issue 9, September 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Discovery of Frequent Patterns from Web Log Data by using FP-Growth algorithm for Web Usage Mining

Mr. Rahul Mishra
Computer Science & CSVTU
India

Ms. Abha Choubey
Computer Science & CSVTU
India

Abstract*— Web usage mining refers to the automatic discovery and analysis of patterns in click stream and associated data collected or generated as a result of user interactions with web resources on one or more web sites. It consists of three phases which are data Preprocessing, pattern discovery and pattern analysis. In the pattern discovery phase, frequent pattern discovery algorithms applied on raw data. In the pattern analysis phase interesting knowledge is extracted from frequent patterns and these results are used for website modification. In this paper we are using the FP-growth algorithm for obtaining frequent access patterns from the web log data and providing valuable information about the user's interest.

Keywords*— Web usage Mining, FP-Growth algorithm, Apriori algorithm, Web Server log data, Pattern Discovery.

I. INTRODUCTION

With the growth of explosive Internet information, the data mined from the web are useful information, which has gradually become a very important research .Generally, according to the different subjects, Web mining can be divided into three categories .Web content mining, Web structure mining and Web usage mining. Web log mining is a way of Web usage mining. Web access patterns mined from Web logs are interesting and useful knowledge in practice. Examples of applications of such knowledge include improving designs of web sites, analyzing system performance to understand user's reaction and motivation, build adaptive web sites [1].

Every day, new information, products and services are being offered by providers on the World Wide Web. At the same time, the number of consumers and the diversity of their interests increase. As a result, providers are seeking ways to infer the customers' interests and to adapt their web sites to make the content of interest more easily accessible. Frequent Pattern mining is a promising approach in support of this goal. Assuming that past navigation behaviour is an indicator of the users' interests, and then the records of this behaviour, kept in the form of the web-server logs, can be mined to infer what the users are interested in. On that basis, recommendations can be dynamically generated, to help new web-site visitors find the information of interest faster.

A user session is all of the page references made by a user during a single visit to a site. A transaction differs from a user session in that the size of a transaction can range from a single page reference to the entire page references in a user session, depending on the criteria used to identify transactions. Once user transactions or sessions have been identified, there are several kinds of access pattern mining that can be performed depending on the needs of the analyst, such as path analysis, discovery of association rules and sequential patterns and clustering, and classification.

II. RELATED WORK

Huiping Peng in 2010 stated the use of FP-growth algorithm for processing the web log records, obtaining a set of frequent access patterns, then using the combination of browse interestingness and site topology interestingness of association rules for web mining [1]. Analysis of web usage mining by using Web Log analyzer tool, "Web Log Expert" was carried out by Sanjay Kumar Malik et al. in 2010. They focused on the development of Ontology for an intelligent or efficient web and it's relation with web usage mining. Finally, they also summarize some other research challenges towards an intelligent machine and web environment [2]. Hao Yan et al. in 2010 proposed a two-step K-means clustering algorithm to search user groups in realistic data collected from WAN. They gave some useful practical conclusions to facilitate design of targeting and recommending applications [3].Jiawei Han et al. in 2004 proposed a novel frequent-pattern tree structure, which is an extended prefix-tree structure for storing compressed, crucial information about frequent patterns, and develop an efficient FP-tree based mining method, FP-growth, for mining the complete set of frequent patterns by pattern fragment growth [4]. Rakesh Agrawal and Ramakrishan Srikant in 1994 consider the problem of discovering association rules between items in a large database of sales transactions. They present two new algorithms for solving this problem that are fundamentally different from the known algorithms. They also show how the best features of the two proposed algorithms can be combined into a hybrid algorithm, called AprioriHybrid [5].Mohd helmy Abd Wahab et al. in 2008 describes the pre-processing techniques on IIS Web Server Logs ranging from the raw log file until before mining process can be performed [6]. C.P Sumathi et al. in 2011 presented an overview of the

© 2012, IJARCSSE All Rights Reserved

various steps involved in the preprocessing stage [7].Renata Ivancsy et al. in 2006 investigated three pattern mining approaches from the web usage mining point of view [8].Vaibhav Kant singh et al. in 2008 shows how the different approaches achieve the objective of frequent mining . They also look for hardware approach of cache coherence to improve efficiency of the above process [9]. Dr. R.Krishnamoorthi and K.R Suneetha in 2009 has done the in-depth analysis of Web Log Data of NASA website to find information about a web site, top errors, potential visitors of the site etc. which help system administrator and Web designer to improve their system by determining occurred systems errors, corrupted and broken links by using web using mining[10].

III. WEB USAGE MINING

Web Usage Mining can be used to make search relevant by determining frequent access behavior for users, needed links can be identified to improve the overall performance of future accesses. Web Usage mining has been defined as the application of data mining techniques to discover usage patterns from Web data in order to understand and better serve the needs of Web based applications. Web usage mining consists of three phases, namely preprocessing, pattern discovery, and pattern analysis. Web Usage Mining may be applied to data such as contained in logs files. A log file contains information related to the user queries on a website. Web usage mining may be used to improve the website structure or giving recommendations to visitors [2].

The aim in web usage mining is to discover and retrieve useful and interesting patterns from a large dataset. In web mining, this dataset is the huge web data. Web data contains different kinds of information, including, web structure data, web log data, and user profiles data. Web mining is the application of data mining techniques to extract knowledge from web data, where at least one of structure or usage data is used in the mining process. Web usage mining has various application areas such as web pre-fetching, link prediction, site reorganization and web personalization. Most important phases of web usage mining is discovering useful patterns from web log data by using pattern discovery techniques such as Apriori, FP-Growth algorithm[4].

IV. WEB USAGE MINING PROCESS

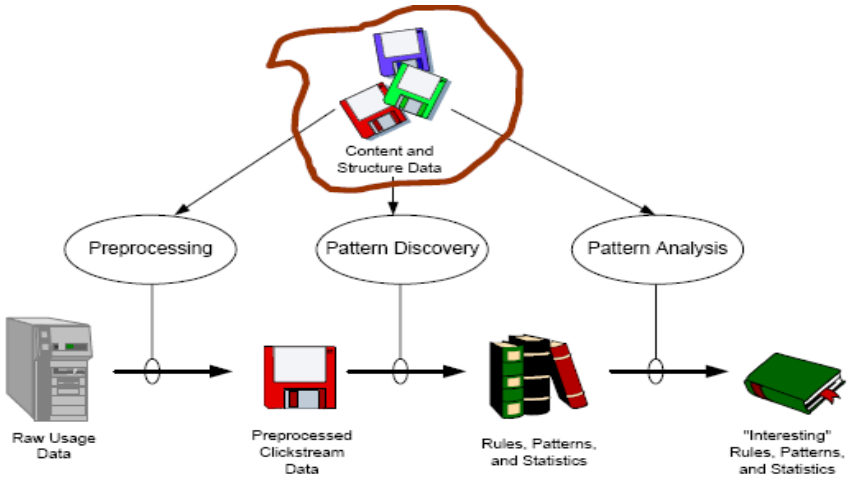


Fig.1 Web Usage Mining Process

A. Web Server Log Data

The web plays an important role and medium for extracting useful information. There is a need for data log to track any transaction of the communications.Log file data can offer valuable information insight into web site usage. It characterizes the activity of many users over a potentially long period of time.Server logs can be used to provide some technical information regarding server load, successful requests, as well as assisting in marketing and site development and management activities. Below is an example of common transfer log collected. This data is collected from NASA web server log.

EXAMPLE: 199.72.81.55 - [01/Jul/1995:00:00:01 -0400] "GET /history/apollo/ HTTP/1.0" 200 6245
unicomp6.unicomp.net - [01/Jul/1995:00:00:06 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985
199.120.110.21 - [01/Jul/1995:00:00:09 -0400] "GET /shuttle/missions/sts-73/mission-sts-73.html HTTP/1.0" 200 4085
burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 304 0
burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/video/livevideo.gif HTTP/1.0"2000
205.212.115.106 - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/countdown.html HTTP/1.0" 200 3985

© 2012, IJARCSSE All Rights Reserved

The server log consists of several attributes. The attributes are as follows:-

1) *Date:* The date from Greenwich Mean Time (GMT x 100) is recorded for each hit. The date format is YYYY-MM-DD. The example above shows that the transaction was recorded at 1995-07-01.

2) *Time:* Time of transactions. The time format is HH:MM: SS. The example from above shows that the transaction time was recorded at 00:00:01.

3) Client IP Address: Client IP is the number of computer who access or request the site.

4) *User Authentication:* Some web sites are set up with a security feature that requires a user to enter username and password. Once a user logs on to a Website, that user's "username" is logged in the log file.

5) *Server IP Address:* Server IP is a static IP provided by Internet Service Provider. This IP will be a reference for access the information from the server.

6) *Server Port:* Server Port is a port used for data transmission. Usually, the port used is port 80.

7) *Server Method (HTTP Request):* The word request refers to an image, movie, sound, pdf, .txt, HTML file and more. The above example indicates that folder.gif was the item accessed.

8) *URI:* URI is path from the host. It represents the structure of the websites. For examples:/tutor/images/icons/fold.gif.

9) *Agent Log:* The Agent Log provides data on a user's browser, browser version, and operating system. This is the significant information, as the type of browser and operating system determines what a user is able to access on a site.

B. Pattern Discovery and Pattern analysis

The three main stages of web usage mining are data preprocessing, pattern discovery and pattern analysis. Data preprocessing involves removal of unnecessary data. Pattern discovery data mining techniques are used in order to extract patterns of usage from Web data. The knowledge that can be discovered is represented in the form of rules, tables, charts, graphs, and other visual presentation forms for characterizing, comparing, predicting, or classifying data from the web access log. Pattern Analysis is the final stage of the Web usage mining. The aim of this process is to extract the interesting rules or patterns from the output of the pattern discovery process by eliminating the irrelative rules or patterns [10].

V. PROPOSED ALGORITHM

We are proposing FP-Growth algorithm for web usage mining, since no real time server available so we tested our algorithm on available log files on HTTP requests to the NASA Kennedy Space Center WWW server in Florida. The log was collected from 00:00:00 July 1, 1995 through 23:59:59 July 31, 1995, a total of 31 days. Now to extract the information such as requested files and most frequently accessed files, first we need to analyze the log file, below some entries of log file names are show:

TABLE I
INDEXING DONE TO REPRESENT STRINGS INTO SPECIFIC NUMBERS

INDEX ID	FILE NAMES
1.	shuttle/countdown/index.html
2.	KSC.html
3.	shuttle/missions/missions.html
4.	shuttle/missions/sts-71/images/images.html
5.	shuttle/missions/sts-71/movies/movies.html
6.	shuttle/countdown/liftoff.html
7.	shuttle/missions/sts-71/mission-sts-71.html
8.	shuttle/missions/sts-70/mission-sts-70.html
9.	shuttle/countdown/countdown.html
10.	history/Apollo/Apollo.html

© 2012, IJARCSSE All Rights Reserved

TABLE I shows indexing operation applied for the purpose of data preprocessing. Here the strings are represented by unique index id. it shows the indexing for the requested files. We have done the same indexing operation for the gif files available in the web server log data and then applying the FP-growth algorithm to obtain various results such as most frequently visited pages, Top downloaded Pages from the web site, Top downloaded gif files and most frequently downloaded gif files from the web server log data.

A. Data Preprocessing

This operation is defined as filtering or pre-processing of data. Since the mathematical operations cannot be performed on strings therefore the strings are represented by specific numbers which is called Indexing. Hence we consider each file name by a unique index id. Next we apply the frequent pattern FP-Growth algorithm on the log files.

B. The analysis of frequent patterns from the web log data

After data preprocessing, we apply the following conditions. The following is a formal statement of the problem: Let L= {i1, i2…im} be a set of literals, called items. Let D be a set of transactions, where each transaction T is a set of items such that T is a subset of L. Associated with each transaction, we say that a transaction T contains X, a set of some items in L. For example, if various users repeatedly access the same series of pages, a corresponding series of log entries will appear in the log file, and this series can be considered as an access pattern.

We have studied the performance of the FP-growth method in comparison with the basic apriori algorithms in large databases. Then we have find out frequent patterns from web log data by using the FP-growth method, our performance study shows that the method mines both short and long patterns efficiently in large databases, The FP-growth algorithm is one of the fastest approaches for frequent item set mining. The FP-growth algorithm uses the FP-tree data structure to achieve a condensed representation of the database transaction and employees a divide-and conquer approach to decompose the mining problem into a set of smaller problems.

VI. COMPARISON BETWEEN FP-GROWTH ALGORITHM AND APRIORI ALGORITHM

Apriori algorithm searches for large itemsets during its initial database pass and use its result as the seed for discovering other large datasets during subsequent passes. Rules having a support level above the minimum are called large or frequent itemsets and those below are called small itemsets. The algorithm is based on the large itemset property which states: Any subset of a large itemset is large and any subset of frequent item set must be frequent.

The FP-growth method is efficient and scalable for mining both long and short frequent patterns and is about an order of magnitude faster than the Apriori algorithm and also faster than some recently reported new frequent-pattern mining methods.

The Apriori heuristic achieves good performance gained by (possibly significantly) reducing the size of candidate sets. However, in situations with a large number of frequent patterns, long patterns, an Apriori algorithm suffer [4].

A. Execution Time Comparisons

The execution time comparison experiment is performed on datasets with 80K, 50K and 30K records; In these graphs the response times of both the algorithms increases as the support threshold is reduced.

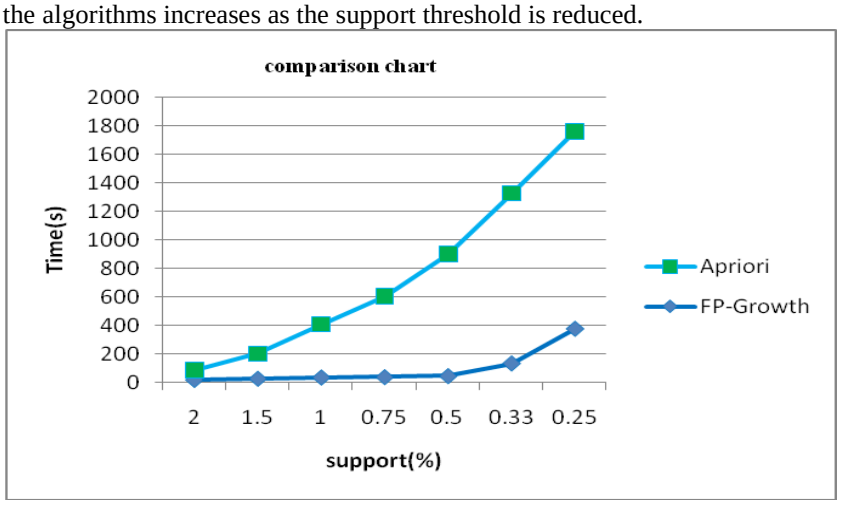


Fig.2 The above graph shows the comparison done using 80K Database

© 2012, IJARCSSE All Rights Reserved

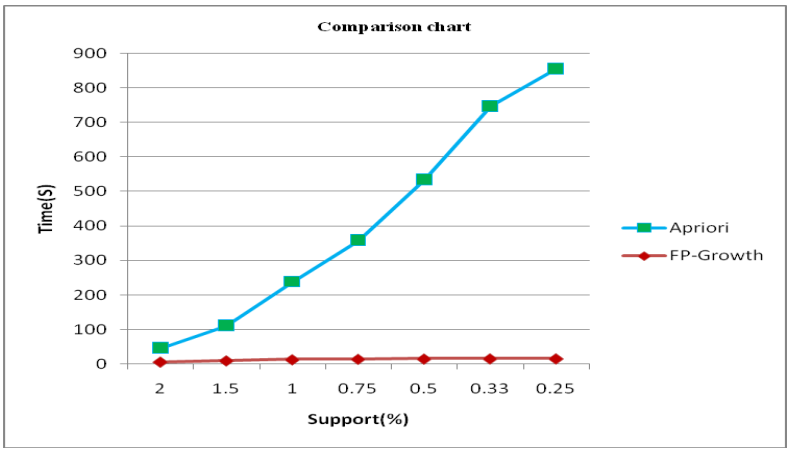


Fig.3 The above graph shows the comparison done using 50K Database

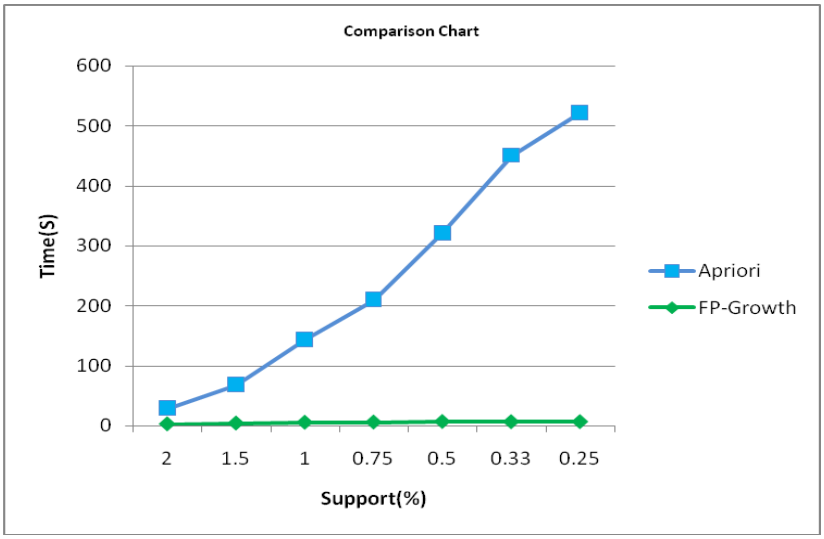


Fig.4 The above graph shows the comparison done using 30K Database

The above result shows that the FP-Growth algorithm is much more efficient than the basic apriori algorithm. Since the FP-Growth algorithm is very efficient therefore we have implemented the FP-growth algorithm for the purpose of Web usage mining.

VII. EXPERIMENTAL RESULTS

The simulation Results for the proposed algorithm is shown below, the training is performed by using first 1000 entries from the log files of NASA Kennedy Space Center WWW server and following results are drawn.

© 2012, IJARCSSE All Rights Reserved

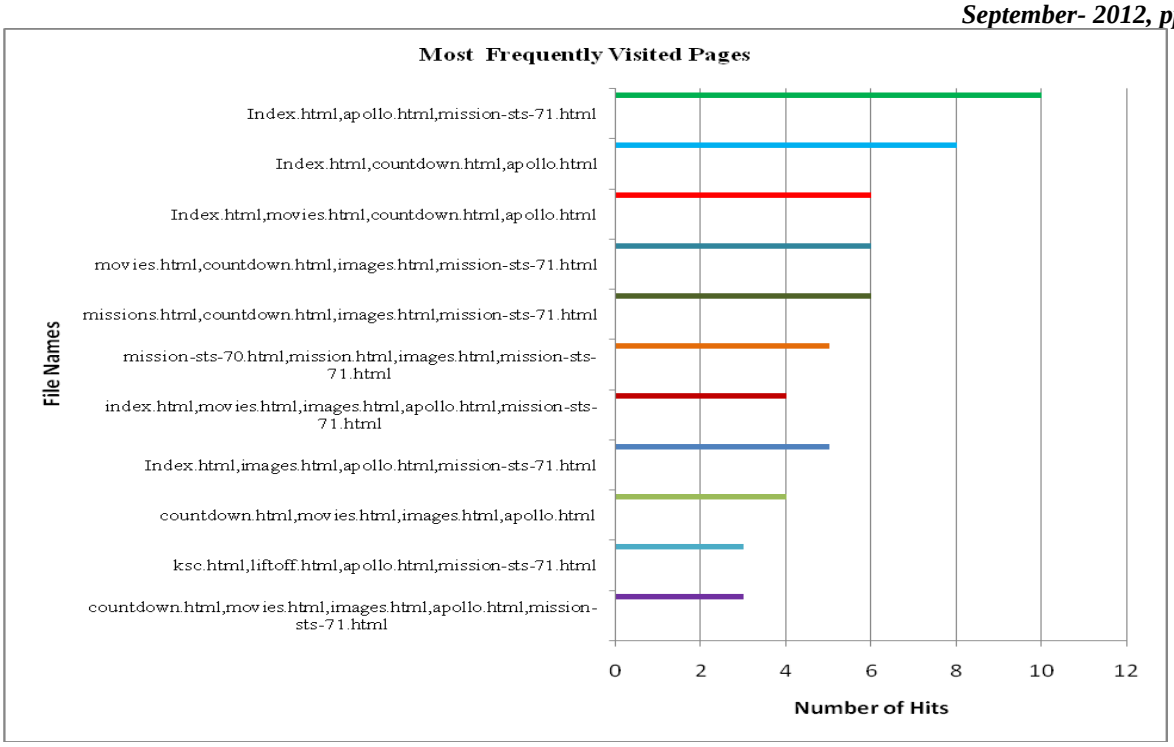


Fig.5 The above graph shows the result for the Most frequently visited pages from the website.

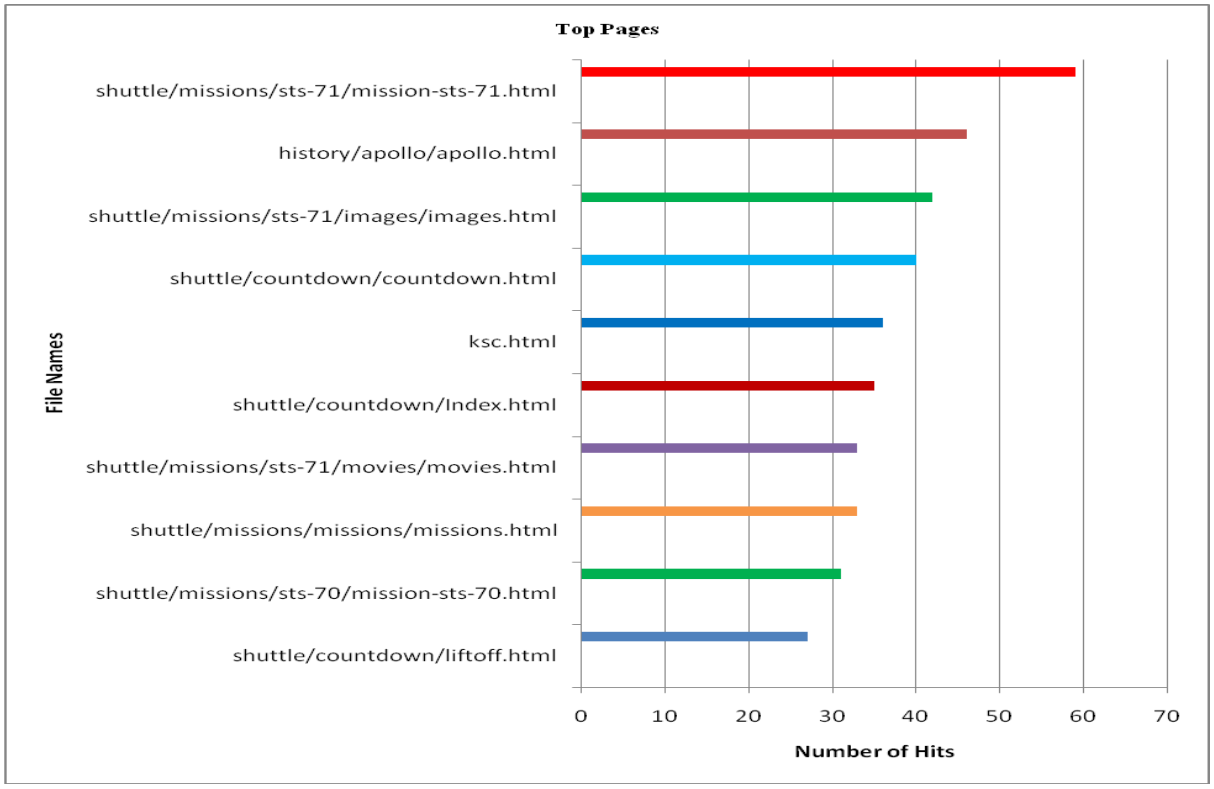


Fig.6 The above graph shows the result for the Top downloaded pages from the website.

© 2012, IJARCSSE All Rights Reserved

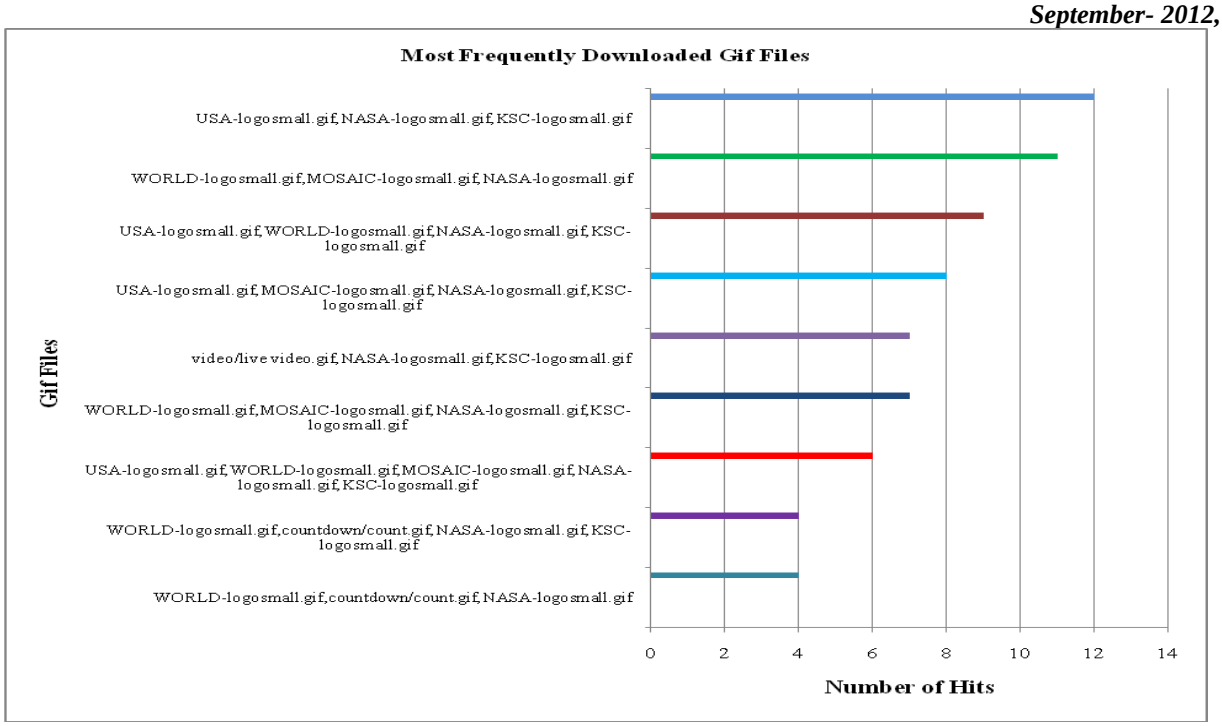


Fig.7 The above graph shows the result for the most frequently downloaded gif files from the website.

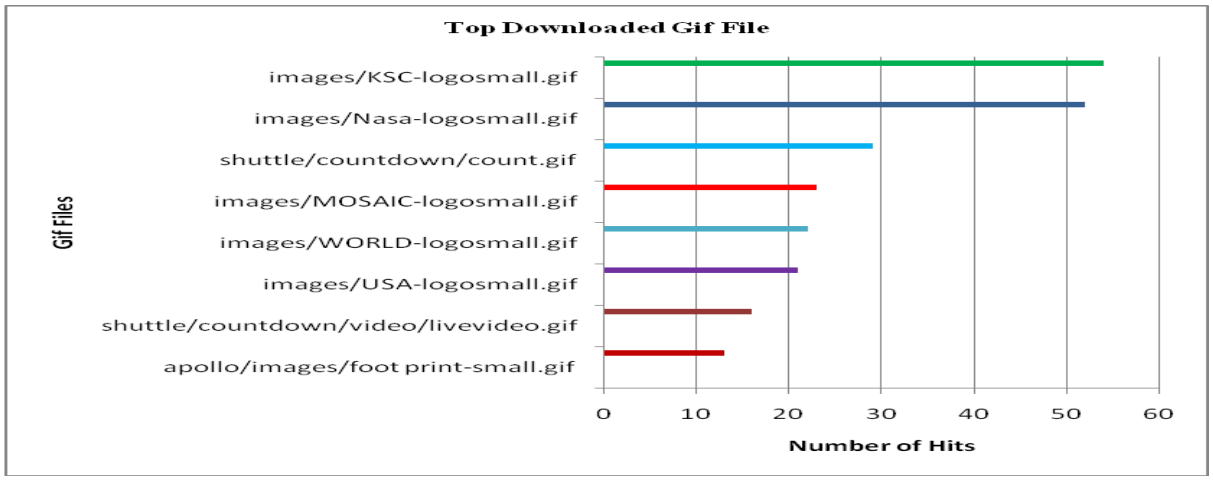


Fig.8 The above graph shows the result for Top Downloaded gif file from the website.

VIII. CONCLUSION AND SCOPE

The simulation result shows that the FP-Growth algorithm is used for finding the most frequently access pattern generated from the web log data, By using the concept of web usage mining we can easily find out the user's interest and we can modify and make our web site more valuable and more easily accessible for the users.The main goal of the proposed system is to identify usage pattern from web log files. FP Growth Algorithm is used for this purpose. Apriori is a classic algorithm for association rule mining. The main drawback of Apriori algorithm is that the candidate set generation is costly, especially if a large number of patterns and/or long patterns exist. The FP-growth algorithm is one of the fastest approaches for frequent item set mining. The FP-growth algorithm uses the FP-tree data structure to achieve a condensed representation of the database transaction and employees a divide-and conquer approach to decompose the mining problem. Our experimental result shows that the FP-growth method is efficient and scalable for mining both long and short frequent patterns. In future the algorithm can be extended to web content mining, web structure mining.

© 2012, IJARCSSE All Rights Reserved

REFERENCES

[1] Huiping Peng"Discovery of Interesting Association Rules Based on Web Usage Mining" 2010 International Conference.
[2] Sanjay Kumar Malik, Nupur Prakash, S.A.M. Rizvi" Ontology and Web Usage Mining towards an Intelligent Web focusing web logs" 2010 International Conference .
[3] Hao Yan, Bo Zhang, Yibo Zhang, Fang Liu, Zhenming Lei "Web usage mining based on WAN users' behaviours" 2010 International Conference.
[4] Han J., Pei J., Yin Y. and Mao R., "Mining frequent patterns without candidate generation: A frequent-pattern tree approach" Data Mining and Knowledge Discovery, 2004.
[5] Agrawal R, Srikant R., "Fast Algorithms for Mining Association Rules", VLDB. Sep 12-15 1994, Chile, 487-99, ISBN 1-55860-153-8.
[6] Mohd Helmy Abd Wahab, Mohd Norzali Haji Mohd ,Mohamad Mohsin"Data Pre-processing on Web Server Logs for Generalized Association Rules Mining Algorithm" 2008.
[7] C.P. Sumathi, r. padmaja valli**,**"An overview of preprocessing of web log files for web usage mining" 2011.
[8] Renata Ivancsy, Istvan Vajk "Frequent Pattern Mining in Web Log Data"2006.
[9] Vaibhav Kant Singh, Vijay Shah, Yogendra Kumar Jain, "Proposing an Efficient Method for Frequent Pattern Mining" 2008
[10] K. R. Suneetha, Dr. R. Krishnamoorthi, "Identifying User Behaviour by Analyzing Web Server Access Log File"2009

© 2012, IJARCSSE All Rights Reserved

PENERAPAN DATA MINING MENGGUNAKAN ALGORITMA FP-TREE DAN FP-GROWTH PADA DATA TRANSAKSI PENJUALAN OBAT

Yuyun Dwi Lestari
Program Studi Teknik Informatika, Sekolah Tinggi Teknik Harapan
Jl. H. M. Jhoni No. 70 C Medan
yuyun.dl@gmail.com

Abstrak
Penelitian ini membahas tentang bagaimana sistem kerja transaksi penjualan obat menggunakan Algoritma FP-Growth. Data transaksi penjualan obat ini digunakan untuk menemukan produk yang dibeli secara bersamaan. Penggunaan Algoritma FP-Growth untuk menemukan kombinasi pola barang. Penggunaan FP-Tree yang digunakan bersamaan dengan algoritma FP-Growth untuk menentukan frequent itemset dari sebuah database. Metode Association Rule digunakan dalam pencarian pola keterikatan produk untuk strategi penjualan dalam kebijakan pengambilan keputusan. Sehingga dapat diketahui obat yang sering dibeli oleh konsumen, berdasarkan rule-rule yang dihasilkan dari data-data yang terdapat di dalam database. Pengujian dilakukan dengan menggunakan Aplikasi Rapidminer 5. Hasil yang didapat dari pengujian tersebut adalah semakin kecil nilai support maka jumlah itemset yang dihasilkan akan semakin banyak dan jumlah rules yang dibentuk semakin banyak pula.
Kata Kunci : Obat, FP-Growth, FP-Tree, rule, support.

Abstract
This study discusses how the systems work using the drug sales transactions FP-Growth Algorithm . Drug sales transaction data is used to find a product that is purchased simultaneously . The use of FP -Growth algorithm to find the combination pattern of goods . The use of FP-Tree is used in conjunction with FP-Growth algorithm to determine frequent itemset from a database . Association Rule methods used in the search pattern for the product attachment sales strategy in the policy decision-making. So that it can be seen that the drug is often purchased by consumers , based on rules which generated from the data contained in the database. Testing is done by using the application RapidMiner 5. The results obtained from these tests is the smaller the value of support , the number of itemsets generated will be more and more and the number of rules which formed more and more also .
Keyword : Drug, FP-Growth, FP-Tree, rule, support

1. Pendahuluan

Obat merupakan kebutuhan terhadap seseorang yang menderita suatu penyakit. Obat dapat dibeli di apotek, rumah sakit atau tempat penyedia obat lainnya. Setiap hari akan terjadi transaksi penjualan dan pembelian obat di tempat tersebut. Oleh karena itu setiap apotek atau rumah sakit harus memiliki sistem pengolahan data agar data transaksi tersebut dapat digunakan dalam membuat laporan. Laporan dari data transaksi itu akan menghasilkan informasi yang berguna seperti obat apa saja yang paling sering dibeli atau obat yang paling banyak terjual.

Di apotek banyak terjadi transaksi penjualan dan pembelian obat. Data transaksi akan terus bertambah setiap harinya dan di apotek tersebut data transaksi penjualan hanya disimpan sebagai arsip atau pembukuan dan tidak diketahui apa manfaat dari data-data tersebut untuk selanjutnya. Sistem pengolahan data yang ada di apotek belum berjalan dengan baik karena data-data penjualan yang ada dalam pembukuan hanya dibiarkan menumpuk tanpa diketahui untuk apa selanjutnya data-data tersebut.

Oleh karena itu apotek tersebut memerlukan sistem untuk mengolah data yang dapat menghasilkan data penjualan obat yang paling sering dibeli atau paling banyak terjual sehingga dari hasil tersebut dapat menjadi acuan untuk menambah stok obat yang habis dan mengurangi beberapa obat yang jarang dibeli oleh konsumen.

Tujuan penelitian ini adalah :

1. Menggali informasi dari data penjualan obat sehingga manajer mendapatkan informasi lebih yang dapat digunakan untuk pengambilan keputusan atau kebijakan strategis perusahaan.
2. Memahami konsep *FP-Growth* untuk transaksi penjualan obat.
3. Menampilkan hubungan assosiasi *item* obat yang sering dibeli oleh konsumen untuk mengetahui *item-item* yang saling berkaitan.
4. Mengimplementasikan analisa data menggunakan algoritma *Fp-Growth* untuk menghasilkan penerapan dari *Data Mining* dalam

mengoptimasi penjualan obat menggunakan *software* Rapidminer 5.

Data Mining adalah proses mencari pola atau informasi menarik dalam data terpilih menggunakan teknik atau metode tertentu. Salah satu metode yang seringkali digunakan dalam *Data Mining* adalah metode *Association Rule*.

Algoritma *FP-Growth* merupakan pengembangan dari algoritma Apriori. Algoritma *Frequent Pattern Growth* adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. [5]

Pada algoritma *FP-Growth* menggunakan konsep pembangunan *tree*, yang biasa disebut *FP-Tree*, dalam pencarian *frequent itemsets* bukan menggunakan *generate candidate* seperti yang dilakukan pada algoritma *Apriori*. Dengan menggunakan konsep tersebut, algoritma *FP-Growth* menjadi lebih cepat daripada algoritma *Apriori*. [2]

Metode *FP-Growth* dibagi menjadi tiga tahapan utama, yaitu [4]:

1. tahap pembangkitan *conditional pattern base*,
2. tahap pembangkitan *conditional FP-Tree*, dan
3. tahap pencarian *frequent itemset*.

Association rule merupakan sutu proses pada data mining untuk menentukan semua aturan asosiatif yang memenuhi syarat minimum untuk *support (minsup)* dan *confidance (minconf)* pada sebuah database. Kedua syarat tersebut akan digunakan untuk *interesting association rules* dengan dibandingkaan dengan batasan yang telah ditentukan, yaitu *minsup* dan *minconf*. [3]

Association Rule Mining adalah suatu prosedur untuk mencari hubungan antar *item* dalam suatu *dataset*. Dimulai dengan mencari *frequent itemset*, yaitu kombinasi yang paling sering terjadi dalam suatu *itemset* dan harus memenuhi *minsup*. [3]

Dalam tahap ini akan dilakukan pencarian kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam database. Untuk mendapatkan nilai *support* dari suatu *item* A dapat diperoleh dengan rumus berikut: [1]

$$Support\ (A) = \frac{Jumlah\ transaksi\ yang\ mengandung\ Item\ A}{Total\ Transaksi}$$

Kemudian, untuk mendapatkan nilai *support* dari dua item diperoleh dengan rumus berikut:

$$Support\ (A,B) = P(A \cap B) = \frac{Jumlah\ Transaksi\ yang\ mengandung\ A\ dan\ B}{Total\ Transaksi}$$

Setelah semua *frequent item* dan *large item set* didapatkan, dapat dicari syarat *minimum confidence (mincof)* dengan menggunakan rumus berikut :

$$Confidence\ (A \rightarrow B) = P(A|B) = \frac{Jumlah\ Transaksi\ yang\ mengandung\ A\ dan\ B}{Jumlah\ Transaksi\ yang\ mengandung\ A}$$

2. Metode Penelitian

Metode dalam penelitian ini adalah sebagai berikut:

1. Identifikasi Masalah
   Permasalahan yang ditemukan penulis, dideskripsikan dengan jelas sehingga akan terlihat permasalahan yang akan dibahas.
2. Analisis Masalah
   Langkah analisis masalah adalah langkah untuk memahami yang telah ditentukan ruang lingkup atau batasannya. Dengan menganalisis masalah yang telah ditentukan tersebut, maka diharapkan masalah tersebut dapat dipahami dengan baik.
3. Menentukan Tujuan
   Tujuan dari penelitian ini untuk menentukan hasil analisa data penjualan menggunakan algoritma *fp-tree* dan *fpgrowth*.
4. Mempelajari Literatur
   Untuk mecapai tujuan yang akan ditentukan, maka perlu dipelajari beberapa literatur-literatur yang sesuai dengan penelitian yang akan dilakukan. Kemudian literatur-literatur yang dipelajari tersebut diseleksi untuk dapat ditentukan literatur mana yang akan digunakan dalam penelitian. Sumber literatur didapat dari buku-buku dan jurnal-junal yang ada hubungannya dengan penelitian ini maupun referensi yang lain.
5. Mengumpulkan Data
   Pengumpulan data dan informasi pada tahap ini dilakukan untuk mengetahui, mendapatkan data dan informasi yang nantinya akan mendukung penelitian ini, dalam pengumpulan data, terdapat beberapa metode yang digunakan yaitu
   a. Penelitian Lapangan (*field research*)
      Penelitian dilakukan pada apotek yaitu melakukan wawancara dengan pihak apotek untuk meneliti data-data transaksi penjualan obat yang terkait dengan permasalahan ini.
   b. Penelitian Perpustakaan (*library research*)
      Penelitian ini dilakukan untuk mengumpulkan informasi tentang literatur dan pedoman dalam penentuan hasil analisa data transaksi penjualan obat.
   c. Penelitian Laboratorium (*laboratory research*).
      Penelitian laboratorium ini dimaksudkan untuk pengujian data penentuan hasil analisa data penjualan yang telah didapatkan dengan

menggunakan algoritma *fp-tree* dan *fp-growth*.

6. Analisa Data dengan Algoritma *fp-tree* dan *fp-growth*
   Pada tahap ini akan dilakukan dua analisa yaitu analisa terhadap algoritma *fp-tree* dan *fp-growth* pada data transaksi penjualan untuk menentukan alternatif yang terbaik dari sekumpulan alternatif yang ada melalui suatu proses seperti yang terjadi pada proses yang terstruktur dan analisa bagaimana algoritma *fp-tree* dan *fp-growth* untuk mendapatkan hasil sebagai tujuan yang akan dicapai oleh peneliti yang kemudian dapat dijadikan pihak apotek sebagai pengetahuan dalam meningkatkan penjualan obat kepada konsumen.
7. Merancang Skema
   Pada bagian ini akan menampilkan bagaimana proses analisa *data mining* dengan algoritma *fp-tree* dan *fp-growth* dirancang berdasarkan data yang telah terkumpul. Dan bagaimana mengembangkan proses analisa *data mining* dengan *assosiation rule* dan algoritma *fp-tree* dan *fp-growth* untuk mendapatkan hubungan keterikatan antar tiap barang yang ada. Perancangan ini dilakukan pada data transaksi penjualan untuk mengetahui obat yang paling banyak dibeli oleh konsumen.
8. Implementasi
   Pada penelitian ini penulis mengimplementasikan berdasarkan hasil analisa data menggunakan algoritma *fp-tree* dan *fp-growth* dengan menggunakan alat bantu komputer dengan *operating system windows* dan menggunakan *software* Rapidminer 5.

3. Hasil dan Pembahasan

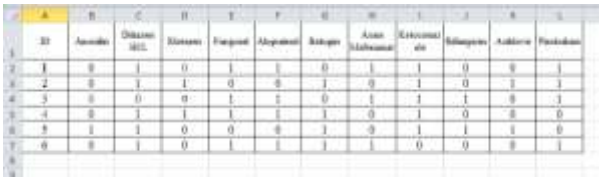
3.1 Pengolahan Data

Pengolahan data dilakukan menggunakan aplikasi database yaitu Microsoft Excel 2007. Data real transaksi penjualan dapat dilihat pada gambar berikut.

ID Transaksi	ItemSet
1	Ditiazem HCL, Ketoconazole, Fungoral, Alopurinol, Piroksikam, Asam Mefenamat
2	Ditiazem HCL, Ketoconazole, Batugin, Piroksikam, Klotaren, Asiklovir
3	Ketoconazole, Fungoral, Alopurinol, Piroksikam, Asam Mefenamat, Rifampicin
4	Ditiazem HCL, Ketoconazole, Fungoral, Alopurinol, Batugin, Klotaren
5	Ditiazem HCL, Ketoconazole, Batugin, Rifampicin, Asiklovir
6	Ditiazem HCL, Fungoral, Alopurinol, Batugin, Piroksikam, Asam Mefenamat

Gambar 1 Data Real Transaksi Penjualan Obat

Data real transaksi penjualan kemudian disusun ke dalam bentuk tabular data, maka data real transaksi penjualan obat dikonversikan ke dalam bentuk 1 dan 0 atau bentuk biner. Dimana 1 adalah jika obat dibeli dan 0 jika obat tidak dibeli. Hasil proses konversi data transaksi penjualan dalam bentuk tabular data dapat dilihat seperti pada gambar berikut.



Gambar 2 Tabular Data Transaksi Penjualan Obat

3.2 FP-Tree

Tahap ini adalah tahap dimana dataset yang telah dibatasi dengan menggunakan *support count* yang telah ditentukan, kemudian dibangun menjadi sebuah *Tree*.

Berikut ini adalah tabel dengan semua barang yang dalam satu transaksi sudah disatukan.

Tabel 1Data Transaksi Awal

No	ItemSet
1	Ditiazem HCL, Ketoconazole, Piroksikam, Fungoral, Alopurinol, Asam Mefenamat
2	Ketoconazole, Ditiazem HCL, Piroksikam, Klotaren, Batugin, Asiklovir
3	Asam Mefenamat, Ketoconazole, Piroksikam, Rifampicin, Alopurinol, Fungoral
4	Ditiazem HCL, Ketoconazole, Alopurinol, Batugin, Klotaren, Fungoral
5	Ketoconazole, Ditiazem HCL, Amoksilim, Rifampicin, Batugin, Asiklovir
6	Ditiazem HCL, Fugoral, Batugin, Piroksikam, Asam Mefenamat, Alopurinol

Kemudian menentukan frekuensi setiap *item* dari keseluruhan transaksi.

Tabel 2 Nama dan Frekuensi Item dari Data Transaksi Awal

No	Nama Obat	Jumlah
1	Amoksilin	1
2	Ditiazem HCL	5
3	Klotaren	2
4	Fungoral	4
5	Alopurinol	4
6	Batugin	4
7	Asam Mefenamat	3
8	Ketoconazole	5
9	Rifampicin	2
10	Asiklovir	2
11	Piroksikam	4

Setelah frekuensi setiap *item* diperoleh, kemudian dibatasi dengan *support count*. Jika frekuensi *item* tidak kurang dari *support count*, maka *item* tersebut akan dihapus dan tidak dipakai dalam proses *data mining*. Misalkan ditentukan *support count* ξ =2, maka hasilnya adalah:

Tabel 3 Nama dan Frekuensi Item Setelah Proses Filter

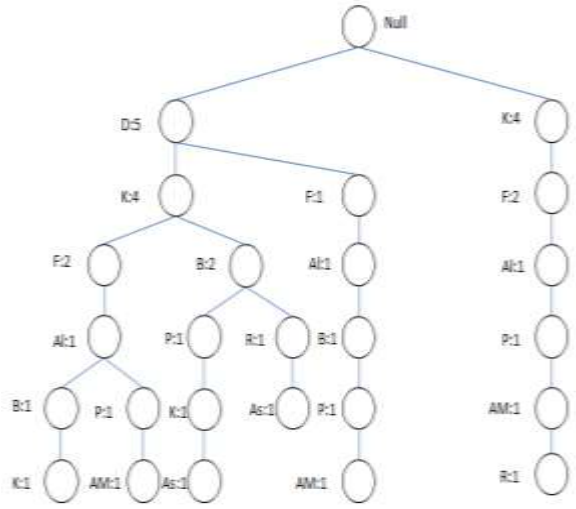
No	Nama Obat	Jumlah
1	Ditiazem HCL	5
2	Ketoconazole	5
3	Fungoral	4
4	Alopurinol	4
5	Batugin	4
6	Piroksikam	4
7	Asam Mefenamat	3
8	Klotaren	2
9	Rifampicin	2
10	Asiklovir	2

Item Air hilang karena frekuensinya tidak lebih dari sama dengan 2. Tahap selanjutnya adalah pembangunan *Tree* berdasarkan per transaksi dengan item yang telah dibatasi.

Tabel 4 Data Transaksi Setelah Proses *Filter*

No	**ItemSet**
1	Ditiazem HCL, Ketoconazole, Fungoral, Alopurinol, Piroksikam, Asam Mefenamat
2	Ditiazem HCL, Ketoconazole, Batugin, Piroksikam, Klotaren, Asiklovir
3	Ketoconazole, Fungoral, Alopurinol, Piroksikam, Asam Mefenamat, Rifampicin
4	Ditiazem HCL, Ketoconazole, Fungoral, Alopurinol, Batugin, Klotaren
5	Ditiazem HCL, Ketoconazole, Batugin, Rifampicin, Asiklovir
6	Ditiazem HCL, Fungoral, Alopurinol, Batugin, Piroksikakm, Asam Mefenamat

Pembangunan *Tree* dari seluruh transaksi tersebut adalah



Gambar 3 Tree dari semua transaksi

3.3 FP-Growth

Setelah FP-Tree terbentuk, maka langkah selanjutnya adalah tahap pembangkitan *conditional pattern base*, tahap pembangkitan *conditional FP-Tree*, dan tahap pencarian *frequent itemset*. Pada tahap ini dapat dilakukan dengan melihat kembali FP-Tree yang sudah dibuat sebelumnya. [6]

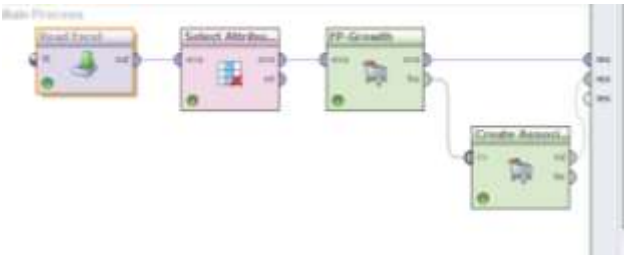
a. Tahap Pembangkitan *Conditional Pattern Base*
   Conditional Pattern Base merupakan subdata yang berisi *prefix path* (lintasan awal) dan *suffix pattern* (pola akhiran). Pembangkitan *conditional pattern base* didapatkan melalui *FP-Tree* yang telah dibangun sebelumnya.
b. Tahap Pembangkitan *Conditional FP-Tree*
   Pada tahap ini, *support count* dari setiap *item* pada setiap *conditional pattern base* dijumlahkan, lalu setiap *item* yang memiliki jumlah *support count* lebih besar atau sama dengan minimum *support count* akandibangkitkan dengan *conditional FP-Tree*.
c. Tahap Pencarian *Frequent Itemset*
   Apabila *Conditional FP-Tree* merupakan lintasan tunggal (*single path*), maka didapatkan *frequent itemset* dengan melakukan kombinasi *item* untuk setiap *conditional FP-Tree*. Jika bukan lintasan tunggal, maka dilakukan pembangkitan *FP-Growth* secara *rekursif* (proses memanggil dirinya sendiri)

Untuk pengujian dari data transaksi penjualan yang telah dihasilkan berupa pola hubungan kombinasi antar *items* dan *association rules* sesuai dengan Algoritma FP-Growth, maka digunakan aplikasi Rapidminer untuk menguji analisa pembelian obat yang dilakukan oleh konsumen.

Adapun langkah-langkah pembentukan model mining menggunakan algoritma *FP-Growth* pada *Software Rapidminer* antara lain :

1. Memilih.file format xls yang akan dimining.
2. algoritma yang digunakan *algoritma FP-Growth*.
3. Mengatur minimal *support, confidence* dan *rules* yang dihasilkan.

Evaluasi data mining akan dilakukan dengan menggunakan dataset pengujian yang berisikan 6 transaksi penjualan obat yang akan dilakukan perhitungan menggunakan algoritma *FP-growth*. Langkah kedua dataset akan dijalankan melalui model mining yang dibuat pada RapidMiner 5. Berikut adalah poses koneksi data menggunakan algoritma *FP-Growth.*



Gambar 4 Proses koneksi data menggunakan *FP-Growth*

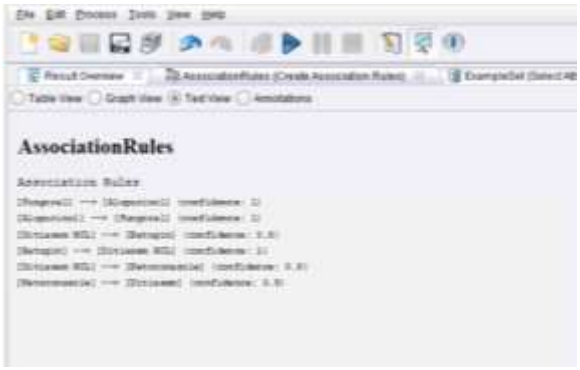
3.4 Association Rule

Pada tahap ini digunakan untuk menentukan nilai *support* dan *confidence* pada setiap itemset dengan rumus yang sudah dijelaskan sebelumnya pada dasar teori. Pada kasus di atas, misalkan diberikan nilai *minimum support* = 0,6 dan *minimum confidence*= 1, maka hasilnya adalah:

.

No.	Premises	Conclusion	Support	Confidence	Lift
1	Fungoral	Alopurinol	0.667	1	1.5
2	Alopurinol	Fungoral	0.667	1	1.5
3	Ditiazem HCL	Batugin	0.667	0.8	1.2
4	Batugin	Ditiazem HCL	0.667	1	1.2
5	Ditiazem HCL	Ketoconazole	0.667	0.8	0.96
6	Ketoconazole	Ditiazem HCL	0.667	0.8	0.96

Gambar 5 Hasil *Frequent Item Set*



Gambar 6 Rules yang Dihasilkan

4. Kesimpulan

Berdasarkan analisa dan hasil pengujian maka dapat diperoleh kesimpulan sebagai berikut:

1. *Data Mining* dengan Algoritma *FP-Growth* dapat diimplementasikan dengan menggunakan *database* penjualan obat karena dapat menemukan pola kombinasi *itemsets*. Sehingga informasi tersebut dapat membantu mengembangkan strategi penjualan terhadap konsumen.
2. Metode *Association Rule* dengan menggunakan Algoritma FP-Growth dengan parameter *support* dan *confidence* dapat memperoleh korelasi barang pembelian untuk lebih meningkatkan penjualan
3. Setelah dilakukan pengimplementasian Algoritma *FP-Growth* pada Rapidminer, maka hasil pengolahan data penjualan obat paling banyak terjual pada apotek adalah Fungoral, Alopurinol, Diatizem HCL, Batugin dan Ketoconazole.

References

[1] Dyah Pramesthi Larasati, Muhammad Nasrun, Umar Ali Ahmad. 2014. Analisis Dan Implementasi Algoritma FP-Growth pada Aplikasi Smart Untuk Menentukan Market Basket Analysis Pada Usaha Retail (Studi Kasus : PT. X).

[2] Erwin. 2009. Analisis Market Basket dengan Algoritma Apriori dan FP-Growth. Jurnal Generic 26-30.

[3] Fatihatul F., Setiawan A., Rosadi R.. *Asosiasi Data Mining Menggunakan Algoritma FP-Growth untuk Market Basket Analysis*. Tersedia : http://www.academia.edu/4758451/ASOSIASI_DATA_MINING_MENGGUNAKAN_ALGORITMA_FP-

GROWTH_UNTUK_MARKET_BASKET_ANALYSIS. [Diakses tanggal 02 Agustus 2015]

[4] Han Jiawei, and M. Kamber. 2006. Data Mining: Concepts and Techniques, Morgan Kaufmann, USA.

[5] Samuel, David. 2008. *Penerapan Struktur* FP-Tree *dan Algoritma* FP-Growth *dalam Optimasi Penentuan* Frequent Itemset. Institut Teknologi Bandung

[6] Anfatul Maulida, Tia. 2014. Analisa Data Mining Menggunakan Algoritma *Frequent Pattern Growth* pada Data Transaksi Penjualan Restauran Joglo Kampoeng Doeloe Semarang.

The Analysis and Implementation of Algorithm of Frequent Pattern – Growth to Support the Promotion Strategy in Victory University Sorong

Iriene Surya Rajagukguk[#1], Sri Yulianto Joko Prasetyo[*2], Irwan Sembiring[#3]
[#]*Faculty of Information System, Satya Wacana Christian University*
Diponegoro Street Number 52-60 Salatiga, Central Java, Indonesia

Abstract

Nowadays, the increasing number of campuses is growing, which makes want to get a better promotion strategy. As for how to find the right promotional strategy will be able to reduce the cost of promotion and achieve the right promotional goals. One way that can be done for the determination of a promotion strategy is to use data mining techniques. The technique used in this case is FP-Growth Algorithm is an alternative algorithm that can be used to determine the most frequent item set in a data set. Research is done by observing some research variables that are often considered by universities, especially the marketing section in determining the promotional target of the last education, the origin of the region, the choice of major and promotion strategy. The result of this research is a suitable promotion strategy to be applied at Victory University Sorong namely advertising, personal selling and publicity. And the most influential is advertising, this can be seen from the amount of the advertising value of 35 transaction from the total sample number of 43 transaction with a significant level 81.40 % of the other dimensions. Personal selling 6 transactions, the effect is 13.95 % a publicity as much as 2 transactions or 4.65 % of the other dimensions.

Keywords

Data mining, association rules, frequent item set, FP-growth algorithm.

I. INTRODUCTION

The competition in this world business is very strict, especially in promoting the university. Every campus tries to give the best facility. Therefore each campus tries to find out the exact strategy to promote its university. According to the developers to find a strategy that can determine marketing strategy in promoting university by maximizing services to the community.

One of technique that use in the implementation of Data Mining is in the promotion scale. If the promotion target is not determined well, in the sense of not seeking potential promotional target, then just spend a lot of time and cost that should be minimized through the selection of good promotional target.

As for in this research will be discussed how to apply one of algorithm in the data mining, that is Frequent Pattern-Growth (FP-Growth) algorithm. This algorithm is part of association technique on the data mining. As for FP-Growth is one of algorithm alternative that can be used to determine frequent item set in a set of data. The characteristic of FP-Growth algorithm is the data structure that used on the tree called FP-Tree. By using FP-Tree, FP-Growth algorithm can directly extract frequent item set from FP-Tree.

Victory University Sorong is private university that is located in Sorong West Papua. As time goes by, then began many emerging private universities and high schools in Sorong or Sorong Regency. Of course to face this competition required strategies to establish the existence of Victory University Sorong. The increasing number of private university and requires Victory University Sorong to conduct a diverse range of active promotions, in order to not less competitive with other universities.

This research will do the grouping based on Victory Sorong student data. The attributes to be used in determining the grouping of territories for promoting the Victory University is the latest education, origin, choice of major and promotion strategy. The result of this student data processing aims to assist the marketing of Victory University Sorong in conducting promotion and search for new students from various regions in Sorong.

The purpose of this research is to analyze and apply the promotion strategy in Victory University Sorong.

II. LITERATUR REVIEW

Research [1] [Improved Algorithm for Frequent Item sets Mining Based on A priori and FP-Tree] In this paper the researchers experiments based on two algorithms into one such APFT, which combines a priori algorithm and FP-tree structure of the FP-growth algorithm. The advantage of APFT does not

produce conditional & sub-conditional patterns of trees recursively and experimental results show that FP-tree works quickly than A priori and is almost as fast as FP-Growth. They have proposed to go further and modify APFT to include correlated items & uncorrelated item set. This additional feature optimizes the FP-Tree & removes the assembled items directly from the item set frequent. They chose to call this method as APCFT method with APFT correlation.

Research [2] Efficient Implementation of FP Growth Algorithm-Data Mining on Medical Data] Implementation of data mining on medical data to generate rules and patterns using FP-Growth algorithm. The research study in this paper presents how data mining can apply to medical data. Therefore, data and empirical results are presented in this paper to provide more guidance to the doctors as well as to better understand the relationship of the doctors and a patient. For that, the first discuss about the importance of data mining using medical data and then the discussion of general data mining techniques have been presented. Next, methodology describes conceptual model for extraction rules about the medical database finally the result can guide the relationship between the different attributes presented in the data.

Research [3] [FP Growth Algorithm Implementation] This paper discusses FP Tree concept and apply it uses Java for general social survey dataset. They use this approach to determine the association rules that happens in dataset. By this way, can set relevant rules and patterns in a set of records.

Research [4] [FP-Tree Based Algorithms Analysis: FP Growth] Mining frequent item sets from large transactional databases have important and very critical tasks. Many algorithms have been proposed from recent years. Yet FP-Tree-like algorithms are regarded as highly effective algorithms for efficient frequent item set circuits. This algorithm is considered efficient because of its compact structure and also a slightly item set generation compared to the A priori Algorithm. Therefore, the purpose of this paper is to provide a basic concept of some algorithms (FP-Growth, COFI-Tree, CT-PRO) based on FP-Tree such as the structure to mine frequent item sets equal to the ability and comparison.

Research [5] [FP Growth Tree Implementation In Bank Transaction Databases] In this paper describes the application of FP-Growth algorithm in the database of bank transactions to ensure the best performance. The purpose of this research to ponder and test existing association techniques for collecting frequent mining items and successful implementation of the FP-Growth algorithm in a database of banking transactions to improve the performance of frequent item sets in data mining.

Research [6] [Overview on Methods for Mining High Utility] In this paper, proposed two algorithms, namely the growth of utility pattern (UP-Growth) and UP-Growth +, to mine item set with high utility with a set of effective strategies for item set candidate. The information of high utility item set is maintained in Up-Tree, item set candidate can be generated efficiently with two database scans. The result of experiments shows that the proposed algorithm, especially UP-Growth+, not only reduces the number of candidates effectively but also outperforms other algorithms sub-stance in terms of run time, especially when the database contains many long transactions.

Research [7] [A Survey on Association Rule Mining in Market Basket Analysis] This method examine customer purchase patterns by identifying associations between different items where customers are in the shopping cart. Identification of such associations can help retailers broaden their marketing strategy by gaining insight into what customers often buy. This paper presents a survey of existing data mining algorithms for shopping cart analysis.

Research [8] [Market Basket Analysis for a Supermarket based on Frequent Item set Mining] In this research, shopping cart Analysis is am importance component in the system analysis in retail organization to determine placement of stuff, design the marketing promotion for various segment of customer to increase customer satisfaction and profit from the supermarket. These issues are aimed at leading supermarkets by using frequent items set mining. The item set is mined from the market basket database using the efficient K-A priori algorithm and the resulting association rules.

The difference between this research and the previous one is on this research has been found promotion strategy that is suitable to use at Victory University Sorong along with 3 promotion mixes are: advertisement dimension, personal selling dimension, and publicity dimension.

A. FP-Growth Algorithm

FP-Growth is one of algorithm alternative that can be use to determine the frequent item set in a data set. On the FP-Growth algorithm uses tree development concept, commonly called FP-Tree, in searching frequent item set is not using generate candidate as it is done on A priori algorithm. By using the concept, FP-Growth algorithm becomes faster than A priori algorithm. [9]

FP-Growth Method is divided into three main stages:

- Generation stage conditional pattern base,
- Generation stage conditional FP-Tree, and
- Searching stage frequent item set.

B. Association Rules

Association rule is one of method that aims to find the frequent pattern that often apper among many transactions, where each transaction consists of several items so that this method will support the recommendation system through the discovery of

patterns between items in transactions that happened. [10] The basic methodology of association analysis is divided into two stages:

- High frequency pattern analysis

This stage looks for a combination of items that fulfil the minimum requirements of the support value in the database. The value of an item's support is obtained by the following formula:

$$Support\ (A) \frac{\text{The number of transaction contains A}}{\text{Transaction Total}} \quad (1)$$

Explanation:

A: which school are you from

B: origin

On the formula 1 explains that support value (A) is obtained by finding the number of transactions containing A in divide by total transaction.

- Formation of Associative Rules

After all the frequency patterns are found, then find the associative rules that supply the minimum requirements for confidence by calculating the confidence of the associative rules A_B The confidence value of rule A_B is obtained from the following formula:

$$Confidence = \frac{\in \text{Transaction contains A and B}}{\in \text{Transaction contains A}} \quad (2)$$

Explanation:

A: which school are you from

B: origin

III.RESEARCH METODOLOGY

In the research methodology there is a sequence of frameworks to be followed, the order of this framework is a description of the steps that must be passed so that this research can work well. The framework to follow can be seen on figure 1.

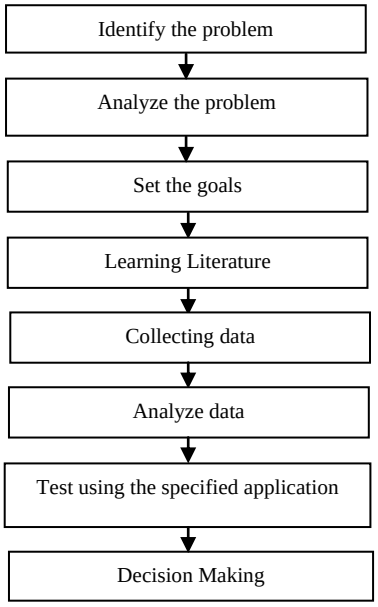


Fig 1: Research Framework [11]

Based on Figure 1 the work sequence steps are described as follows:

A. *Identifying Problem*

The problem identified in this research is to find the cause of the decreasing number of new student candidates amidst intense competition among universities.

B. *Analyzing the Problem*

Problem analysis in this research is done by two methods that are descriptive method and comparative method.

- Descriptive method

In this method, the data is submitted, arranged, classified and analyzed so that the clear description of the problem which will be discuss can be taken.

- Comparative method

In this method, analysis is done by comparing the theory and practice so that the description about the similarities and the differences between of them can be taken.

C. *Determining the Aim*

Based on the explanation above, the aim of this research is to know how far is the implementation of promotion strategy which has been done by Victory Sorong University.

D. *Studying Literature*

The literature which is used as reference in this research is from scientific journals, learning module and books about Data Mining. These literatures will be guidelines to do the research in order to make easy the process of the research.

E. *Collecting Data*

The method of collecting data is done by doing the observation to the College of Victory Sorong University. Besides of observation, the interview is also done toward the related parties of this research.

F. *Analyzing Data*

In this stage, the data which has been collected will be analyzed. Analysis using Association Rule is done by support and confidence counting mechanism from an item relation. An association rule can be said as important if the support value is bigger than minimum support and also the confidence value is bigger than minimum confidence. By using FP-Growth technic which result Frequent Itemset without doing candidates generation with the purpose to get new knowledge in form of information about the effective and efficient promotion strategy in getting the candidate of new student in the middle of the binding competition between Universities.

G. *Testing Use Certain Application*

In this stage, rules are tested or examined using data mining system. Tools which are used as system testing is Rapidminer Studio 6.4.for the first, the manual data "the data of the new student candidate

2008-2016" which is available is transformed into Microsoft Excel.

As the first stage in the testing process is importing the data of new student in the collage of Victory Sorong University which will be made as the table of item frequency transaction on tools Rapidminer Studio 6.4. FT-Growth analysis is definite as a process to find all of FT-Growth rule which fulfill the minimum support requirement and minimum requirement for confidence. From the process of Data Input and Affinity, it will be found a certain pattern or criteria on getting promotion strategy and will be processed back to result an association rule. From the association rule, the result will be seen. The result is a new knowledge and also as reference to analyze the strategy on promoting education.

H. Taking Decision

After being examined, the result of analysis will show the comparison between the manual manner and tool utilizing. The next stage is determining and decision taking toward new knowledge which is gotten in form of finding related with the strategy in promoting education.

IV. RESULT OF THE RESEARCH AND DISCUSSION

The characteristic of FP-Growth algorithm is data structure which using FP-Tree. By using FP-Tree, FP-Growth algorithm can directly extract frequent itemset from FP-Tree. Itemset excavation by using FP-Growth algorithm will be done by generating the structure of data tree or usually known as FP-Tree.

Input: FP-Tree Tree

Output: Rt complete group of frequent pattern

Method: FP-Growth (Tree, null)

Procedure: FP-Growth (Tree, Θ)

[

01: if Tree consist single path P;

02: then to every combination (detonated β)

From the nodes in path do

03: generate β ά pattern with the support from nodes in β;

04: else for every a1 on the header of Tree do

05: generating pattern

06: form β=a1 ά with the support = a1.

Support

07: if Tree β = ά

08: then summon FP-Growth (Tree, β)

}

}

FP-Growth Algorithm [11]

Before doing the mining process, preprocessing is done first by changing the format of the data so that it can be used on the software. The appearance of the stages and the result is as the follows:

TABLE I
ITEMSET

TID	Itemset
1	{A3,B4,C6}
2	{A4,B4,C6}
3	{A3,B6,C6}
4	{A1,B4,C6}
5	{A3,C6}
6	{A3,B6,C6}
7	{A4,B4,C6}
8	{A1,B4,C6}
9	{A1,B1,C6}
10	{A1,B1,C6}
11	{A1,B4,C6}
12	{A1,B4,C6}
13	{A2,B1,C6}
14	{A4,B1,C6}
15	{A3,B4,C6}
16	{A2,B1,C6}
17	{A1,B4,C6}
18	{A1,C6}
19	{A1,C6}
20	{A1,C6}
21	{A1,C6}
22	{A1,C6}
23	{A2,B4,C6}
24	{A1,B4,C6}
25	{A2,B4,C6}
26	{A2,B4,C6}
27	{A3,B4,C6}
28	{A2,B4,C6}
29	{B4,C6}
30	{A3,B4,C6}

So, the next stage is forming the tree of FP-Tree by seeing the table 1 as a follow:

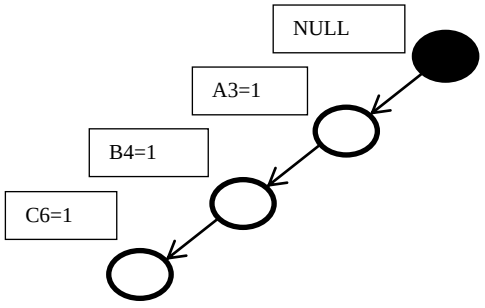


Fig 2: The result of fp-tree forming after reading tid 1

Explanation:

= Root

= Data Flows

= Branch

Figure 2: is the explanation of FP-Tree forming after reading TID 1 which contains: NULL - A3 (Public SMA) = 1 – B4 (Ayamaru) = 1 – C6 (Indonesian language education) = 1.

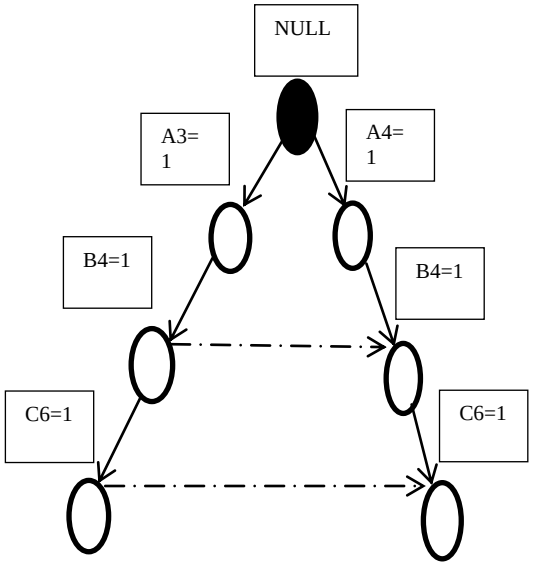


Fig 3: The result of fp-tree forming after reading tid 2

Explanation:

= Root
= Data Flows
= Branch
= same data flows

Figure 3: is gotten after doing TID 2 that is: NULL - A4 (Private SMA) = 1 - B4 (Ayamaru) = 1 – C6 (Indonesian language education) = 1.

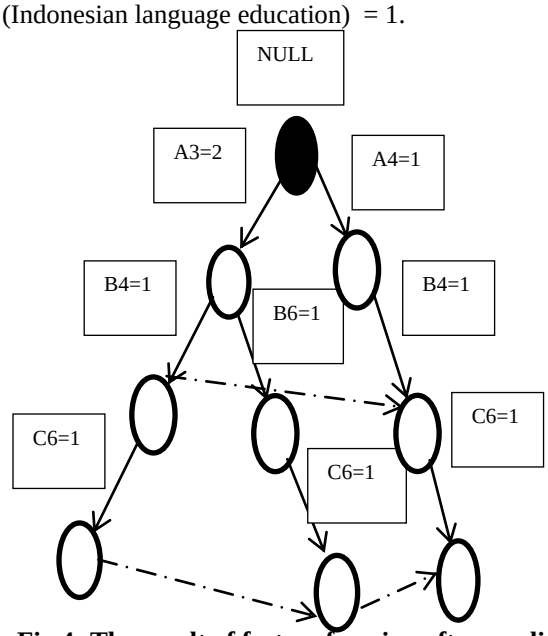


Fig 4: The result of fp-tree forming after reading tid 3

Explanation:

= Root
= Data Flows
= Branch
= Same data flows

Figure 4: is gotten after doing TID 3 that is: NULL – A3 (Public SMA) = 2 - B6 (Tamrauw) = 1 – C6 (Indonesian language education) = 1.

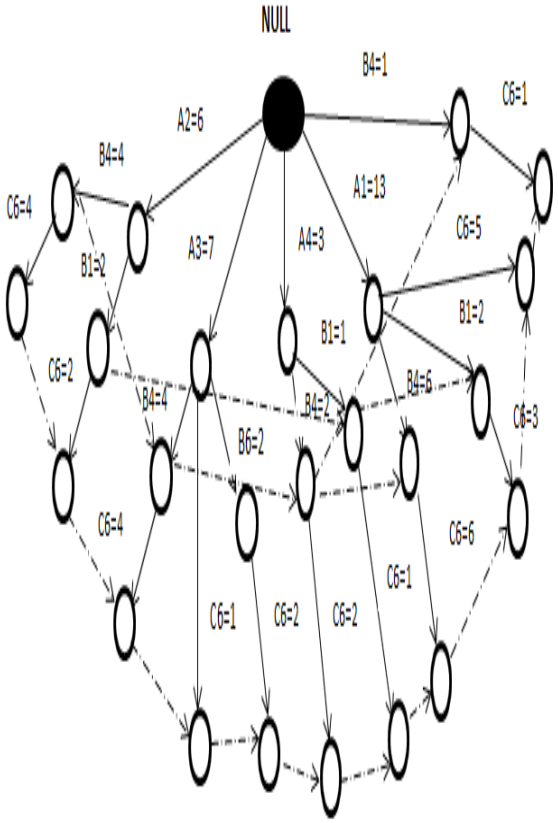


Fig 5: The result of fp-tree forming after reading tid 30

Explanation:

= Root
= Data Flows
= Branch
= same data flows

Figure 5: is gotten after doing TID 3 that is : NULL – A3 (Public SMA) = 7 – B4(Ayamaru) = 4 – C6 (Indonesian language education) = 4.

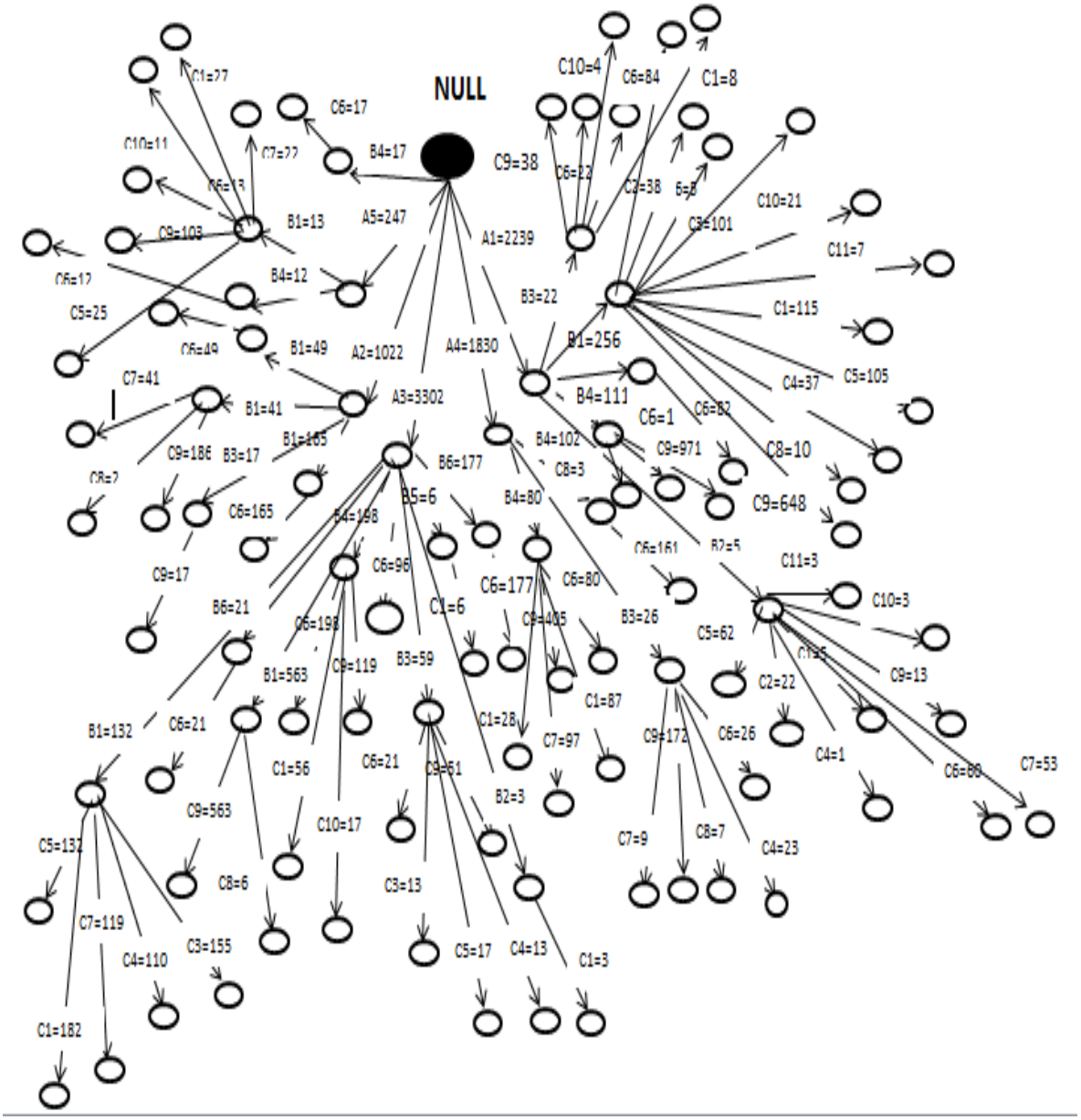


Fig 6: The result of fp-tree after reading tid 8729

Figure 6 is gotten after doing TID 8729. From frequent itemset which is gotten from FP-Tree forming and FP-Growth so the support and confidence value can be counted as the follows:

- Support (Public SMA, Sorong, Electro) = Count (Public SMA, Sorong, Electro)/ the sum of transaction = 1/8729.

While for the confidence our trusted value is as the follows:

- Confidence (Public SMA, Sorong, Electro) = Count (government supported senior high school, Sorong, Electro)/ the sum of transaction = 1/51.

After support and confidence value is gotten from the whole combination on the data with FP-Tree and FP-Growth counting so there is found the highest and most accurate support and confidence value that is the combination (A3, B1, C9) {Public SMA, Sorong, Information system} which has support value: 563/8729)=0,064 and confidence value : 563/4139=0,136.

A. Promotion Strategy

This research shows that in doing promotion the Victory Sorong University did 3 promotion strategies, that are advertisement, personal selling and publicity. Advertisement is the most affected and publicity is non-affected promotion in increasing the sum of student.

The dimension of advertisement is done by making billboard, banner, poster, brochure and tassle that they set on the strategic place. Such as in the edge of street, surrounding school environment like government

supported and private SMA/SMK and another busy place.

Advertisement can be done through media, such as:

- Electronic media (Television, radio, cinema, internet).
- Print media (newspaper, magazine, bulletin)
- Outdoor media (electronic board which is set in the edge of the street which moved electronically, billboard, banner, poster, ect).

The dimension of personal selling is done by visiting or socialization to some schools like goverment supported and private SMA/SMK. Usually, promotion team is consisted of 2 until 3 people who has a job to give information directly to the school and the third grade student who will join national final examination. The dimension of publicity is done by making a workshop or event in the collage and in the outside of collage then inviting reporter from print media to pervade the event. Besides of that the promotion team uses internet media through social network FB and website to share the picture or photos of the workshop and event which they have done to the entire user of another social network. This thing is expected to make good image of the collage in the society. The result of data analysis which is gotten from the research by using quantitative research method to answer the problem statement. First to see the effect, the whole result is as the follows:

TABLE II
THE RESULT OF PROMOTION USING RAPIDMINER

Index	Nilai Nominal	Frequent	Fraction
1	Advertisement	5778	0.662
2	Personal selling	1801	0.206
3	Publicity	1150	0.132

To get the whole information about the promotion strategy which is done by Victory Sorong University, choosing research information is done based on the research aim that is by using sample.

Data sample which is taken are 43 people on October 2017 as it seen in the table III.

TABLE III
THE RESULT OF PROMOTION STRATEGI IN THE FIELD

Index	Nilai Nominal	Frequent	Fraction
1	Advertisement	35	0.8140
2	Personal selling	6	0.1395
3	Publicity	2	0.0465

It shows that advertisement is the most affected promotion strategy toward the increasing sum of student in Victory Sorong University than using personal selling and publicity.

The effect of advertisement dimension is most significant. It can be seen that 35 transaction from the whole sample with significant 81.40% than another dimension. This research explains that advertisement is the most dominant variable in affecting the sum of student. In making advertisement, university uses the electronic media which consist of Television, radio, and internet, and also print media through banner, brochure and tassle.

The effect of Personal Selling dimension is quiet significant. It can be seen that the effect of 6 transaction is 13.95% than another dimension. The effect of publicity dimension is not significant. It can be seen that the value is 2 transactions with significant rate 4.65%.

The inhibiting factors of promotional activities are the funds promotional activities provided for operational activities felt not sufficient to carry out advertising promotion activities, personal selling, and publicity to the maximum in order to get results as expected.

V. SYSTEM IMPLEMENTATION

To prove the truth of the analysis results required a testing process to test the truth of the results of data processing done manually in chapter 3 that has been done before, for the testing process we can use one software application such as Rapidminer, with the steps as follows:

The dataset consists of 8729 data records and contains the variables attributes of public SMK, private SMK, public SMA, private high school, Madrasah Aliyah, Sorong, information systems, etc. are stored in Microsoft Excel with the new student candidate name 2008-2016. xls and which will be attempted using Rapidminer studio 6.4 software to see the same results as a frequent itemset search through:

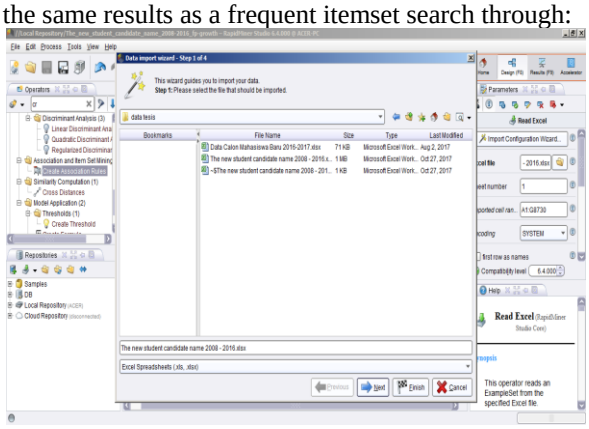


Fig 7: The appearance of excel sheet

Figure 7: is the process of data import of new student candidate.xls into rapidminer.

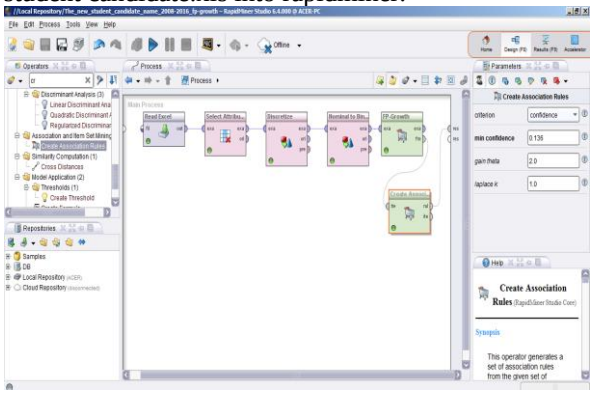


Fig 8: The process of connecting fp-growth to first race and create association rules into second races.

In the figure 8: it is explained about the process of connecting FP-Growth into first race and create association rules into second races and set into minimum support and minimum confidence.

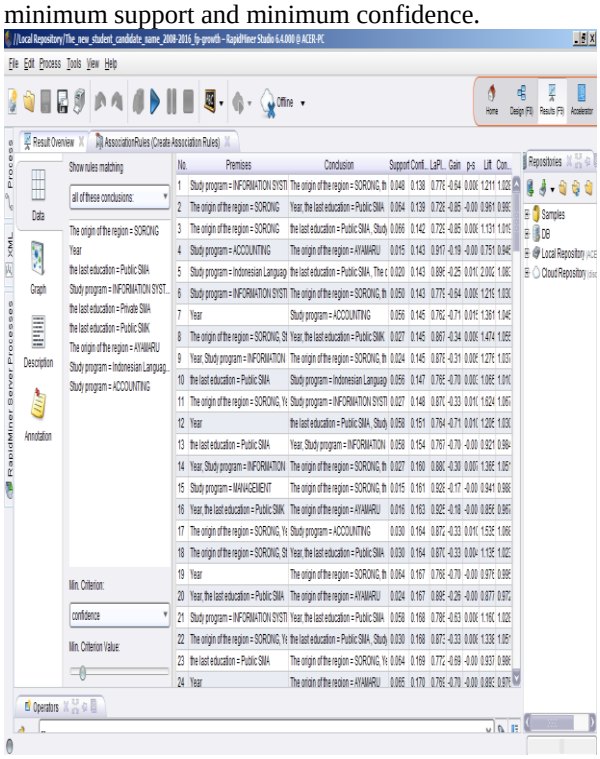


Fig 9: The appearance of the result of data processing use fp-growth

In figure 9: it is explained about the appearance of the result of data processing use FP-Growth and its relation to association rule.

Lift ratio is important parameter besides of support and confidence on association rule. Lift ration measures how important the rule that has been formed based on the value of support and confidence. Lift ratio is values that indicates the validity of the transaction process and provides information whether it is true that product A is purchased together with product B.

Improvement Ratio is obtained by the following formula:

$$\frac{Support\ (A \cap B)}{Support\ A \bullet Support\ B} = \qquad (3)$$

To find valid rule value is if have lift ration value >1 by way of support lift ration = support containing value A and value B divided support A * support B support value containing value A and value B is minimum result from minimum support divided with items (A3, B1, C9) {public SMA, Sorong, system information} At each occurrence transacted.

As for the results of the rule that most influence is: if the prospective student is from high school public school, Sorong address then he will choose the system of information systems with 100% confidence level and supported 47% of the overall data. The results of that rule will be targeted in promoting education.

VI. CONCLUSION AND SUGGESTION

A. *Conslusion*

After finishing the analysis and also the system implementation, it can be concluded as the follows:

1. As big as the sum of transaction which is processed so as the FP-Growth load process will be longer.
2. After doing research in the field, by using 43 transaction sample, it results: advertisement 81.40% or 35 transaction, personal selling 13.95% or 6 transaction and publicity 4.65% or 2 transaction. The suitable promotion strategy to being applied in Victory Sorong University is advertisement, Personal selling and publicity.
3. The obstacles which are found along promotion event is the minimum fund which is given to the whole promotion so that the promotion gives less maximal contribution toward the increasing sum of student.

B. *Suggestion*

Suggestions which can be noticed for the implementation of this technique are:

1. To get the good promotion result, it is needed more data sources and more complete data resource.
2. To implement rule which is resulted from FP-Growth algorithm with the management side can develop the strategy on education promotion so that the collage side will be more advance.

ACKNOWLEDGMENT

I hereby express my gratitude to the Victory Sorong University foundation that has provided funding and support during the course of this research.

REFERENCES

[1] Sujatha Dandu, B.L. Deekshatulu & Priti Chandra.“Improved Algorithm for Frequent Item sets Mining Based on Apriori and FP-Tree”. Global Journal of Computer Science and Technology Software & Data Engineering, Volume 13 Issue 2 Version 1.0 Year 2013, Type: Double Blind Peer Reviewed International Research Journal, Publisher: Global Journals Inc. (USA), Online ISSN: 0975-4172 & Print ISSN: 0975-4350.

[2] Abdullah Saad Almalaise Alghamdi. “Efficient Implementation of FP Growth Algorithm-Data Mining on Medical Data”. IJCSNS International Journal of Computer Science and Network Security, Vol.11 No.12, Dec 2011.

[3] Shivam Sidhu, Upendra Kumar Meena, Aditya Nawani, Himanshu Gupta, Narina Thakur. “FP Growth Algorithm Implementation”. International Journal of Computer Applications (0975 – 8887) Vol. 93 – No.8, May 2014.

[4] Bharat Gupta, Dr. Deepak Garg. “FP-Tree Based Algorithms Analysis: FPGrowth”. Bharat Gupta et al. / International Journal on Computer Science and Engineering (IJCSE). ISSN : 0975-3397 Vol. 3 No. 7 Jul 2011.

[5] Aroosa Hameed. “FP Growth Tree Implementation In Bank Transactional Databases”. ITEE Journal Information

Technology & Electrical Engineering, ISSN: - 2306-708X, Vol.5, Issue 3 Jun 2015.

[6] Smita R. Londhe, Rupali A. Mahajan, Bhagyashree J. Bhoyar. "Overview on Methods for Mining High Utility Itemset from Transactional Database". International Journal of Scientific Engineering and Research (IJSER),www.ijser.in, ISSN (Online): 2347-3878, Vol. 1 Issue 4, DEC 2013.

[7] Savi Gupta and Roopal Mamtora. "A Survey on Association Rule Mining in Market Basket Analysis". International Journal of Information and Computation Technology. ISSN 0974-2239 Vol. 4, No. 4 (2014), pp. 409-414 © International Research Publications House http://www. irphouse.com /ijict.htm.

[8] Loraine Charlet Annie M.C. and Ashok Kumar D. "Market Basket Analysis for a Supermarket based on Frequent Itemset Mining". IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, Sep 2012, ISSN (Online): 1694-0814. www.IJCSI.org.

[9] Ririanti. "Implementation of FP-GROWTH Algorithm on Motorcycle Prediction Application (Case Study of PT Pilar Deli Labumas)". Pelita Informatika Budi Darma, Vol : VI No: 1, Mar 2014. ISSN : 2301-9425.

[10] Ali Ikhwan, Dicky Nofriansyah, Sriani. "Implementation of Data Mining with Fp-Growth Algorithm to Support Education Promotion Strategy (Case Study Campus STMIK Triguna Dharma)". Jurnal Ilmiah SAINTIKOM, ISSN : 1978-6603, Vol.14, No.3, Sep 2015.

[11] Erwin. "Market Basket Analysis With Apriori Algorithm and FP-Growth". JURNAL GENERIC, ISSN 1907-4093, Vol. 4 No. 2, Jul 2009.

ISSN : 1978-6603

JURNAL Ilmiah SAINTIKOM Sains dan Komputer

Penerapan Data Mining dengan Algoritma Fp-Growth untuk Mendukung Strategi Promosi Pendidikan (Studi Kasus Kampus STMIK Triguna Dharma)

Ali Ikhwan[*1], Dicky Nofriansyah[#2], Sriani[#3]

[*1]Magister Ilmu Komputer Universitas Putra Indonesia"YPTK" Padang

[#2,3]Program Studi Sistem Komputer, STMIK TrigunaDharma

Jl. Raya Lubuk Begalung, Sumbar – Indonesia

Jl. A.H. Nasution No. 73 F-Medan

Email :Aliikhwan053@gmal.com

ABSTRAK

Saat ini semakin banyaknya jumlah kampus yang semakin berkembang yang membuat para pengelola ingin mendapatkan strategi promosi yang lebih baik. Adapun cara dalam menemukan strategi promosi yang tepat akan dapat mengurangi biaya promosi dan mencapai sasaran promosi yang tepat. Salah satu cara yang dapat dilakukan untuk penentuan strategi promosi adalah dengan menggunakan teknik data mining. Adapun teknik yang digunakan dalam hal ini adalah Algoritma FP-Growth. FP-Growth adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (frequent item set) dalam sekumpulan data. Algoritma FP-Growth merupakan pengembangan dari algoritma Apriori. Sedangkan di dalam algoritma FP-Growth tidak dilakukan generate candidate karena FP-Growth menggunakan konsep pembangunan tree dalam pencarian frequent itemset. Penelitian dilakukan dengan mengamati beberapa variabel penelitian yang sering dipertimbangkan oleh perguruan tinggi khusunya bagian marketing dalam menentukan sasaran promosinya, Yaitu Pendidikan Terakhir,Alamat Rumah,Jurusan,Pilihan Prodi. Hasil penelitian ini adalah berupa suatu perangkat lunak dengan mengimplementasikan algoritma FP-Growth yang menggunakan konsep pembangunan FP-Tree dalam mencari Frequent Itemset.

Kata Kunci : Data Mining,Association Rules,Frequent Itemset,FP-Growth.

ABSTRAC

Nowadays more and more growing amount of campus that make the managers want to get better promotion strategy. So how to find the right promotional strategy will be able to reduce costs and achieve targeted promotional campaign right. One of proces that can be done to determine promotion strategy is to use data mining techniques . The technique used in this case is the FP-Growth Algorithm. FP-Growth algorithm is one of the alternatives that can be used to determine the set of data that appears most frequently (frequent item sets) in a set of data. FP-Growth algorithm is an extension of Apriori algorithm. While in the FP-Growth algorithm do not generate candidate because the FP-Growth using the concept of development in the search for frequent itemset tree. The research doing by observing and some research variables are often considered by colleges especially the marketing in targeting the promotion, Namely Latest Education, Home Address, Subject, Options Prodi. The results of this research are in the form of a software to implement the FP-Growth algorithm that uses the concept of development of FP-Tree in finding Frequent Itemset.

Keywords: Data Mining,Association Rules,Frequent Itemset,FP-Growth.

1. PENDAAHULUAN

1.1 Latar Belakang Masalah

Persaingan di dalam dunia bisnis ini sangat ketat , khususnya dalam mempromosikan perguruan tinggi. Setiap kampus berusaha memberikan fasilitas yang terbaik. Oleh karena itu setiap kampus berusaha mencari strategi yang tepat untuk mempromosikan kampusnya. Menurut para pengembang untuk menemukan suatu strategi yang dapat menentukan strategi pemasaran dalam mempromosikan perguruan tinggi dengan memaksimalkan pelayanan kepada masyarakat. Salah satu teknik yang dipakai dalam penerapan *Data Mining* adalah di dalam bidang promosi. Bila sasaran promosi tidak ditentukan secara baik, dalam arti tidak diupayakan mencari sasaran promosi yang potensial, maka hanya akan menghabiskan banyak waktu dan biaya yang seharusnya bisa diminimalisir melalui pemilihan target promosi yang baik.

Salah satu cara yang dapat diterapkan adalah dengan menerapkan penggunaan *Data Mining*. Karena di dalam data mining sendiri terdapat cara dan teknik dalam pemenuhan kebutuhan salah satunya adalah kebutuhan informasi yang luas, dan dari informasi yang kita dapat bia kita gunakan sebagai suatu keputusan atau menentukan sebuah kualitas dalam menentukan sauatu keputusan. Adapun kumpulan dari sebuah data atau informasi memeiliki banyak potensi untuk dijadikan suatu kesimpulan dalam mengambil sebuah keputusan dengan melakukan analisis dan menggali suatu unformasi yang terdapat di dalam sebuah data. Maka dimungkinkan untuk dibuat strategi dalam mendukung promosi pendidikan. Dari definisi *Data Mining* yang luas terdapat banyak jenis teknik analisa yang dapat digolongkan dalam *Data Mining*. Teknik *Data Mining* yang akan digunakan dalam penelitian ini adalah *Association Rule Mining*. *Association rule mining* adalah teknik *mining* untuk menemukan aturan asosiatif antara suatu kombinasi *item*.

Adapun dalam penelitian ini akan dibahas bagaimana cara mengimplementasikan salah satu algoritma dalam data mining, yaitu algoritma *frequent Pattern-Growth (FP-Growth)*. Algoritma ini adalah bagian dari teknik asosiasi pada data mining. Adapun *FP-Growth* sendiri adalah salah satu altenatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering mucul *(frequent itemset)* dalam sekumpulan data. Karakteristik algoritma *FP-Growth* adalah struktur data yang digunakan dalam *tree* yang disebut *FP-Tree*. Dengan menggunakan *FP-Tree*, algoritma *FP-Growth* dapat langsung mengekstrak *frequent itemset* dari *FP-Tree.*

1.2 Perumusan Masalah

1. Bagaimana sistem memberikan informasi kepada user dalam mempromosikan pendidikan di Perguruan Tinggi Swasta secara efektif dan efisien?
2. Bagaimana sistem ini menentukan target dalam mempromosikan pendidikan?
3. Bagaimana menerapkan algoritma *FP-Growth* di dalam promosi pendidikan sehingga dalam proses tersebut dapat menentukan strategi yang tepat ?
4. Bagaimana melakukan pengujian Metode *FP-Growth* dengan *Rapidminer* sehingga menghasilkan penyelesain masalah dalam mencari strategi promosi yang tepat?

1.3 Batasan Masalah

penulis mencoba memberikan batasan masalah antara lain:

1. Algoritma *FP- Growth* yaitu algoritma yang digunakan di dalam penyeleksian untuk mempercepat proses penetuan *frequent itemset* sebelum meng-*generate rull* sebagai rekomendasi keputusan.
2. Penelitian ini khusus membahas tentang strategi promosi pendidikan pada Perguruan Tinggi khususnya di STMIK TRIGUNA DHARMA .
3. Data yang digunakan sebagai sumber merupakan data yang diambil di STMIK TRIGUNA DHARMA 1 tahun terakhir, yang meliputi asal sekolah, daerah asal, jurusan dan pilihan prodi.
4. *Tools* yang digunakan di dalam pengolahan data adalah *Rapidminer 5.2*

1.4 Tujuan Penelitian

1. Untuk menganalisa permasalahan promosi pendidikan di STMIK TRIGUNA DHARMA.
2. Untuk menerapkan metode *FP- Growth* di dalam strategi promosi pendidikan di STMIK TRIGUNA DHARMA.
3. Untuk melakukan pengujian metode *FP-Growth* dengan menggunakan tools *Rapidminer* sehingga menghasilkan strategi promosi pendidikan yang efektif dan efisien.
4. Menghasilkan sebuah hubungan antara variable atau item berdasarkan implementasi dari algoritma *FP-Growth.*
5. Menguji hubungan variable atau item tersebut untuk mendapatkan pengetahuan baru dari proses ekstraksi menggunakan *Tools* yang ada.

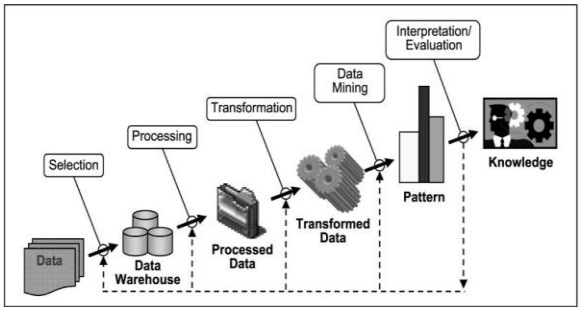
1.5 Manfaat Penelitian

1. Mengetahui algoritma *FP-Growth* dapat memberikan alternatif pengetahuan tentang menemukan strategi dalam promosi pendidikan.
2. Menemukan strategi atau sasaran yang lebih tepat dalam mempromosikan pendidikan.
3. Menjadikan hasil software ini dapat digunakan pihak kampus untuk menentukan strategi dalam promosi pendidikan.

2. LANDASAN TEORI

2.1 *Knowledge Discovery In Database* (KKD)

Knowledge Discovery in Database (KDD) adalah proses menentukan informasi yang berguna serta pola-pola yang ada dalam data. Informasi ini terkandung dalam basis data yang berukuran besar yang sebelumnya tidak diketahui dan potensial bermanfaat. *Data Mining* merupakan salah satu langkah dari serangkaian proses *iterative* KDD. Tahapan proses KDD dapat dilihat pada gambar 2.1.



Gambar 2.1 Tahapan dalam KDD

Tahapan proses KDD terdiri dari:

a. Data *Selection*

Pada proses ini dilakukan pemilihan himpunan data, menciptakan himpunan data target, atau memfokuskan pada subset

variable (sampel data) dimana penemuan (*discovery*) akan dilakukan. Hasil seleksi disimpan dalam suatu berkas yang terpisah dari basis data operasional.

b. Pre-*Processing* dan *Cleaning* Data

Pre-*Processing* dan *Cleaning* Data dilakukan membuang data yang tidak konsisten dan *noise*, duplikasi data, memperbaiki kesalahan data, dan bisa diperkaya dengan data *eksternal* yang relevan.

c. *Transformation*

Proses ini *mentransformasikan* atau menggabungkan data ke dalam yang lebih tepat untuk melakukan proses *mining* dengan cara melakukan peringkasan (*agregasi*).

d. *Data Mining*

Proses *Data Mining* yaitu proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik, metode atau algoritma tertentu sesuai dengan tujuan dari proses KDD secara keseluruhan.

e. *Interpretation* / Evaluasi

Proses untuk menerjemahkan pola-pola yang dihasilkan dari *Data Mining*. Mengevaluasi (menguji) apakah pola atau informasi yang ditemukan bersesuaian atau bertentangan dengan fakta atau hipotesa sebelumnya. Pengetahuan yang diperoleh dari pola-pola yang terbentuk dipresentasikan dalam bentuk visualisasi.

2.2 Data *Mining*

Data *Mining* adalah proses untuk mendapatkan informasi dengan melakukan pencarian pola dan relasi-relasi yang tersembunyi di dalam timbunan data yang banyak. (Fadlina, 2014) Data *Mining* atau sering disebut sebagai *knowledge discovery in database* (KDD) adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam data berukuran besar. Keluaran *Data Mining* ini bisa dipakai untuk membantu pengambilan keputusan di masa depan. Pengembangan KDD ini menyebabkan penggunaan *pattern recognition* semakin berkurang karena telah menjadi bagian *Data Mining*.

2.3 Tahapan dalam Data Mining

Menurut (Buulolo, 2013) ada empat tahap yang dilalui dalam data mining, antara lain:

1. Tahap pertama : *Pricise statement of the problem* (pernyataan tepat terhadap permasalahan)

Sebelum mengakses perangkat lunak data mining, seorang analis harus memiliki kejelasan perihal ‘pertanyaan apa yang akan ingin dijawabnya’. Jika tidak ada *formula* yang tepat untuk problematika yang ada maka anda hanya akan membuang-buang dan uang dalam membuat solusinya.

2. Tahap kedua : *Initial exploration*

Tahap ini dimulai dengan mempersiapkan data yang juga juga termasuk kedalam data mining “*cleaning*” (misalnya : mengedentifikasi dan menyikirkan data yang dikodekkan salah), transformasi data, memilih *subset record, data set,* langkah awal seleksi. Mendeskripsikan dan memvisualisasikan data adalah kunci dari tahap ini.

3. Tahap tiga : *Model building and validation*

Tahap ini melibatkan pertimbangan terhadap ragam permodelan dan memilih yang terbaik bagi performasi prediktif.

4. Tahap ke-empat : *Deployment*

Memilih aplikasi yang tepat berikut permodelan untuk membuat (*generate*) prediksi. Selanjutnya kita kan melihat rincian perihal tahapan-tahapan data mining.

2.4 Pengelompokan *Data Mining*

Data Mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu (Kusrini dan Emha Taufiq Luthfi, 2009):

1. Deskripsi
   Terkadang peneliti dan analis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menentukan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden.

2. Estimasi
   Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih kearah numerik daripada kearah kategori. Sebagai contoh akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya.

3. Prediksi
   Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang.
   Contoh prediksi bisnis dan penelitian adalah:
   a. Prediksi harga beras dalam tiga bulan yang akan datang.
   b. Prediksi persentasi kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikkan.
   Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

4. Klasifikasi
   Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang dan pendapatan rendah.
   Contoh lain klasifikasi dalam bisnis dan penelitian adalah:
   a. Menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang atau tidak.
   b. Memperkirakan apakah suatu pengajuan hipotek oleh nasabah merupakan suatu kredit yang baik atau buruk.
   c. Mendiagnosis penyakit seorang pasien untuk mendapatkan termasuk kategori penyakit apa.

5. Pengklusteran
   Pengklusteran merupakan pengelom-pokan *record*, pengamatan atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan.
   Contoh pengklusteran dalam bisnis dan penelitian adalah:

a. Mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari satu suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar.
b. Untuk tujuan audit akuntansi, yaitu melakukan pemisahan terhadap perilaku *financial* dalam baik dan mencurigakan.
c. Melakukan pengklusteran terhadap ekspresi dari *gen*, untuk mendapatkan kemiripan perilaku dari *gen* dalam jumlah besar.

6. Asosiasi
   Tugas asosiasi dalam *Data Mining* adalah menemukan *attribut* yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja. Contoh asosiasi dalam bisnis dan penelitian adalah:
   a. Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respon positif terhadap penawaran *upgrade* layanan yang diberikan.
   b. Menentukan barang dalam supermarket yang dibeli secara bersamaan dan yang tidak pernah dibeli secara bersamaan.

2.5 Tahapan Asosiasi

Analisis asosiasi dikenal juga sebagai salah satu teknik *Data Mining* yang menjadi dasar dari berbagai teknik *Data Mining* lainnya. Khususnya salah satu tahap dari analisis asosiasi yang disebut analisis pola frequensi tinggi *(frequent pattern mining)* menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien. Penting tidaknya suatu aturan assosiatif dapat diketahui dengan dua parameter, *support* (nilai penunjang) yaitu persentase kombinasi item tersebut dalam *database* dan *confidence* (Fadlina, 2014).

A. Data Preparation.

Pada banyak bidang keilmuan, terutama komputer sains, diperlukan data yang berkualitas melalui proses *preparation* dari data mentah. Dalam prakteknya, ditemukan bahwa data *cleaning* dan *preparation* membutuhkan total usaha 80% dari usaha untuk merekayasa data, sehingga menjadikan data *preparation* sebagai proses yang krusial. Pentingnya proses ini dapat dilihat dari tiga aspek, yakni:

1. Data *real world* merupakan data kotor. Data *real world* bisa mengandung data yang tidak, terdapat *noise*, tidak konsisten, yang dikarenakan:
   a. Tidak lengkap (*incomplete*), yaitu kekurangan nilai atribut atau hanya mengandung agregat data (contoh : address = " ")
   b. *Noise*, yaitu masih mengandung *error* dan *outliers*.
   c. Tidak konsisten (*inconsistent*), yaitu data yang mengandung discrepansi dalam code dan nama atau singkatnya datanya tidak konsisten.
2. Sistem *mining* dengan performa tinggi membutuhkan data yang berkualitas. Data *preparation* atau *preparation* menghasilkan dataset yang lebih sedikit daripada dataset yang asli, ini bias meningkatkan efisiensi dari *Data Mining*. Langkah ini mengandung:
   a. Memilih data yang relevan
   b. Mengurangi data.
3. Data yang berkualitas menghasilkan pola yang berkualitas. Dengan data *preparation*, maka data yang dihasilkan adalah data yang berkualitas, yang mengarah pada pola yang berkualitas pula, dengan:

a. Mengembalikan data yang tidak lengkap
b. Membenarkan *eror*, atau menghilangkan *outliers*
c. Membenahi data yang bertentangan.

B. Association Rules

Association rule merupakan salah satu metode yang bertujuan mencari pola yang sering muncul di antara banyak transaksi, dimana setiap transaksi terdiri dari beberapa *item* sehingga metode ini akan mendukung sistem rekomendasi melalui penemuan pola antar *item* dalam transaksi-transaksi yang terjadi
Metodologi dasar analisis asosiasi terbagi menjadi dua tahap:

1. Analisa pola frekuensi tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai support dalam *database*. Nilai *support* sebuah *item* diperoleh dengan rumus berikut:

$$Support\ (A) = \frac{\text{Jumlah Transaksi Mengandung A}}{\text{Total Transaksi}} \quad (1)$$

$$Support\ (A\ C\ B) = \frac{\text{Jumlah Transaksi Mengandung A dan B}}{\text{Total Transaksi}} \quad (2)$$

2. Pembentukan aturan assosiatif

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan assosiatif yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan assosiatif A _B Nilai *confidence* dari aturan A _B diperoleh dari rumus berikut:

C. Frequent Itemset

Langkah pertama pada *association rule* adalah menghasilkan semua *itemset* yang memungkinkan dengan kemungkinan *itemset* yang muncul dengan *m-item* adalah 2*m*. Karena besarnya komputasi untuk menghitung *frequent itemset*, yang membandingkan setiap kandidat *itemset* dengan setiap transaksi, maka ada beberapa pendekatan untuk mengurangi komputasi tersebut, salah satunya dengan algoritma apriori.

D. Apriori

Algoritma apriori digunakan untuk mencari *frequent itemset* yang memenuhi *minsup* kemudian mendapatkan *rule* yang memenuhi *minconf* dari *frequent itemset* tadi. Algoritma ini mengontrol berkembangnya kandidat *itemset* dari hasil *frequent itemset* dengan *support-based pruning* untuk menghilangkan *itemset* yang tidak menarik dengan menetapkan *minsup*. Prinsip dari apriori ini adalah bila *itemset* digolongkan sebagai *frequent itemset*, yang memiliki *support* lebih dari yang ditetapkan sebelumnya, maka semua subsetnya juga termasuk golongan *frequent itemset*, dan sebaliknya.

E. Rule Generation

Setelah mendapatkan *frequent itemset* menggunakan algoritma apriori, langkah selanjutnya adalah mendapatkan *rule* yang memenuhi *confidence*. Karena *rule* yang dihasilkan berasal dari *frequent itemset*, dengan kata lain, dalam menghitung *rule* menggunakan *confidence*, tidak perlu lagi menghitung *support*-nya karena semua calon *rules* yang dihasilkan telah memenuhi *minsup* sesuai yang ditentukan. Penghitungan ini juga tidak perlu melakukan perulangan *scanning* pada *database* untuk menghitung *confidence*, cukup dengan mengambil *itemset* dari hasil *support.*

2. 6 Algoritma *FP-Growth*

(Sensuse, 2012)*FP-growth* adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. *FP-growth* menggunakan pendekatan yang berbeda dari paradigma yang digunakan pada algoritma Apriori.

(Ririanti, 2014). *FP-Growth* adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent item set*) dalam sekumpulan data. Algoritma *FP-Growth* merupakan pengembangan dari algoritma *Apriori*.

FP-growth adalah metode yang sering *itemset* pertambangan tanpa Generasi calon. Ini membangun sebuah struktur data yang sangat padat *(FP-tree)* untuk kompres *database* transaksi asli.

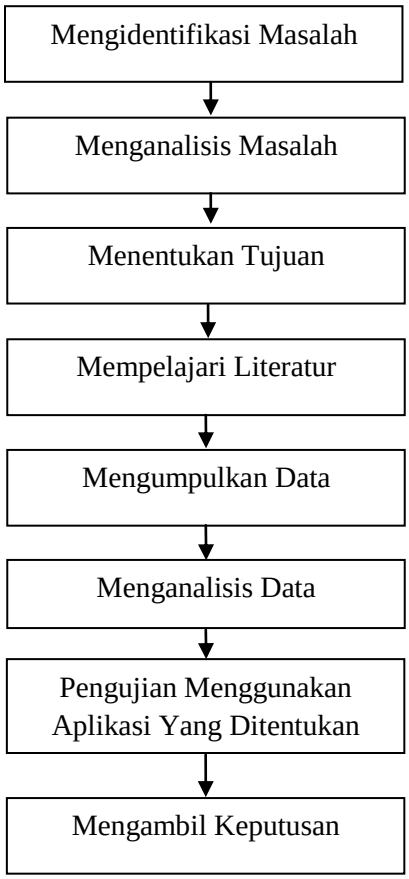
3. METODOLOGI PENELITIAN

3.1 Pendahuluan

Dalam melakukan penelitian agar hasilnya bisa maksimal, penulis mengikuti kaidah-kaidah (metode) yang telah ditetapkan. Metode penelitian yang akan diterapkan adalah metode penelitian ilmiah yang telah diajarkan di bangku perkuliahan sarjana. Metodologi penelitian ini memuat tentang kerangka kerja penelitian yang akan dibahas di bawah ini. Pada bab ini akan dibahas metodologi penelitian untuk mengidentifikasi permasalahan, analisa permasalahan dan pada akhirnya mencari penyelesaian masalah di dalam menganalisa strategi dalam mempromosikan pendidikan pada STMIK TRIGUNA DHARMA MEDAN.

3.2 Kerangka Kerja

Dalam metodologi penelitian ada urutan kerangka kerja yang harus diikuti, urutan kerangka kerja ini merupakan gambaran dari langkah–langkah yang harus dilalui agar penelitian ini bisa berjalan dengan baik. Kerangka kerja yang harus diikuti bisa dilihat pada gambar 3.1.



Gambar 3.1 Kerangka Kerja

Berdasarkan gambar 3.1 dijabarkan urutan–urutan langkah kerja sebagai berikut:

3.2.1. Mengidentifikasi Masalah

Masalah yang diidentifikasi dalam penelitian ini untuk mendapatkan strategi promosi yang efektif dan efisien dalam mendapatkan calon mahasiswa baru

ditengah ketatnya persaingan antar Perguruan Tinggi .

3.2.2. Menganalisis Masalah

Analisis masalah pada penelitian ini dilakukan dengan dua metode yaitu metode deskriptif dan metode komperatif.

a. Metode Deskriptif
   Pada metode ini data yang ada dikumpulkan, disusun, dikelompokkan dan dianalisis sehingga diperoleh beberapa gambaran yang jelas pada masalah yang akan dibahas.

b. Metode Komperatif
   Pada metode ini analisis dilakukan dengan cara membandingkan teori dan praktek sehingga diperoleh gambaran yang jelas tentang persamaan dan perbedaan di antara keduanya.

3.2.3. Menentukan Tujuan

Berdasarkan uraian di atas, tujuan dari penelitian ini adalah:

a. Untuk mengklasifikasikan setiap wilayah di Sumatera Utara dalam menentukan target promosi pendidikan di STMIK TRIGUNA DHARMA MEDAN.
b. Untuk mendapatkan atau menghasilkan suatu knowledge (pengetahuan) tentang wilayah yang akan dipilih dalam mempromosikan pendidikan.
c. Untuk memberikan masukan kepada pihak akademik sebagai bahan pertimbangan pihak akademik di dalam menentukan wilayah yang akan dijadikan target dalam promosikan pendidikan.

Mempelajari Literatur

Literatur-literatur yang dipakai sebagai bahan referensi dalam penelitian ini adalah dari jurnal-jurnal ilmiah, modul pembelajaran dan buku tentang *Data Mining*. literatur-literatur ini akan menjadi pedoman untuk melakukan penelitian agar memudahkan proses penelitian.

3.2.4. Mengumpulkan Data

Metode pengumpulan data dilakukan dengan cara melakukan pengamatan langsung ke kampus STMIK TRIGUNA DHARMA. Selain pengamatan, juga dilakukan wawancara kepada pihak-pihak yang terkait dengan penelitian ini. Di samping itu juga, melakukan pengambilan sampel *database* nasabah untuk menunjang penelitian ini.

3.2.5. Menganalisis Data

Pada tahap ini data yang telah dikumpulkan akan dianalisis. Analisis menggunakan *Association Rule* yang dimaksud dilakukan melalui mekanisme penghitungan *support* dan *confidence* dari suatu hubungan item. Sebuah rule asosiasi dikatakan *interesting* jika nilai *support* adalah lebih besar dari *minimum support* dan juga nilai *confidence* adalah lebih besar dari *minimum confidence* .Dengan menggunakan teknik *FP-Growth* yang menghasilkan *Frequent Itemset* tanpa melakukan *candidates generation*. dengan tujuan untuk mendapatkan pengetahuan yang baru (*knowledge*) berupa informasi tentang strategi promosi yang efektif dan efisien dalam mendapatkan calon mahasiswa baru ditengah ketatnya persaingan antar Perguruan Tinggi.

3.2.6. Pengujian Menggunakan Aplikasi Yang Ditentukan

Pada tahap ini, *rule* dites kembali atau diuji lagi menggunakan sistem *data*

mining yang sudah ada. *Tools* yang digunakan sebagai pengujian sistem adalah *Ramidminer 5. 2*.

Mula-mula data manual ”Data Mahasiswa Baru” yang sudah ada sebelumnya, di transformasikan ke dalam *Microsoft Excel*.

Sebagai tahap awal di dalam proses pengujian yaitu *mengimport* data mahasiswa baru pada kampus STMIK TRIGUNA DHARMA yang akan dijadikan sebagai data tabel transaksi *frequensi item* di dalam *tools Ramidminer 5 2.* tersebut. Kemudian tahap selanjutnya yaitu, melakukan proses *Data Input* dan *Affinity analysis* atau *market basket analysis*. Analisis asosiasi atau *association rule* adalah teknik *Data Mining* untuk menemukan aturan asosiatif antara suatu kombninasi *item*. Analasis *FP-Growth* didefenisikan suatu proses untuk menemukan semua aturan *FP-Growth* yang memenuhi syarat *minimum support* dan syarat minimum untuk *confidence*. Dari proses *Data Input* dan *Affinity* tersebut akan terdapat pola atau kriteria tertentu dalam mendapatkan strategi promosi. Hasil dari Proses *Data Input* dan *Affinity* tersebut akan diolah kembali untuk menghasilkan suatu *association rule*. Dari *association rule* tersebut akan terlihat hasilnya. Hasilnya tersebut merupakan pengetahuan baru *(knowledge)* dan juga merupakan sebagai acuan untuk menganalisa strategi dalam mempromosikan pendidikan.

3.2.7. Pengambilan Keputusan

Setelah diuji, hasil analisis antara *manual ways* (cara manual) dan pengujian memanfaatkan *tools* akan terlihat perbandingannya. Langkah berikutnya adalah penentuan atau pengambilan keputusan terhadap *knowledge* yang baru didapat yaitu berupa penentuan dalam menentukan strategi dalam mempromosikan pendidikan.

4. ANALISA

4.1 Analisa Data

Karakteristik algoritma *FP-Growth* adalah struktur data yang digunakan adalah *tree* yang disebut dengan *FP-Tree*. Dengan menggunakan *FP-Tree*, algoritma *FP-growth* dapat langsung mengekstrak *frequent Itemset* dari *FP-Tree.* Penggalian *itemset* yang *frequent* dengan menggunakan algoritma *FP-Growth* akan dilakukan dengan cara membangkitkan struktur data *tree* atau disebut dengan *FP-Tree*. Metode *FP-Growth* dapat dibagi menjadi 3 tahapan utama yaitu sebagai:

1. Tahap pembangkitan *conditional pattern base*,
2. Tahap pembangkitan *conditional FP-Tree*, dan
3. Tahap pencarian *frequent itemset*.

   Ketiga tahap tersebut merupakan langkah yang akan dilakukan untuk mendapat *frequent itemset*.

Input : *FP-Tree Tree*
Output : *Rt* Sekumpulan lengkap pola *frequent*
Methode : *FP-Growth (Tree, null)*
Procedure : *FP-Growth (Tree, α)*
{
01*: if Tree* mengandung *single path* P;
02*: then* untuk tiap kombinasi (dinotasikan β) dari *node-node* dalam *path do*
03: bangkitkan pola β α dengan *support* dari *node-node* dalam *path do* β;
04: *else u*ntuk tiap a1 dalam *header* dari *tree do*
}
05: bangkitkan pola

06: bangun β = a1 α dengan *support* = a1 *support*
07*: if Tree* β =

4.2 Pra Proses

Tabel 4.1 Untuk Memberi Tanda Pada Setiap *Itemset*

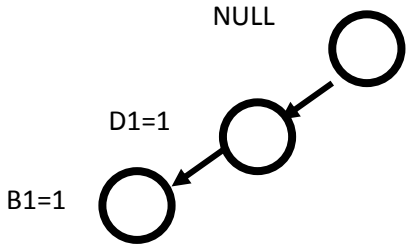
Item
SMK Negeri (A1)
SMK Swasta(A2)
SMA Negeri (A3)
SMA Swasta (A4)
Madrasah Aliyah (A5)
Medan (B1)
Aceh (B2)
Lubuk Pakam (B3)
Deli Serdang (B4)
Perbaungan (B5)
Tebing Tinggi (B6)
Riau (B7)
IPA (C1)
IPS (C2)
TKJ (C3)
MULTIMEDIA (C4)
Sistem Informasi (D1)
Sistem Komputer (D2)
Teknik Komputer (D3)
Manajemen Informatika (D4)

Tabel 4.2 Frekuensi dari Setiap Transaksi *Item*

TID	ITEM
1	D1,B1
2	C1,A3,B1
3	A3,C2
4	D1,A4,B1,C2
5	D1,C1,A4
6	C1,A3
7	D1,C1,A3
8	D1
9	D1,C1,A3
10	D1,C1,A4
11	D1,C1,A4
12	A3,B1.C2
13	D1,C1
14	C1
15	D1,C1,A3
16	C1,A3
17	D1,C1,A3
18	-
19	D1,A4,C2
20	D1,A4,B1,C2
21	A3,B1,C2
22	A3,B1,C2
23	D1,B1
24	D1
25	D1,C1,A4,B1
26	D1,C1,A4,B1
27	D1,C1,A4
28	D1,C1,A3
29	C2
30	D1,A4,C2

Maka langkah selanjutnya adalah membentuk pohon *FP-Tree* dengan melihat tabel 4.2 sebagai berikut:

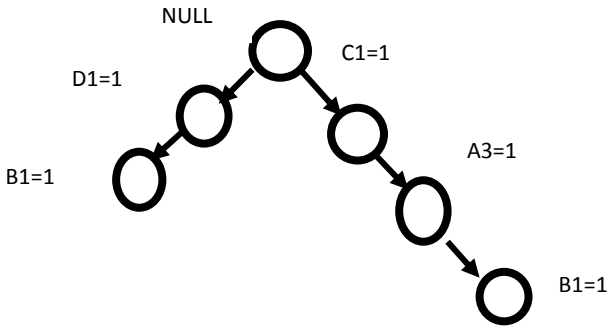
Gambar di bawah ini memberikan ilustrasi mengenai pembentukan *FP-tree* setelah pembacaan TID pada tabel 4.2



Gambar 4.1: Hasil Pembentukan *FP-tree* Setelah Pembacaan TID 1

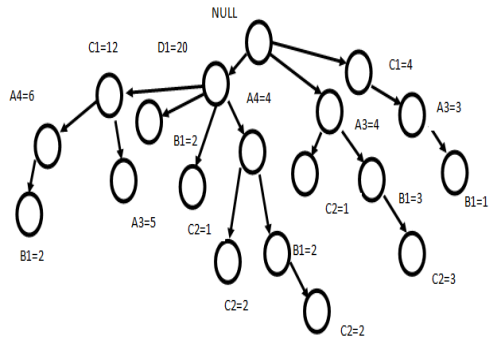
Gambar 4.1 adalah penjelasan tentang pembentukan *FP-Tree* setelah

pembacaan didapat setelah melakukan TID 1, Yaitu berisi : NULL- D1 (Sistem Informasi) =1 –(B1) Medan= 1.



Gambar 4.2 : Hasil Pembentukan *FP-tree* Setelah Pembacaan TID 2

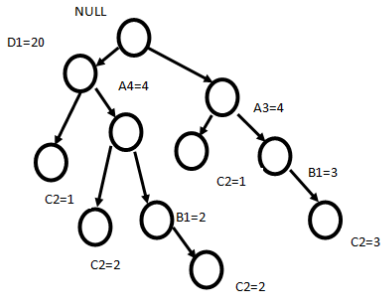
Gambar 4.2 didapat setelah melakukan TID 2 ,Yaitu: NULL-C1 (IPA) =1-A3 (SMAN) =1-B1(Medan) =1.



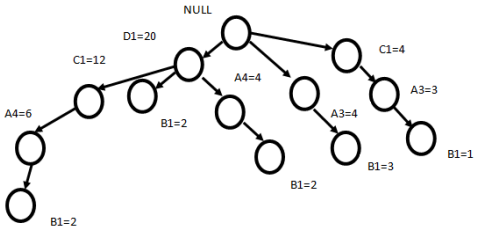
Gambar 4.3 : Hasil Pembentukan *FP-tree* Setelah Pembacaan TID 30

Gambar didapat setelah melakukan TID 30 yang dijumlahkan, yaitu berisi Null-Sistem Informasi (D1) = 20 -IPA (C1) =16 -SMA Negeri (A3) =12 -SMA Swasta (A4) =10-Medan (B1) = 10-IPS (C2) = 9.

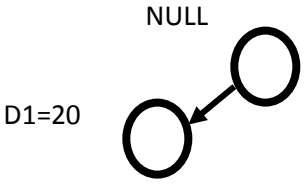
Untuk menemukan *Frequent itemset* dari tabel 4.2, maka perlu ditentukan terlebih dahulu lintasan yang berakhir dengan *support count* terkecil, yaitu C2 yang diikuti dengan A4,A3, C1,D1 dan di akhiri D1. Proses pembentukan masing-masing *node* dapat dilihat pada gambar 4.4.



Gambar 4.4 Lintasan yang Mengandung Simpul C2



Gambar 4.5 Lintasan yang Mengandung Simpul B1



Gambar 4.6 Lintasan yang Mengandung Simpul D1

Setelah mencari *frequent itemset* untuk beberapa akhiran *suffix* maka didapat hasil yang dirangkum dalam tabel berikut:

Tabel 4.3 Daftar *Frequent Itemset* Diurutkan Berdasarkan Hubungan Akhiran

Suffix	*Frequent Itemset*
C2	{C2},{C2,A4}{C2,A3}{C2,D1}{C2,B1,A3},{C2,B1,A4}
B1	{B1},{B1,D1},{B1,A3},{B1,A3,C1},(B1,A4,D1},{B1,A4,C1,D1}
A4	{A4},{A4,C1,D1}

A3	{A3},{A3,C1},{A3,C1,D1}
C1	{C1},{ C1,D1}
D1	{D1}

Dari *frequent itemset* yang didapat dari pembentikan *Fp-Tree* dan *FP-Growth* maka dapat dihitung nilai *Support* dan *Confidence* sebagai berikut:

Support = (Sistem Informasi,IPA,SMA Swasta,Medan) = *Count* (Sistem Informasi,IPA,SMA Swasta,Medan)/Jumlah Transaksi = 1/30.

Support = (Sistem Informasi,IPA,SMA Negeri) = *Count* (Sistem Informasi,IPA,SMA Negeri)/Jumlah Transaksi = 1/30.

Support = (Sistem Informasi, Medan) *Count* (Sistem Informasi, Medan)/Jumlah Transaksi =1/30.

Support = (Sistem Informasi,SMA Swasta, IPS) *Count*(Sistem Informasi,SMA Swasta, IPS)/Jumlah Transaksi = 1/30.

Support = (Sistem Informasi,SMA Swasta,Medan, IPS) *Count* (Sistem Informasi,SMA Swasta, Medan,IPS)/Jumlah Transaksi =1/30.

Support = (SMA Negeri ,IPS) *Count* (SMA Negeri ,IPS) / Jumlah Transaksi = 1/30.

Support = (SMA Negeri ,Medan,IPS) *Count* (SMA Negeri ,Medan,IPS) / Jumlah Transaksi = 1/30.

Support = (IPA,SMA Negeri ,Medan) *Count* (IPA,SMA Negeri ,Medan) / Jumlah Transaksi = 1/30.

Sedangkan untuk *confidence* atau nilai kepercayaannya adalah sebagai berikut :

Confidence = (SMA Negeri,IPS) = Count (SMA Negeri,IPS)/*Count* Medan= 1/9.

Confidence = (Sistem Informasi,SMA Swasta,Medan,IPS) = *Count* (Sistem Informasi,SMA Swasta,Medan,IPS /*Count* Medan = 1/9.

Confidence = (SMA Negeri,Medan,IPS) *Count* (SMA Negeri,Medan,IPS /*Count* Medan =1/9.

Confidence = (Sistem Informasi,SMA Swasta,IPS) *Count*(Sistem Informasi,SMA Swasta,IPS)/*Count* Medan= 1/19.

Confidence = (Sistem Informasi,SMA Swasta,Medan,IPS) *Count* (Sistem Informasi,SMA Swasta,Medan,IPS)/*Count* Medan =1/9.

Confidence = (SMA Negeri, Medan,IPS) *Count* (SMA Negeri, Medan,IPS) / *Count* Medan =1/9.

Confidence = (Sistem Informasi,IPS) *Count* (Sistem Informasi,IPS) / *Count* Medan =1/9.

Confidence = (Sistem Informasi,SMA Swasta,IPS) *Count* (Sistem Informasi,SMA Swasta,IPS) / *Count* Medan =1/9.

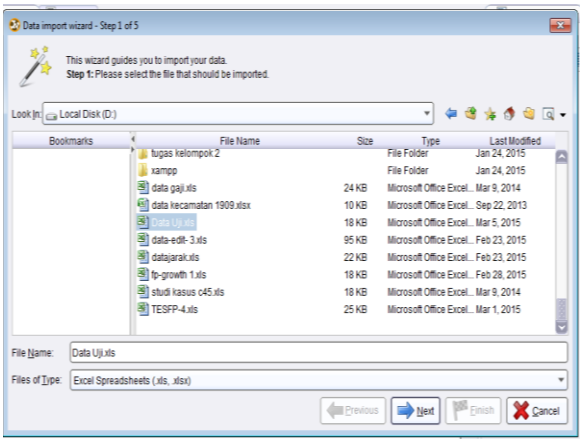
Setelah didapat nilai *support* dan *confidence* dari keseluruhan kombinasi pada data dengan perhitungan *FP-Tree* dan *FP-Growth* maka didapat niai *support* dan *confidence* yang paling tinggi dan akurat yaitu kombinasi. (D1,C1,A3) { Sistem Informasi, IPA, SMA Negeri } yang mempunyai nilai *support* : 5/30 = 0,16 dan nilai *confidence* : 5/20 = 0.25.

5. IMPLEMENTASI SISTEM

Untuk membuktikan kebenaran hasil analisa diperlukan sebuah proses pengujian untuk menguji kebenaran dari hasil pengolahan data yang dikerjakan secara manual pada BAB IV yang telah dikerjakan sebelumnya, untuk proses pengujian tersebut kita dapat menggunakan salah satu *software* aplikasi seperti *Rapidminer,* dengan langkah-langkah sebagai berikut:

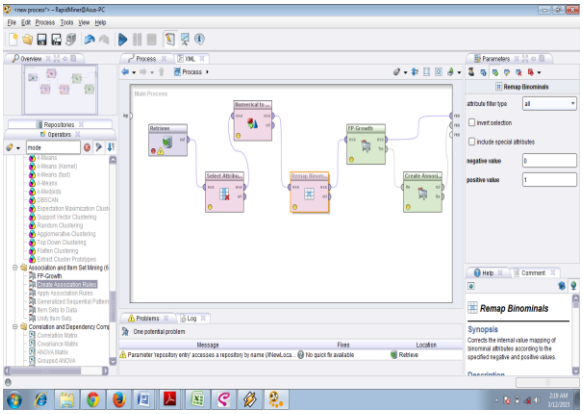
1. Dataset yang terdiri 30 record data dan berisi variabel-variabel atribut Sistem Informasi,IPA,SMA Negeri,SMA

Swasta,Medan, IPS disimpan dalam aplikasi *microsoft excel* dengan nama *file* datauji.xls dan yang akan dicoba menggunakan *software rapidminer* 5.2 untuk melihat hasil sama tidaknya dengan pencarian frequent itemsetnya melalu :



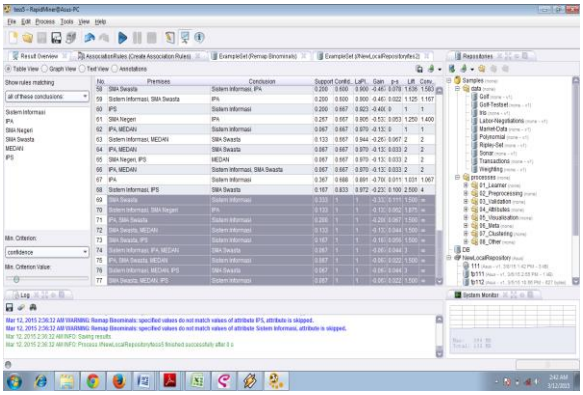
Gambar 5.1 Tampilan *Import Excel Sheet*

Gambar 5.1 adalah proses dari import data uji Xls ke dalam rapidminer



Gambar 5.2 Proses menghubungkan *FP-Growth* ke ras pertama dan *Create Association Rules* ke res kedua

Pada Gamabar 5.2 dijelaskan tentang proses menghubungkan *FP-Growth* ke ras Pertama dan *Create Association Rules* ke ras kedua dan diatur pada minimum *support* dan *minimum confidence*nya.



Gambar 5.20 *Rule valid* jika nilai *Lift Ratio* > 1

Lift Ratio adalah parameter penting selain *support* dan *confidence* dalam *association rule*. *Lift Ratio* mengukur seberapa penting *rule* yang teleh terbentuk berdasarkan nilai *support* dan *confidence*. *Lift Ratio* merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar produk A dibeli bersamaan dengan produk B.

Improvement Ratio dapat dihitung dengan rumus

$$\frac{Support(A \cap B)}{SupportA * SupportB} =$$

Untuk mencari nilai *rule* yang *valid* adalah jika mempunyai nilai *Lift Ratio* > 1 dengan cara *Support Lift Ratio* = *Support* yang mengandung nilai A dan nilai B dibagi *support* A* *Support* B nilai *support* yang mengandung nilai A dan nilai B Adalah hasil dari minimum *support* dibagi dengan *item* (D1,C1,A3) { Sistem Informasi, IPA, SMA Negeri }Pada setiap kemunculan ditransaksi.

=

$$\frac{5/30}{20/30*16/30*12/30} = \frac{0{,}16}{0{,}66*0{,}53*0{,}4} =$$

$$\frac{0{,}16}{0{,}1392} = 1{,}1494$$

Jadi dari hasil pencarian nilai *Lift Ratio* > 1 maka kita bisa menentukan sebuah *rule* yang *valid* dari sekian banyaknya rule yang dijalankan dengan *software Rapidminer*.

Adapun hasil dari rule yang paling mempangaruhi adalah:

Jika dia Asal sekolahnya SMA Swasta, Alamatnya Medan dan Asal Jurusanya IPS maka dia memilih Prodi Sistem Informasi dengan tingkat kepercayaan 100% dan didukung 6% dari data keseluruhan.

Hasil dari rule tersebut yang akan dijadikan target dalam mempromosikan pendidikan.

6. SIMPULAN DAN SARAN

6.1 Simpulan

1. Metode dalam pencarian *Frequent Itemsset* pohon keputusan menggunakan algoritma *FP-Ggrowth* bekerja sangat baik dalam melakukan *Frequent Itemsset* dengan proses pembentukan *FP-Tree* dengan menghasilkan *rule* dari data sampel mahasiswa baru.
2. Penentuan data variabel sangat menentukan tingkat akurasi *FP-Growth* yang dibuat dan besarnya presentase dalam menentukan minum *support* dan minimum *confidence* dipengaruhi oleh data variabel yang digunakan untuk mencari *frequent itemset* yang saling berhubungan untuk menemukan data variabel yang akan dijadikan stretegi dalam promosi pendidikan.
3. Dari penelitian yang dilakukan ada beberapa atribut yang tidak digunakan dalam *rule* yang dihasilkan, sehingga pemilihan atribut di dalam *dataset* sangat penting.
4. Algoritma *FP-Growth* dapat diterapkan untuk mendukung strategi promosi pendidikan pada Perguruan Tinggi. Adapun Informasi yang berkaitan dengan pelaksanaan promosi dapat tersedia dengan cepat, contohnya dalam pemilihan tempat yang akan dijadikan promosi pendidikan, sehingga pihak manajemen dari kampus sendiri dapat melakukan pengambilan keputusan dengan cepat.
5. Pelaksanaan promosi pendidikan pada perguruan tinggi sangat dibantu dengan adanya penerapan algoritma *FP-Growth* ini sehingga diharapkan efektifitas pelaksanaan promosi pendidikan akan dapat semakin ditingkatkan.
6. Untuk menemukan *rule* atau *knowledge* yang sangat berarti dalam menganalisa strategi dalam menentukan promosi pendidikan, dibutuhkan banyak data sampel yang digunakan untuk proses ekstraksi *rule* atau *knowledge* tersebut.

6.2 Saran

Salah satu tujuan penelitian ini adalah agar teknik dalam menentukan strategi promosi pendidikan di Kampus STMIK TRIGUNA DHARMA MEDAN dapat *diimplementasikan* atau diterapkan dengan baik, untuk itu saran yang dapat diperhatikan untuk *pengimplementasian* dari teknik ini adalah:

1. Pada penelitian ini, penulis hanya mencoba salah satu teknik dalam menemukan hubungan beberapa *item* yang digunakan untuk membentuk FP-*Tree* dalam menentukan strategi

promosi pendidikan dengan menggunakan Algorima *FP-Growth.* Untuk menentukan stategi dalam promosi pendidikan yang baik dapat digunakan beberapa atau penggabungan beberapa teknik yang akan digabungkan dalam sistem pengambilan keputusan untuk menentukan stategi yang tepat dalam menentukan stategi promosi pendidikan atau bisa juga menggunakan algoritma *CT-Pro.* Oleh sebab itu perlunya dilakukan penelitian lebih lanjut untuk mendapatkan strategi promosi dan menentukan teori mana yang menghasilkan strategi yang paling baik.

2. Untuk mendapatkan hasil promosi yang baik sangat diperlukan sumber data yang yang lebih banyak dan lebih lengkap.
3. Untuk mengimplementasikan *rule* yang dihasilkan dari teknik dari algoritma *FP-Growth* dapat berkonsultasi dengan tim marketing yang ada di kampus tersebut. Sehingga dengan adanya hubungan algoritma *FP-Growth* dengan pihak manajemen dapat mengembangkan strategi dalam promosi pendidikan sehingga pihak kampus akan lebih maju lagi kedepannya.

7. DAFTAR PUSTAKA

BUULOLO, E. 2013. Implementasi Algoritma Apriori Pada Sistem Persediaan Obat (Studi Kasus : Apotik

Rumah Sakit Estomihi Medan). *Pelita Informatika Budi Darma,* Vol : IV No: 1**,** 71-83.

FADLINA. 2014. Data Mining Untuk Analisa Tingkat Kejahatan Jalanan Dengan Algoritma Association Rule Metode Apriori (Studi Kasus Di Polsekta Medan Sunggal). *Informasi dan Teknologi Ilmiah (INTI),* Vol : III No : 1**,** 144-154.

RIRIANTI. 2014. Implementasi Algoritma FP-GROWTH Pada Aplikasi Prediksi Persedian Sepeda Motor (Studi Kasus PT. Pilar Deli Labumas). *Pelita Informatika Budi Darma,* Vol : VI No: 1.

SENSUSE, G. G. D. D. I. 2012. Penerapan Metode Data Mining Market Basket Analysis Terhadap Data Penjualan Produk Buku Dengan Menggunakan Algoritma Apriori dan Frequent Pattern Growth (FP-GROWTH) : Studi Kasus Percetakan PT. GRAMEDIA. *Jurnal TELEMATIKA MKOM,* Vol.4 No.1**,** 118-132.

KUSRINI dan EMHA TAUFIQ LUTHFI. 2009. “Algoritma Data Minning.” Jakarta : Andi. 10

Pattern Mining on Stars with FP-Growth

Andreia Silva and Cláudia Antunes

Instituto Superior Técnico
Av Rovisco Pais 1
1049-001 Lisboa
{andreia.silva,claudia.antunes}@ist.utl.pt

Abstract. Most existing data mining (DM) approaches look for patterns in a single table. Multi-relational DM approaches, on the other hand, look for patterns that involve multiple tables. In recent years, the most common DM techniques have been extended to the multi-relational case, but there are few dedicated to star schemas. These schemas are composed of a central fact table, linking a set of dimension tables, and joining all the tables before mining may not be a feasible solution. This work proposes a method for frequent pattern mining in a star schema based on FP-Growth. It does not materialize the entire join between the tables. Instead, it constructs an FP-Tree for each dimension and then combines them to form a super FP-Tree, that will serve as input to FP-Growth.

1 Introduction

While most existing data mining approaches look for patterns in a single data table, multi-relational data mining (MRDM) approaches look for patterns that involve multiple tables (relations) from a relational database or data warehouse. In recent years, the most common types of patterns and approaches considered in data mining have been extended to the multi-relational case and MRDM now encompasses multi-relational (MR) association rule discovery, MR decision trees and MR distance-based methods, among others. MRDM approaches have been successfully applied to a number of problems in a variety of areas[4].

From those works, just a few are dedicated to frequent itemset mining on star schemas [2,8,10].

This work aims to find frequent patterns in a set of tables of a data warehouse, following a star schema, without materializing the join of its tables.

A data warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process [7]. In terms of data modeling, a data warehouse consists of one or several dimensional models that are composed of a central fact table and a set of surrounding dimension tables, each corresponding to one of the dimensions of the fact table. The most used dimensional model is the star schema, which consists of multiple dimension tables that are associated by foreign keys to a central fact table.

At first glance, it may seem easy to join the tables of a star schema, and then do the mining process on the joined result[8]. However, when multiple tables

V. Torra, Y. Narukawa, and M. Daumas (Eds.): MDAI 2010, LNAI 6408, pp. 175–186, 2010.
© Springer-Verlag Berlin Heidelberg 2010

are joined, the resulting table will be much larger and the mining process more expensive and time consuming. There are two major problems: First, in large applications, often the join of all related tables cannot be realistically computed because of the distributed nature of data, large dimension tables and the many-to-many relationship blow up. Second, even if the join can be computed, the multifold increase in both size and dimensionality presents a huge overhead to the already expensive pattern mining process:

(1) the number of columns will be close to the sum of the number of columns in the individual tables.
(2) If the join result is stored on disk, the I/O cost will increase significantly for multiple scanning steps;
(3) For mining frequent itemsets of small sizes, a large portion of the I/O cost is wasted on reading the full records containing irrelevant dimensions;
(4) Each tuple in a dimension table will be read multiple times in one scan of the joined result. The number of times that a tuple appears in the fact table is the number of times the whole tuple will be read in the joined result.

One of the great potential benefits of MRDM is the ability to automate the mining process to a significant extent. Fulfilling this potential requires solving the significant efficiency problems that arise when attempting to do data mining directly from a relational database, instead of a single pre-extracted flat file[3].

The proposed algorithm is an adaptation of FP-Growth [5] to mine a star schema. The main idea is to adapt the construction of the FP-Tree, so that FP-Growth can run.

Like FP-Growth, it scans each table only twice: first to count the support of each item, and second to construct the FP-Tree. It is divided in 3 stages:

1. **Support Counting:** The fact table is scanned to count the support of each foreign key.
2. **Local Mining:** An FP-Tree is constructed for each dimension table (DimFP-Tree), with a slight modification of the original FP-Tree, taking into account the support calculated in the previous step.
3. **Global Mining:** The FP-Trees of each dimension are combined to form a Super FP-Tree, according to each fact and an established order of dimensions. This Super FP-Tree is then mined with FP-Growth, without a change, giving all the frequent patterns.

Several orders for the dimensions were studied and are presented and compared in this work.

The rest of this paper is organized as follows. Section 2 presents the related work on MRDM on star schemas. The proposed algorithm is described on section 3. Section 4 gives some experimental results and section 5 presents the conclusions.

2 Related Work

The work related to multi-relational pattern mining on star schemas is increasing. Experiments showed that the approach of “mining before join” outperforms the

approach of "join before mining" even when the latter adopts known to be fastest single-table mining algorithms[8].

Jensen and Soparkar (2000) presented an Apriori based algorithm [2], that first generates frequent itemsets in each single table using a slightly modified version of Apriori[1], and then looks for frequent itemsets whose items belong to distinct tables via a multi-dimensional count array. It does not construct the whole joined table but processes each row as the row is formed, thus storage cost for the joined table is avoided. However, the number of candidates generated explodes as the number of dimensions, attributes and values increase.

Ng et al.(2002) proposed an efficient algorithm without actually performing the join operation [8]. They perform local mining on each dimension table, and then "bind" two dimension tables at each iteration, i.e. mine all frequent itemsets with items from two different tables without joining them. After binding, those two tables are virtually combined into one, which will be "binded" to the next dimension table. They use vertical data format, and a prefix tree to compress the fact table and to speed up the calculation of support.

Xu and Xie (2006) proposed MultiClose [10], which discover frequent closed itemsets without materializing join tables. It first converts the dimension tables to vertical data format, and then mines each of them with a closed algorithm. After local mining, frequent closed itemsets are stored in two-level hash table result trees, and the frequent closed itemsets across two tables are discovered by traversing those result trees.

Several multi-relational methods have been developed by the Inductive Logic Programming community over the recent years, but they are usually not scalable with respect to the number of relations and attributes in the database. Therefore they are inefficient for databases with complex schemas. Another drawback of the ILP approaches is that they need the data in the form of prolog tables.

There are other algorithms for finding multi-relational frequent itemsets, however they just consider one common attribute at a time. They would have to run as much times as the number of dimensions, since there is no attribute common to all the tables. Instead, in a star schema, the fact table has one attribute in common with each dimension, and the dimensions have no common attribute between them. The patterns discovered by those algorithms will not reflect the relationships between dimensions.

The proposed algorithm also mines star schemas without computing the entire join nor materializing join tables. After constructing an FP-tree for each dimension, the corresponding tables are discarded. The trees already take into account the minimum frequency and incorporate transaction ids. The super FP-tree that represents the whole star is then constructed, by aggregating the dimension trees all together, based on the fact table. Finally, this tree in mined using the known pattern growth method, FP-Growth[5].

3 Mining Stars

Consider a relational database modeled as a star schema.

There are multiple dimension tables, which we will denote as A, B, C, ..., each containing only one primary key, denoted by transaction id (*tid*), some other attributes and no foreign keys. In fact, what we want is to discover potentially useful relationships among the attributes, other than primary keys. The set of values for an attribute is called the domain of the attribute.

In order to simplify our discussion we assume the fact table, denoted as FT, only contains the tids from dimension tables as foreign keys (tid_A, tid_B, tid_C, ...). If it contains some fields other than primary keys, we can place them into an extra dimension table and insert a new foreign key corresponding to it into the fact table. They are considered facts or measures.

As example, lets consider a dataset of the real movies database used in the experiments, only with three dimensions: Award (A), Studio (B) and Movie (C). Table 1 presents a conceptual representation of the dimension tables, where a_i, b_i and c_i denote the *tid* of dimension tables A, B and C, respectively, and x_i, y_i and z_i denote each possible value of A, B and C. Table 2a shows the fact table.

This example will be used to show how the proposed algorithm works, with a minimum support equals to 40% of the database. The sample of the database has

Table 1. Dimension Tables

(a) Table A

tid_A	**Itemsets**	**Support**
a_1	x_1	3
a_2	$x_2x_3x_4$	1
a_3	$x_5x_6x_7$	5
a_4	$x_8x_6x_7$	1

(b) Table B

tid_B	**Itemsets**	**Support**
b_1	y_1	2
b_2	$y_2y_3y_4$	1
b_3	$y_5y_3y_6$	1
b_4	$y_7y_3y_6$	2
b_5	y_8y_3	1
b_6	$y_9y_3y_6$	2
b_7	$y_{10}y_{11}y_6$	1

(c) Table C

tid_C	**Itemsets**	**Support**
c_1	z_1z_2	1
c_2	z_3z_4	1
c_3	z_5z_4	1
c_4	z_6z_4	1
c_5	z_7	1
c_6	z_8z_2	1
c_7	z_9z_4	1
c_8	$z_{10}z_{11}$	1
c_9	$z_{12}z_4$	1
c_{10}	$z_{13}z_4$	1

Table 2. Fact table and the frequent itemsets corresponding to each *tid*

(a) Fact Table

tid_A	tid_B	tid_C
a_3	b_2	c_1
a_3	b_4	c_2
a_3	b_6	c_3
a_3	b_4	c_4
a_1	b_1	c_5
a_1	b_5	c_6
a_3	b_7	c_7
a_1	b_1	c_8
a_2	b_3	c_9
a_4	b_6	c_{10}

(b) Denormalized Fact

$Itemsets_A$	$Itemsets_B$	$Itemsets_C$
$x_6x_7x_5$	y_3	–
$x_6x_7x_5$	y_3y_6	z_4
$x_6x_7x_5$	y_3y_6	z_4
$x_6x_7x_5$	y_3y_6	z_4
–	–	–
–	y_3	–
$x_6x_7x_5$	y_6	z_4
–	–	–
–	y_3y_6	z_4
x_6x_7	y_3y_6	z_4

10 transactions, therefore 40% of it corresponds to 4 transactions. This means that an itemset is frequent if its support is no less than 4 transactions.

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of distinct literals, called items. A subset of items is denoted as an itemset. A transaction $T = (t_{id}, X)$ is a tuple where t_{id} is a transaction-id and X is an itemset in I. Each table, in a relational database D, is a set of transactions. The *support* (or occurrence frequency) of an itemset It, is the number of transactions containing It in the database. It is frequent if its support is no less than a predefined minimum support threshold, σ.

In a database modeled as a star schema, where there are several tables, we have to be more specific: the *local support* of an itemset It, with items belonging to a table A (A.localSup(It)), is the number of occurrences of It in A. For example, on table 1a, A.localSup(x_1) = 1 and A.localSup(x_6) = 2. The *global support* (or just *support*) of an itemset It is the number of transactions of the fact table containing all the *tid*s that contain It, as in equation 1:

$$globalSup(It) = \sum_{tid}^{tid(It)} FT.localSup(tid) \tag{1}$$

Following the example above, globalSup(x_1) = FT.localSup(a_1) = 3 and globalSup(x_6) = FT.localSup(a_3) +FT.localSup(a_4) = 5 + 1 = 6.

3.1 The Algorithm

Star FP-Growth mines multiple relations for frequent patterns in a database following a star schema. The result is the same as mining the joined table, but without materializing it.

It is based on FP-Growth [5], and the main idea is to construct a Super FP-Tree, combining the FP-Trees of each dimension, so that the original FP-Growth can run and find multi relational patterns. Like FP-Growth, it scans each table only twice: first to count the support of each item and second to construct the FP-Tree.

The overall steps are:

Step 1: Support Counting: The fact table is scanned to count the support of each *tid* of each dimension. In the example, the *tid* support is shown in the third column of each dimension table (Table 1).

Step 2: Local Mining: An FP-Tree is constructed for each dimension table (DimFP-Tree), with a slight modification of the original FP-Tree, taking into account the support calculated in the previous step.

Step 3: Global Mining:

Step 3.1: Construct the Super FP-Tree: The DimFP-Trees of each dimension are combined to form a Super FP-Tree, according to each fact and an established order among dimensions.

Step 3.2: Mining the Super FP-Tree: Run FP-Growth [5], without a change, with the Super FP-Tree and the minimum support threshold. The result of this step is a list of all patterns, not just those relating to one dimension, but also those which relate the various dimensions.

Constructing the DimFP-Tree. A DimFP-Tree is very similar to an FP-Tree[5]. There are two major differences between the construction of a DimFP-Tree and an FP-Tree:

First, the support used here is the global support of each item, i.e. we consider the occurrences of an item in all database, not only in the item's table. Therefore, a node does not start with the support equals to one, but with support = support(T), with T the *tid* of the transaction that originated the node. It also is not incremented by only one, but by support(T).

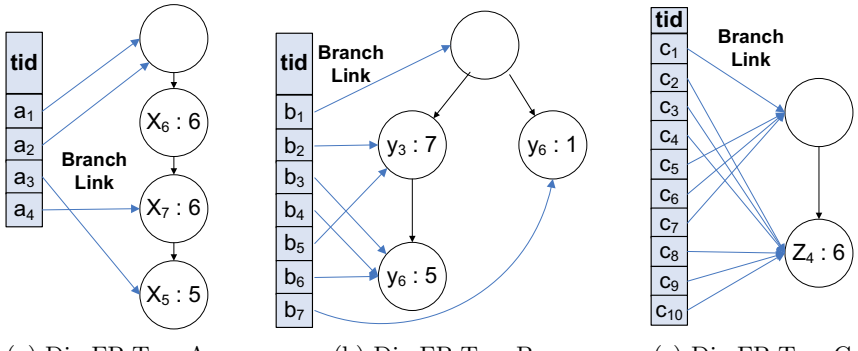
For example, b_4 has a support = 2, which means that that transaction occurs two times in the database. Adding two times the same transaction with support = 1 is the same as adding it one time with support = 2. Starting a node with support = support(T) and incrementing by support(T) avoids being repeatedly inserting the same transaction.

Second, instead of the header table, the DimFP-Tree has other structure, the branch table, that keeps track of the path correspondent to each *tid*. It stores the last node of that path for each *tid*. This structure will help the global mining. If we want to know which frequent items belong to a *tid*, we follow the link in that table to find the last node, and then we just have to climb through its parents till we reach the root node. The items of the nodes in the path we took are the frequent items of that transaction.

The result is the same as if we have the table with the transactions and their frequent sorted items. However, the size of the tree is usually much smaller than its original database, therefore, not having that table materialized in memory usually saves a lot of space and avoids duplicates [6]. If we keep the table instead of the tree, and if two transactions have the same frequent items, those items will be repeated two times. With the tree, there is just one path corresponding to the two transactions. Further, shared parts can also be merged using the tree. Therefore, in a larger scale, the more transactions there are, the greater the difference.

Lets consider the construction of the DimFP-Tree (figure 1b) of table B:

Step 2 starts with a first scan to the table B to calculate the support of each item. Only y_3 and y_6 are frequent, i.e. have a support no less than four transactions ($sup(y_3) = 7$ and $sup(y_6) = 6$). Therefore, only the itemsets containing just y_3



(a) DimFP-Tree A (b) DimFP-Tree B (c) DimFP-Tree C

Fig. 1. DimFP-Trees for each dimension table

and/or y_6 would be frequent, according to the anti-monotone property[1]. For each transaction $b_1, b_2, \ldots, b_7$, frequent items are selected and sorted according to the support descending order, and then inserted in the tree. At the same time, the branch table is constructed, linking each *tid* to the respective node in the tree. b_1, for example, does not have any frequent item, therefore its branch link links to the root node. b_3 corresponds to the first path of the tree, therefore its branch link points to the last node in that path. The other transactions follow the same reasoning. Figure 1 shows the results for all dimensions.

Constructing the Super FP-Tree. The Super FP-Tree is just like an FP-Tree, since it will serve as input to FP-Growth. The construction is very similar to the construction of an FP-Tree [5]. Despite this, there are three differences:

First, it is not necessary the first scan to any table to calculate the supports. They are already calculated and stored in each dimension tree.

Second, a fact is a set of *tid*s, therefore the denormalization of each fact is necessary before ordering the items or inserting them in the tree. A denormalized fact is an itemset with the items corresponding to its *tid*s. Through the branch table of each DimFP-Tree we can get the path corresponding to the itemset of each *tid*. Furthermore, according to the anti-monotone property, if an itemset is not frequent, no other itemset containing it will be. Thus, we only check the frequent items in the transactions of each *tid*, ensuring that the final tree has only the frequent items of each dimension. Table 2b, one can see the result of denormalizing each fact of our example.

And third, the ordering of items in a transaction does not have to be the frequency descending order. As verified in the improvement of FP-Growth proposed in [6], an FP-tree based on frequency descending ordering may not always be minimal.

The support descending ordering enhances the compactness of the FP-tree structure. However, this does not mean that the tree so constructed always achieves the maximal compactness. With the knowledge of particular data characteristics, it is sometimes possible to achieve even better compression.

There are two related and important properties of the FP-tree that can be derived from its construction process [6].

On one hand, given a transaction database DB, and without considering the root, the size of an FP-tree is bounded by $\sum_{T \in DB} |freq(T)|$.

The height of the tree is bounded by $max_{T \in DB}\{|freq(T)|\}$, where $freq(T)$ gives the frequent items of transaction T.

This means that the number of nodes of an FP-Tree (size) is, at most, the number of frequent items in all the transactions, and the number of levels (height) is, at most, the maximal number of frequent items in a transaction.

On the other hand, given a transaction database DB, the number of paths in an FP-tree is bounded by $|DB|$, i.e., it is, at most, the number of transactions in the database, if each transaction contributes to one different path of the FP-tree, with the length equal to the number of frequent items in that transaction.

With those lemmas in mind, several orders among dimensions were studied and compared in terms of the properties defined above. The three most relevant are the following:

1. Support descending order of items

This ordering does not have into account any order of dimensions. After denormalizing the fact, the transaction may have items from multiple dimensions. Sorting them in a support descending order may result in an itemset with items from multiple dimensions intermixed. This is the order used in the original FP-Growth. So, the tree resulting from applying this ordering is the same as the tree resulting from joining the tables in one and applying directly FP-Growth to it (but in this case, the joining is not materialized).

2. Support descending order of dimensions

As the support descending ordering enhances the compactness of the FP-tree structure, it is a promising order for the dimensions. Dimensions with higher support are more likely to be shared and thus arranged closer to the top of the FP-tree.

Since each dimension can have multiple items with different supports, we consider that the dimension support corresponds to the support of its least frequent item. Dimensions with the same support are ordered alphabetically, and the items of a dimension are also ordered in a support descending order.

Note that, with this ordering, items from multiple dimensions are not intermixed. Items from dimensions with higher support will always appear before those from dimensions with lower support.

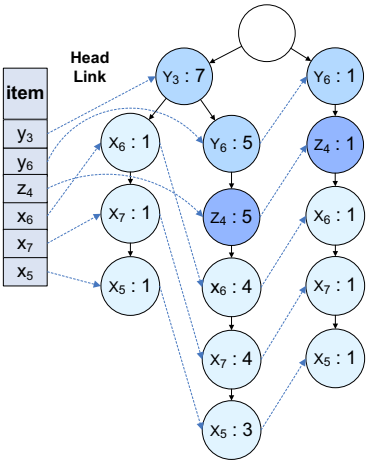


Fig. 2. Super FP-Tree with a support descending order of dimensions

In our example, the lowest support in dimensions is five in dimension A, and six in B and C. The support descending order of these dimensions is $B \rightarrow C \rightarrow A$ $((y_i s) \rightarrow (z_i s) \rightarrow (x_i s))$. Figure 2 presents the resulting Super FP-Tree. The branch table shows the items according to this ordering.

3. Path ascending order of dimensions

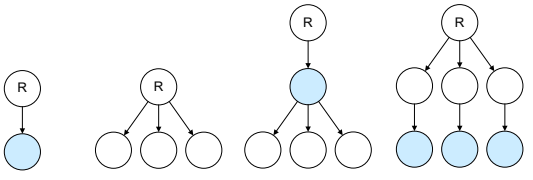
If we look at the number of paths of a tree, $|paths|$ (or just P), we can state that, when joining 2 or more trees of different $|paths|$ (i.e. adding one tree to every leaf of the other), the order in which the trees are joint influences the size of the resulting tree. Note that the $|paths|$ of the joined tree is always the same and equals to $\prod_{t \in TS} |P(t)|$, where TS is the set of the trees we want to join. The number of paths of a tree is the same as the number of leafs.

Its size (without the root) is given by

$$\sum_{i=1}^{|TS|} \left(\prod_{j=1}^{i-1} P(t_j) \right) \times size(t_i) \tag{2}$$

where $P(t)$ gives the number of paths in the tree t and TS is the set of trees in the order they are joint. The explanation is the following: a tree is inserted in each leaf of the tree immediately above, which in turn, was also inserted in each leaf of the preceding tree.

For example, imagine we have a tree A with 1 path, and another, B, with 3 paths (Figure 3a and 3b). If we join A with B, a copy of B is inserted in each leaf of tree A. Therefore, the resulting tree will have 4 nodes (without the root), as shown in figure 3c. Joining B with A will result in a tree with more nodes, 6 (figure 3d). This is because, the more leafs the tree above had, more copies of the tree below are needed.



(a) Tree A (b) Tree B (c) A with B (d) B with A

Fig. 3. Joining two trees

As we stated, joining trees in a path ascending order, will result in the smallest tree. However, joining two DimFP-Trees is not that linear. We may have to insert one tree not just in the leafs of the other, but also in a middle node, if that node corresponds to the last node of one transaction. Expressions above are also valid for this case, if we consider that $P(t)$ gives the number of nodes that correspond to the last node of a transaction.

Summing, this ordering consists in joining DimFP-Trees in a path ascending order. Dimensions with the same number of paths are ordered alphabetically, and the items of one dimension are ordered in a support descending order, like the other orderings.

These orderings have been applied to the movies database and the results are presented in section 4. Note that the set of frequent items is independent of the order applied. The result is the same for every orderings.

4 Experimental Results

The Super FP-Tree is the main structure of this algorithm. This is the tree that will be mined with FP-Growth, and this is the tree that holds the patterns we want to find. Therefore, the Super FP-Tree is the central object of these experiments.

Our goal is to analyze the impact of different orderings on the performance of our algorithm. In order to do that, the three orderings for the construction of the Super FP-Tree are compared, varying the minimum support threshold and the time spent in each step is analyzed.

The dataset is a real movies database[9]. The real database has six dimensions, with different numbers of records, from about 20 to 11000, and with a fact table with about 11000 transactions.

Among the data, we encounter a description of the directors, producers and awards received for each film and time information about them.

To achieve reliable results, the data was split into five equal datasets. The tests were applied to each dataset and we considered the average of each local result. Therefore, we analyze about 2000 facts at each time.

When comparing the size of the Super FP-Tree (figure 4), i.e. the number of nodes, the tree that is more compact is the one resulting from applying some order to the dimensions. In terms of compression, the support descending and path ascending orders of dimensions are very similar, and better than the support descending order of items. In this experiments, this ordering gave always the tree with more nodes. This difference happens because assigning an ordering for the dimensions, taking into account their characteristics and the properties described above, increase the number of shared nodes, and therefore, the compactness of the Super FP-Tree.

Note that the support descending order of items gives the same results as constructing an FP-Tree from the flat table (resulting from the join of the star), therefore, it serves as a reference to the other orderings.

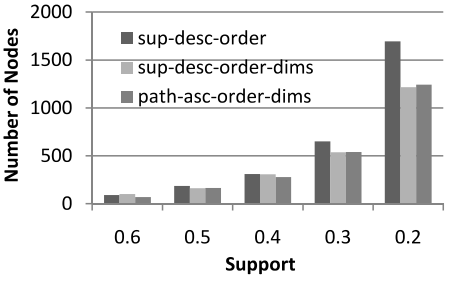


Fig. 4. Average size of the Super FP-Tree

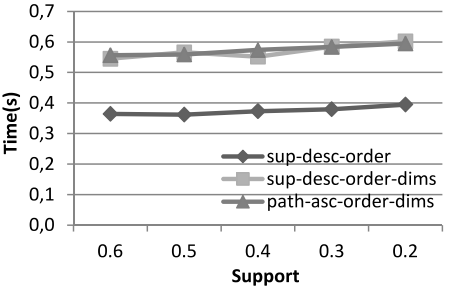


Fig. 5. Average time on Super FP-Tree construction

In terms of the number of paths in the Super FP-Tree, the resulting trees are very similar. Although the support descending order of items gives the less compact tree, it gives a tree with slightly less paths than the other orders.

The average time spent in the mining process was also studied. On average, 98% of the time is spent in the construction of the DimFP-Trees (step 2), and counting the global support (step 1) only takes 0,20% of the time. The difference between the application of the three orderings is mostly seen in step 3. As can be seen in figure 5, the support descending order of items takes less time constructing the Super FP-Tree than the other orders. With the support descending and the path ascending orders of dimensions, each transaction of the fact table has to be ordered according to that ordering before it can be inserted in the tree, yielding the previous results.

Even though the time for the construction of the Super FP-Tree is smaller for the support descending order of items, the FP-Growth will take longer to

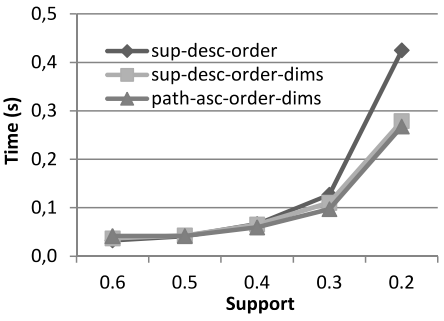


Fig. 6. Average time of FP-Growth

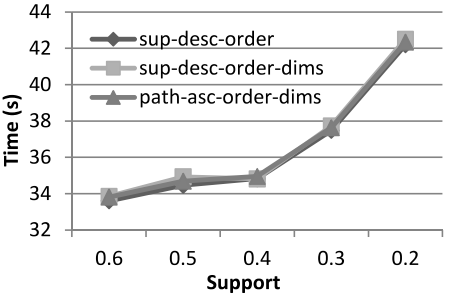


Fig. 7. Average total time on Star FP-Growth

execute (figure 6), due to resulting tree's size (figure 4 shows its size is bigger). Therefore, in the end, the total time needed for running Star FP-Growth (figure 7) is very similar for all orderings.

The size of the trees, as well as the time spent, depend not just on the size and characteristics of the data, but also on what the user wants. As minimum support decreases, the number of patterns increases, and therefore the tree and time will also increase. However, the memory needed to keep the trees will generally be much less than the memory needed to keep the tables.

The computer used to run the experiments was an Intel Xeon E5310 1.60GHz (Quad Core), with 2GB of RAM. The operating system used was GNU/Linux amd64 and the algorithm was implemented using the Java Programming language (Java Virtual Machine version 1.6.0_02). The tables were maintained in memory, as well as all the trees. However, the dimension tables were freed before Global Mining, as well as the fact table before running FP-Growth.

5 Conclusions

Star FP-Growth is a simple algorithm for mining patterns in a star schema. It does not perform the join of the tables, making use of the star properties. Building a tree for each dimension taking into account the global support of items allows us to discard the respective table and to keep only the frequent items. The main purpose is to prepare the FP-Tree that represents the data, combining the dimension trees, so that it can serve as input to FP-Growth.

Three orderings for dimensions were analyzed and the results state that applying a support descending or a path ascending order for the dimensions achieve better compression than the usual support descending order of items. The time spent in the mining process was very similar for both orderings, but it actually depends on the size and characteristics of the data, and on what we want: minimum support, performance, memory.

Using a pattern growth method and the FP-Tree gives us an important benefit: the size of an FP-tree is bounded by the size of its corresponding database

because each transaction will contribute at most one path to the FP-tree, with the length equal to the number of frequent items in that transaction. Since there are often a lot of sharing of frequent items among transactions, the size of the tree is usually much smaller than its original database. Unlike the Apriori-like method which may generate an exponential number of candidates in the worst case, under no circumstances, may an FP-tree with an exponential number of nodes be generated.

If the tree cannot be maintained in main memory, several techniques can be used, whether representing and storing the tree in hard disk, or partitioning the database into a set of projected databases, and then for each projected database, constructing and mining its corresponding FP-tree [6].

The proposed algorithm can also be generalized to be applied to a snowflake structure, where there is a star structure with a fact table FT, but a dimension table can be replaced by another fact table FT', which is connected to a set of other dimension tables. We can consider mining across dimension tables related by FT' first. Then consider the resulting Super FP-Tree as a derived DimFP-Tree and continue processing the star structure with FT. This means that mining a snowflake starts from their "leaves".

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB 1994, Proceedings of 20th International Conference on Very Large Data Bases, pp. 487–499 (September 1994)
2. Crestana-Jensen, V., Soparkar, N.: Frequent itemset counting across multiple tables. In: Terano, T., Chen, A.L.P. (eds.) PAKDD 2000. LNCS, vol. 1805, pp. 49–61. Springer, Heidelberg (2000)
3. Domingos, P.: Prospects and challenges for multi-relational data mining. SIGKDD Explor. Newsl. 5(1), 80–83 (2003)
4. Džeroski, S.: Multi-relational data mining: an introduction. SIGKDD Explor. Newsl. 5(1), 1–16 (2003)
5. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: SIGMOD 2000: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 1–12. ACM, New York (2000)
6. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining and Knowledge Discovery 8(1), 53–87 (2004)
7. Inmon, W.H.: Building the data warehouse, 2nd edn. John Wiley & Sons, Inc., New York (1996)
8. Ng, E.K.K., Fu, A.W.-C., Wang, K.: Mining association rules from stars. In: Proceedings of the 2002 IEEE International Conference on Data Mining, pp. 322–329 (2002)
9. Wiederhold, G.: Movies database documentation (1989)
10. Xu, L.-J., Xie, K.-L.: A novel algorithm for frequent itemset mining in data warehouses. Journal of Zhejiang University - Science A 7(2), 216–224 (2006)

ISSN : 1978-6603

JURNAL Ilmiah SAINTIKOM Sains dan Komputer

Penerapan Data Mining dengan Algoritma Fp-Growth untuk Mendukung Strategi Promosi Pendidikan (Studi Kasus Kampus STMIK Triguna Dharma)

Ali Ikhwan[*1], Dicky Nofriansyah[#2], Sriani[#3]

[*1]Magister Ilmu Komputer Universitas Putra Indonesia"YPTK" Padang

[#2,3]Program Studi Sistem Komputer, STMIK TrigunaDharma

Jl. Raya Lubuk Begalung, Sumbar – Indonesia

Jl. A.H. Nasution No. 73 F-Medan

Email :Aliikhwan053@gmal.com

ABSTRAK

Saat ini semakin banyaknya jumlah kampus yang semakin berkembang yang membuat para pengelola ingin mendapatkan strategi promosi yang lebih baik. Adapun cara dalam menemukan strategi promosi yang tepat akan dapat mengurangi biaya promosi dan mencapai sasaran promosi yang tepat. Salah satu cara yang dapat dilakukan untuk penentuan strategi promosi adalah dengan menggunakan teknik data mining. Adapun teknik yang digunakan dalam hal ini adalah Algoritma FP-Growth. FP-Growth adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (frequent item set) dalam sekumpulan data. Algoritma FP-Growth merupakan pengembangan dari algoritma Apriori. Sedangkan di dalam algoritma FP-Growth tidak dilakukan generate candidate karena FP-Growth menggunakan konsep pembangunan tree dalam pencarian frequent itemset. Penelitian dilakukan dengan mengamati beberapa variabel penelitian yang sering dipertimbangkan oleh perguruan tinggi khusunya bagian marketing dalam menentukan sasaran promosinya, Yaitu Pendidikan Terakhir,Alamat Rumah,Jurusan,Pilihan Prodi. Hasil penelitian ini adalah berupa suatu perangkat lunak dengan mengimplementasikan algoritma FP-Growth yang menggunakan konsep pembangunan FP-Tree dalam mencari Frequent Itemset.

Kata Kunci : Data Mining,Association Rules,Frequent Itemset,FP-Growth.

ABSTRAC

Nowadays more and more growing amount of campus that make the managers want to get better promotion strategy. So how to find the right promotional strategy will be able to reduce costs and achieve targeted promotional campaign right. One of proces that can be done to determine promotion strategy is to use data mining techniques . The technique used in this case is the FP-Growth Algorithm. FP-Growth algorithm is one of the alternatives that can be used to determine the set of data that appears most frequently (frequent item sets) in a set of data. FP-Growth algorithm is an extension of Apriori algorithm. While in the FP-Growth algorithm do not generate candidate because the FP-Growth using the concept of development in the search for frequent itemset tree. The research doing by observing and some research variables are often considered by colleges especially the marketing in targeting the promotion, Namely Latest Education, Home Address, Subject, Options Prodi. The results of this research are in the form of a software to implement the FP-Growth algorithm that uses the concept of development of FP-Tree in finding Frequent Itemset.

Keywords: Data Mining,Association Rules,Frequent Itemset,FP-Growth.

1. PENDAAHULUAN

1.1 Latar Belakang Masalah

Persaingan di dalam dunia bisnis ini sangat ketat , khususnya dalam mempromosikan perguruan tinggi. Setiap kampus berusaha memberikan fasilitas yang terbaik. Oleh karena itu setiap kampus berusaha mencari strategi yang tepat untuk mempromosikan kampusnya. Menurut para pengembang untuk menemukan suatu strategi yang dapat menentukan strategi pemasaran dalam mempromosikan perguruan tinggi dengan memaksimalkan pelayanan kepada masyarakat. Salah satu teknik yang dipakai dalam penerapan *Data Mining* adalah di dalam bidang promosi. Bila sasaran promosi tidak ditentukan secara baik, dalam arti tidak diupayakan mencari sasaran promosi yang potensial, maka hanya akan menghabiskan banyak waktu dan biaya yang seharusnya bisa diminimalisir melalui pemilihan target promosi yang baik.

Salah satu cara yang dapat diterapkan adalah dengan menerapkan penggunaan *Data Mining*. Karena di dalam data mining sendiri terdapat cara dan teknik dalam pemenuhan kebutuhan salah satunya adalah kebutuhan informasi yang luas, dan dari informasi yang kita dapat bia kita gunakan sebagai suatu keputusan atau menentukan sebuah kualitas dalam menentukan sauatu keputusan. Adapun kumpulan dari sebuah data atau informasi memeiliki banyak potensi untuk dijadikan suatu kesimpulan dalam mengambil sebuah keputusan dengan melakukan analisis dan menggali suatu unformasi yang terdapat di dalam sebuah data. Maka dimungkinkan untuk dibuat strategi dalam mendukung promosi pendidikan. Dari definisi *Data Mining* yang luas terdapat banyak jenis teknik analisa yang dapat digolongkan dalam *Data Mining*. Teknik *Data Mining* yang akan digunakan dalam penelitian ini adalah *Association Rule Mining*. *Association rule mining* adalah teknik *mining* untuk menemukan aturan asosiatif antara suatu kombinasi *item*.

Adapun dalam penelitian ini akan dibahas bagaimana cara mengimplementasikan salah satu algoritma dalam data mining, yaitu algoritma *frequent Pattern-Growth (FP-Growth)*. Algoritma ini adalah bagian dari teknik asosiasi pada data mining. Adapun *FP-Growth* sendiri adalah salah satu altenatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering mucul *(frequent itemset)* dalam sekumpulan data. Karakteristik algoritma *FP-Growth* adalah struktur data yang digunakan dalam *tree* yang disebut *FP-Tree*. Dengan menggunakan *FP-Tree*, algoritma *FP-Growth* dapat langsung mengekstrak *frequent itemset* dari *FP-Tree.*

1.2 Perumusan Masalah

1. Bagaimana sistem memberikan informasi kepada user dalam mempromosikan pendidikan di Perguruan Tinggi Swasta secara efektif dan efisien?
2. Bagaimana sistem ini menentukan target dalam mempromosikan pendidikan?
3. Bagaimana menerapkan algoritma *FP-Growth* di dalam promosi pendidikan sehingga dalam proses tersebut dapat menentukan strategi yang tepat ?
4. Bagaimana melakukan pengujian Metode *FP-Growth* dengan *Rapidminer* sehingga menghasilkan penyelesain masalah dalam mencari strategi promosi yang tepat?

1.3 Batasan Masalah

penulis mencoba memberikan batasan masalah antara lain:

1. Algoritma *FP- Growth* yaitu algoritma yang digunakan di dalam penyeleksian untuk mempercepat proses penetuan *frequent itemset* sebelum meng-*generate rull* sebagai rekomendasi keputusan.
2. Penelitian ini khusus membahas tentang strategi promosi pendidikan pada Perguruan Tinggi khususnya di STMIK TRIGUNA DHARMA .
3. Data yang digunakan sebagai sumber merupakan data yang diambil di STMIK TRIGUNA DHARMA 1 tahun terakhir, yang meliputi asal sekolah, daerah asal, jurusan dan pilihan prodi.
4. *Tools* yang digunakan di dalam pengolahan data adalah *Rapidminer 5.2*

1.4 Tujuan Penelitian

1. Untuk menganalisa permasalahan promosi pendidikan di STMIK TRIGUNA DHARMA.
2. Untuk menerapkan metode *FP- Growth* di dalam strategi promosi pendidikan di STMIK TRIGUNA DHARMA.
3. Untuk melakukan pengujian metode *FP-Growth* dengan menggunakan tools *Rapidminer* sehingga menghasilkan strategi promosi pendidikan yang efektif dan efisien.
4. Menghasilkan sebuah hubungan antara variable atau item berdasarkan implementasi dari algoritma *FP-Growth.*
5. Menguji hubungan variable atau item tersebut untuk mendapatkan pengetahuan baru dari proses ekstraksi menggunakan *Tools* yang ada.

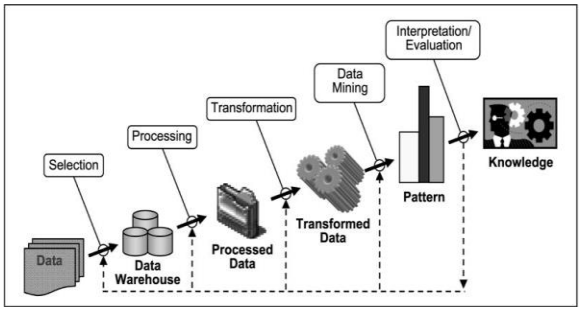
1.5 Manfaat Penelitian

1. Mengetahui algoritma *FP-Growth* dapat memberikan alternatif pengetahuan tentang menemukan strategi dalam promosi pendidikan.
2. Menemukan strategi atau sasaran yang lebih tepat dalam mempromosikan pendidikan.
3. Menjadikan hasil software ini dapat digunakan pihak kampus untuk menentukan strategi dalam promosi pendidikan.

2. LANDASAN TEORI

2.1 *Knowledge Discovery In Database* (KKD)

Knowledge Discovery in Database (KDD) adalah proses menentukan informasi yang berguna serta pola-pola yang ada dalam data. Informasi ini terkandung dalam basis data yang berukuran besar yang sebelumnya tidak diketahui dan potensial bermanfaat. *Data Mining* merupakan salah satu langkah dari serangkaian proses *iterative* KDD. Tahapan proses KDD dapat dilihat pada gambar 2.1.



Gambar 2.1 Tahapan dalam KDD

Tahapan proses KDD terdiri dari:

a. Data *Selection*

Pada proses ini dilakukan pemilihan himpunan data, menciptakan himpunan data target, atau memfokuskan pada subset

variable (sampel data) dimana penemuan (*discovery*) akan dilakukan. Hasil seleksi disimpan dalam suatu berkas yang terpisah dari basis data operasional.

b. Pre-*Processing* dan *Cleaning* Data

Pre-*Processing* dan *Cleaning* Data dilakukan membuang data yang tidak konsisten dan *noise*, duplikasi data, memperbaiki kesalahan data, dan bisa diperkaya dengan data *eksternal* yang relevan.

c. *Transformation*

Proses ini *mentransformasikan* atau menggabungkan data ke dalam yang lebih tepat untuk melakukan proses *mining* dengan cara melakukan peringkasan (*agregasi*).

d. *Data Mining*

Proses *Data Mining* yaitu proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik, metode atau algoritma tertentu sesuai dengan tujuan dari proses KDD secara keseluruhan.

e. *Interpretation* / Evaluasi

Proses untuk menerjemahkan pola-pola yang dihasilkan dari *Data Mining*. Mengevaluasi (menguji) apakah pola atau informasi yang ditemukan bersesuaian atau bertentangan dengan fakta atau hipotesa sebelumnya. Pengetahuan yang diperoleh dari pola-pola yang terbentuk dipresentasikan dalam bentuk visualisasi.

2.2 Data *Mining*

Data *Mining* adalah proses untuk mendapatkan informasi dengan melakukan pencarian pola dan relasi-relasi yang tersembunyi di dalam timbunan data yang banyak. (Fadlina, 2014) Data *Mining* atau sering disebut sebagai *knowledge discovery in database* (KDD) adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam data berukuran besar. Keluaran *Data Mining* ini bisa dipakai untuk membantu pengambilan keputusan di masa depan. Pengembangan KDD ini menyebabkan penggunaan *pattern recognition* semakin berkurang karena telah menjadi bagian *Data Mining*.

2.3 Tahapan dalam Data Mining

Menurut (Buulolo, 2013) ada empat tahap yang dilalui dalam data mining, antara lain:

1. Tahap pertama : *Pricise statement of the problem* (pernyataan tepat terhadap permasalahan)

Sebelum mengakses perangkat lunak data mining, seorang analis harus memiliki kejelasan perihal ‘pertanyaan apa yang akan ingin dijawabnya’. Jika tidak ada *formula* yang tepat untuk problematika yang ada maka anda hanya akan membuang-buang dan uang dalam membuat solusinya.

2. Tahap kedua : *Initial exploration*

Tahap ini dimulai dengan mempersiapkan data yang juga juga termasuk kedalam data mining “*cleaning*” (misalnya : mengedentifikasi dan menyikirkan data yang dikodekkan salah), transformasi data, memilih *subset record, data set,* langkah awal seleksi. Mendeskripsikan dan memvisualisasikan data adalah kunci dari tahap ini.

3. Tahap tiga : *Model building and validation*

Tahap ini melibatkan pertimbangan terhadap ragam permodelan dan memilih yang terbaik bagi performasi prediktif.

4. Tahap ke-empat : *Deployment*

Memilih aplikasi yang tepat berikut permodelan untuk membuat (*generate*) prediksi. Selanjutnya kita kan melihat rincian perihal tahapan-tahapan data mining.

2.4 Pengelompokan *Data Mining*

Data Mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu (Kusrini dan Emha Taufiq Luthfi, 2009):

1. Deskripsi
   Terkadang peneliti dan analis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menentukan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden.

2. Estimasi
   Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih kearah numerik daripada kearah kategori. Sebagai contoh akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya.

3. Prediksi
   Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang.
   Contoh prediksi bisnis dan penelitian adalah:
   a. Prediksi harga beras dalam tiga bulan yang akan datang.
   b. Prediksi persentasi kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikkan.
   Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

4. Klasifikasi
   Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang dan pendapatan rendah.
   Contoh lain klasifikasi dalam bisnis dan penelitian adalah:
   a. Menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang atau tidak.
   b. Memperkirakan apakah suatu pengajuan hipotek oleh nasabah merupakan suatu kredit yang baik atau buruk.
   c. Mendiagnosis penyakit seorang pasien untuk mendapatkan termasuk kategori penyakit apa.

5. Pengklusteran
   Pengklusteran merupakan pengelom-pokan *record*, pengamatan atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan.
   Contoh pengklusteran dalam bisnis dan penelitian adalah:

a. Mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari satu suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar.
b. Untuk tujuan audit akuntansi, yaitu melakukan pemisahan terhadap perilaku *financial* dalam baik dan mencurigakan.
c. Melakukan pengklusteran terhadap ekspresi dari *gen*, untuk mendapatkan kemiripan perilaku dari *gen* dalam jumlah besar.

6. Asosiasi
   Tugas asosiasi dalam *Data Mining* adalah menemukan *attribut* yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja. Contoh asosiasi dalam bisnis dan penelitian adalah:
   a. Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respon positif terhadap penawaran *upgrade* layanan yang diberikan.
   b. Menentukan barang dalam supermarket yang dibeli secara bersamaan dan yang tidak pernah dibeli secara bersamaan.

2.5 Tahapan Asosiasi

Analisis asosiasi dikenal juga sebagai salah satu teknik *Data Mining* yang menjadi dasar dari berbagai teknik *Data Mining* lainnya. Khususnya salah satu tahap dari analisis asosiasi yang disebut analisis pola frequensi tinggi *(frequent pattern mining)* menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien. Penting tidaknya suatu aturan assosiatif dapat diketahui dengan dua parameter, *support* (nilai penunjang) yaitu persentase kombinasi item tersebut dalam *database* dan *confidence* (Fadlina, 2014).

A. Data Preparation.

Pada banyak bidang keilmuan, terutama komputer sains, diperlukan data yang berkualitas melalui proses *preparation* dari data mentah. Dalam prakteknya, ditemukan bahwa data *cleaning* dan *preparation* membutuhkan total usaha 80% dari usaha untuk merekayasa data, sehingga menjadikan data *preparation* sebagai proses yang krusial. Pentingnya proses ini dapat dilihat dari tiga aspek, yakni:

1. Data *real world* merupakan data kotor. Data *real world* bisa mengandung data yang tidak, terdapat *noise*, tidak konsisten, yang dikarenakan:
   a. Tidak lengkap (*incomplete*), yaitu kekurangan nilai atribut atau hanya mengandung agregat data (contoh : address = " ")
   b. *Noise*, yaitu masih mengandung *error* dan *outliers*.
   c. Tidak konsisten (*inconsistent*), yaitu data yang mengandung discrepansi dalam code dan nama atau singkatnya datanya tidak konsisten.
2. Sistem *mining* dengan performa tinggi membutuhkan data yang berkualitas. Data *preparation* atau *preparation* menghasilkan dataset yang lebih sedikit daripada dataset yang asli, ini bias meningkatkan efisiensi dari *Data Mining*. Langkah ini mengandung:
   a. Memilih data yang relevan
   b. Mengurangi data.
3. Data yang berkualitas menghasilkan pola yang berkualitas. Dengan data *preparation*, maka data yang dihasilkan adalah data yang berkualitas, yang mengarah pada pola yang berkualitas pula, dengan:

a. Mengembalikan data yang tidak lengkap
b. Membenarkan *eror*, atau menghilangkan *outliers*
c. Membenahi data yang bertentangan.

B. Association Rules

Association rule merupakan salah satu metode yang bertujuan mencari pola yang sering muncul di antara banyak transaksi, dimana setiap transaksi terdiri dari beberapa *item* sehingga metode ini akan mendukung sistem rekomendasi melalui penemuan pola antar *item* dalam transaksi-transaksi yang terjadi
Metodologi dasar analisis asosiasi terbagi menjadi dua tahap:

1. Analisa pola frekuensi tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai support dalam *database*. Nilai *support* sebuah *item* diperoleh dengan rumus berikut:

$$Support\ (A) = \frac{\text{Jumlah Transaksi Mengandung A}}{\text{Total Transaksi}} \quad (1)$$

$$Support\ (A\ C\ B) = \frac{\text{Jumlah Transaksi Mengandung A dan B}}{\text{Total Transaksi}} \quad (2)$$

2. Pembentukan aturan assosiatif

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan assosiatif yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan assosiatif A _B Nilai *confidence* dari aturan A _B diperoleh dari rumus berikut:

C. Frequent Itemset

Langkah pertama pada *association rule* adalah menghasilkan semua *itemset* yang memungkinkan dengan kemungkinan *itemset* yang muncul dengan *m-item* adalah 2*m*. Karena besarnya komputasi untuk menghitung *frequent itemset*, yang membandingkan setiap kandidat *itemset* dengan setiap transaksi, maka ada beberapa pendekatan untuk mengurangi komputasi tersebut, salah satunya dengan algoritma apriori.

D. Apriori

Algoritma apriori digunakan untuk mencari *frequent itemset* yang memenuhi *minsup* kemudian mendapatkan *rule* yang memenuhi *minconf* dari *frequent itemset* tadi. Algoritma ini mengontrol berkembangnya kandidat *itemset* dari hasil *frequent itemset* dengan *support-based pruning* untuk menghilangkan *itemset* yang tidak menarik dengan menetapkan *minsup*. Prinsip dari apriori ini adalah bila *itemset* digolongkan sebagai *frequent itemset*, yang memiliki *support* lebih dari yang ditetapkan sebelumnya, maka semua subsetnya juga termasuk golongan *frequent itemset*, dan sebaliknya.

E. Rule Generation

Setelah mendapatkan *frequent itemset* menggunakan algoritma apriori, langkah selanjutnya adalah mendapatkan *rule* yang memenuhi *confidence*. Karena *rule* yang dihasilkan berasal dari *frequent itemset*, dengan kata lain, dalam menghitung *rule* menggunakan *confidence*, tidak perlu lagi menghitung *support*-nya karena semua calon *rules* yang dihasilkan telah memenuhi *minsup* sesuai yang ditentukan. Penghitungan ini juga tidak perlu melakukan perulangan *scanning* pada *database* untuk menghitung *confidence*, cukup dengan mengambil *itemset* dari hasil *support.*

2. 6 Algoritma *FP-Growth*

(Sensuse, 2012)*FP-growth* adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. *FP-growth* menggunakan pendekatan yang berbeda dari paradigma yang digunakan pada algoritma Apriori.

(Ririanti, 2014). *FP-Growth* adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent item set*) dalam sekumpulan data. Algoritma *FP-Growth* merupakan pengembangan dari algoritma *Apriori*.

FP-growth adalah metode yang sering *itemset* pertambangan tanpa Generasi calon. Ini membangun sebuah struktur data yang sangat padat *(FP-tree)* untuk kompres *database* transaksi asli.

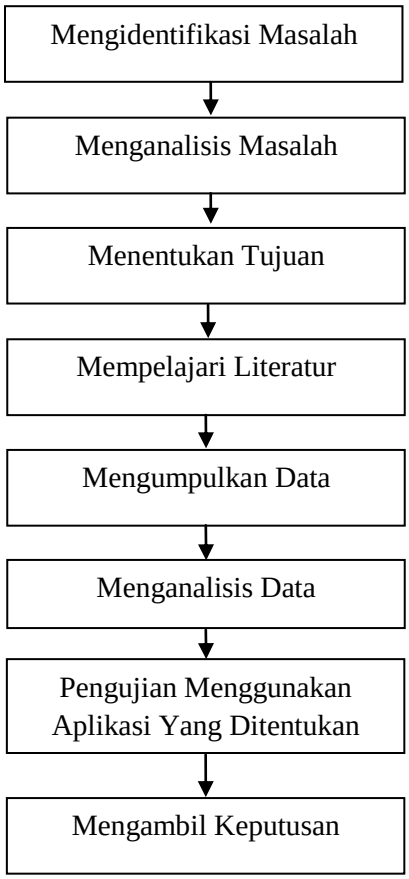
3. METODOLOGI PENELITIAN

3.1 Pendahuluan

Dalam melakukan penelitian agar hasilnya bisa maksimal, penulis mengikuti kaidah-kaidah (metode) yang telah ditetapkan. Metode penelitian yang akan diterapkan adalah metode penelitian ilmiah yang telah diajarkan di bangku perkuliahan sarjana. Metodologi penelitian ini memuat tentang kerangka kerja penelitian yang akan dibahas di bawah ini. Pada bab ini akan dibahas metodologi penelitian untuk mengidentifikasi permasalahan, analisa permasalahan dan pada akhirnya mencari penyelesaian masalah di dalam menganalisa strategi dalam mempromosikan pendidikan pada STMIK TRIGUNA DHARMA MEDAN.

3.2 Kerangka Kerja

Dalam metodologi penelitian ada urutan kerangka kerja yang harus diikuti, urutan kerangka kerja ini merupakan gambaran dari langkah–langkah yang harus dilalui agar penelitian ini bisa berjalan dengan baik. Kerangka kerja yang harus diikuti bisa dilihat pada gambar 3.1.



Gambar 3.1 Kerangka Kerja

Berdasarkan gambar 3.1 dijabarkan urutan–urutan langkah kerja sebagai berikut:

3.2.1. Mengidentifikasi Masalah

Masalah yang diidentifikasi dalam penelitian ini untuk mendapatkan strategi promosi yang efektif dan efisien dalam mendapatkan calon mahasiswa baru

ditengah ketatnya persaingan antar Perguruan Tinggi .

3.2.2. Menganalisis Masalah

Analisis masalah pada penelitian ini dilakukan dengan dua metode yaitu metode deskriptif dan metode komperatif.

a. Metode Deskriptif
   Pada metode ini data yang ada dikumpulkan, disusun, dikelompokkan dan dianalisis sehingga diperoleh beberapa gambaran yang jelas pada masalah yang akan dibahas.

b. Metode Komperatif
   Pada metode ini analisis dilakukan dengan cara membandingkan teori dan praktek sehingga diperoleh gambaran yang jelas tentang persamaan dan perbedaan di antara keduanya.

3.2.3. Menentukan Tujuan

Berdasarkan uraian di atas, tujuan dari penelitian ini adalah:

a. Untuk mengklasifikasikan setiap wilayah di Sumatera Utara dalam menentukan target promosi pendidikan di STMIK TRIGUNA DHARMA MEDAN.
b. Untuk mendapatkan atau menghasilkan suatu knowledge (pengetahuan) tentang wilayah yang akan dipilih dalam mempromosikan pendidikan.
c. Untuk memberikan masukan kepada pihak akademik sebagai bahan pertimbangan pihak akademik di dalam menentukan wilayah yang akan dijadikan target dalam promosikan pendidikan.

Mempelajari Literatur

Literatur-literatur yang dipakai sebagai bahan referensi dalam penelitian ini adalah dari jurnal-jurnal ilmiah, modul pembelajaran dan buku tentang *Data Mining*. literatur-literatur ini akan menjadi pedoman untuk melakukan penelitian agar memudahkan proses penelitian.

3.2.4. Mengumpulkan Data

Metode pengumpulan data dilakukan dengan cara melakukan pengamatan langsung ke kampus STMIK TRIGUNA DHARMA. Selain pengamatan, juga dilakukan wawancara kepada pihak-pihak yang terkait dengan penelitian ini. Di samping itu juga, melakukan pengambilan sampel *database* nasabah untuk menunjang penelitian ini.

3.2.5. Menganalisis Data

Pada tahap ini data yang telah dikumpulkan akan dianalisis. Analisis menggunakan *Association Rule* yang dimaksud dilakukan melalui mekanisme penghitungan *support* dan *confidence* dari suatu hubungan item. Sebuah rule asosiasi dikatakan *interesting* jika nilai *support* adalah lebih besar dari *minimum support* dan juga nilai *confidence* adalah lebih besar dari *minimum confidence* .Dengan menggunakan teknik *FP-Growth* yang menghasilkan *Frequent Itemset* tanpa melakukan *candidates generation*. dengan tujuan untuk mendapatkan pengetahuan yang baru (*knowledge*) berupa informasi tentang strategi promosi yang efektif dan efisien dalam mendapatkan calon mahasiswa baru ditengah ketatnya persaingan antar Perguruan Tinggi.

3.2.6. Pengujian Menggunakan Aplikasi Yang Ditentukan

Pada tahap ini, *rule* dites kembali atau diuji lagi menggunakan sistem *data*

mining yang sudah ada. *Tools* yang digunakan sebagai pengujian sistem adalah *Ramidminer 5. 2*.

Mula-mula data manual ”Data Mahasiswa Baru” yang sudah ada sebelumnya, di transformasikan ke dalam *Microsoft Excel*.

Sebagai tahap awal di dalam proses pengujian yaitu *mengimport* data mahasiswa baru pada kampus STMIK TRIGUNA DHARMA yang akan dijadikan sebagai data tabel transaksi *frequensi item* di dalam *tools Ramidminer 5 2.* tersebut. Kemudian tahap selanjutnya yaitu, melakukan proses *Data Input* dan *Affinity analysis* atau *market basket analysis*. Analisis asosiasi atau *association rule* adalah teknik *Data Mining* untuk menemukan aturan asosiatif antara suatu kombninasi *item*. Analasis *FP-Growth* didefenisikan suatu proses untuk menemukan semua aturan *FP-Growth* yang memenuhi syarat *minimum support* dan syarat minimum untuk *confidence*. Dari proses *Data Input* dan *Affinity* tersebut akan terdapat pola atau kriteria tertentu dalam mendapatkan strategi promosi. Hasil dari Proses *Data Input* dan *Affinity* tersebut akan diolah kembali untuk menghasilkan suatu *association rule*. Dari *association rule* tersebut akan terlihat hasilnya. Hasilnya tersebut merupakan pengetahuan baru *(knowledge)* dan juga merupakan sebagai acuan untuk menganalisa strategi dalam mempromosikan pendidikan.

3.2.7. Pengambilan Keputusan

Setelah diuji, hasil analisis antara *manual ways* (cara manual) dan pengujian memanfaatkan *tools* akan terlihat perbandingannya. Langkah berikutnya adalah penentuan atau pengambilan keputusan terhadap *knowledge* yang baru didapat yaitu berupa penentuan dalam menentukan strategi dalam mempromosikan pendidikan.

4. ANALISA

4.1 Analisa Data

Karakteristik algoritma *FP-Growth* adalah struktur data yang digunakan adalah *tree* yang disebut dengan *FP-Tree*. Dengan menggunakan *FP-Tree*, algoritma *FP-growth* dapat langsung mengekstrak *frequent Itemset* dari *FP-Tree.* Penggalian *itemset* yang *frequent* dengan menggunakan algoritma *FP-Growth* akan dilakukan dengan cara membangkitkan struktur data *tree* atau disebut dengan *FP-Tree*. Metode *FP-Growth* dapat dibagi menjadi 3 tahapan utama yaitu sebagai:

1. Tahap pembangkitan *conditional pattern base*,
2. Tahap pembangkitan *conditional FP-Tree*, dan
3. Tahap pencarian *frequent itemset*.

   Ketiga tahap tersebut merupakan langkah yang akan dilakukan untuk mendapat *frequent itemset*.

Input : *FP-Tree Tree*
Output : *Rt* Sekumpulan lengkap pola *frequent*
Methode : *FP-Growth (Tree, null)*
Procedure : *FP-Growth (Tree, α)*
{
01*: if Tree* mengandung *single path* P;
02*: then* untuk tiap kombinasi (dinotasikan β) dari *node-node* dalam *path do*
03: bangkitkan pola β α dengan *support* dari *node-node* dalam *path do* β;
04: *else u*ntuk tiap a1 dalam *header* dari *tree do*
}
05: bangkitkan pola

06: bangun β = a1 α dengan *support* = a1 *support*
07*: if Tree* β =

4.2 Pra Proses

Tabel 4.1 Untuk Memberi Tanda Pada Setiap *Itemset*

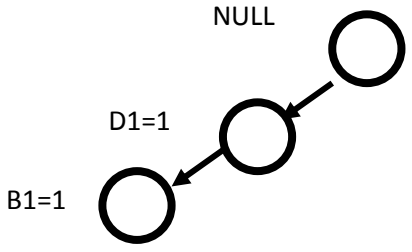
Item
SMK Negeri (A1)
SMK Swasta(A2)
SMA Negeri (A3)
SMA Swasta (A4)
Madrasah Aliyah (A5)
Medan (B1)
Aceh (B2)
Lubuk Pakam (B3)
Deli Serdang (B4)
Perbaungan (B5)
Tebing Tinggi (B6)
Riau (B7)
IPA (C1)
IPS (C2)
TKJ (C3)
MULTIMEDIA (C4)
Sistem Informasi (D1)
Sistem Komputer (D2)
Teknik Komputer (D3)
Manajemen Informatika (D4)

Tabel 4.2 Frekuensi dari Setiap Transaksi *Item*

TID	ITEM
1	D1,B1
2	C1,A3,B1
3	A3,C2
4	D1,A4,B1,C2
5	D1,C1,A4
6	C1,A3
7	D1,C1,A3
8	D1
9	D1,C1,A3
10	D1,C1,A4
11	D1,C1,A4
12	A3,B1.C2
13	D1,C1
14	C1
15	D1,C1,A3
16	C1,A3
17	D1,C1,A3
18	-
19	D1,A4,C2
20	D1,A4,B1,C2
21	A3,B1,C2
22	A3,B1,C2
23	D1,B1
24	D1
25	D1,C1,A4,B1
26	D1,C1,A4,B1
27	D1,C1,A4
28	D1,C1,A3
29	C2
30	D1,A4,C2

Maka langkah selanjutnya adalah membentuk pohon *FP-Tree* dengan melihat tabel 4.2 sebagai berikut:

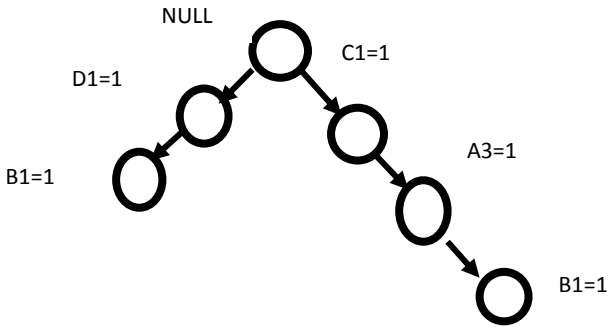
Gambar di bawah ini memberikan ilustrasi mengenai pembentukan *FP-tree* setelah pembacaan TID pada tabel 4.2



Gambar 4.1: Hasil Pembentukan *FP-tree* Setelah Pembacaan TID 1

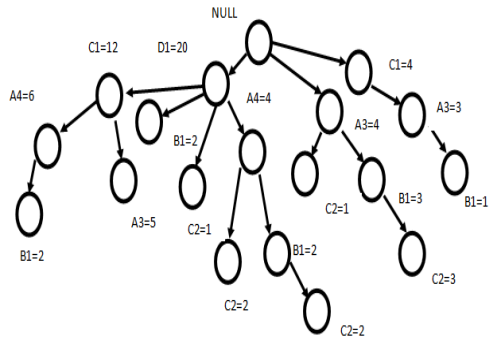
Gambar 4.1 adalah penjelasan tentang pembentukan *FP-Tree* setelah

pembacaan didapat setelah melakukan TID 1, Yaitu berisi : NULL- D1 (Sistem Informasi) =1 –(B1) Medan= 1.



Gambar 4.2 : Hasil Pembentukan *FP-tree* Setelah Pembacaan TID 2

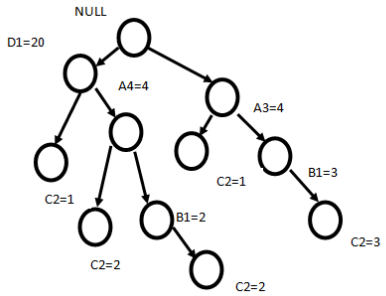
Gambar 4.2 didapat setelah melakukan TID 2 ,Yaitu: NULL-C1 (IPA) =1-A3 (SMAN) =1-B1(Medan) =1.



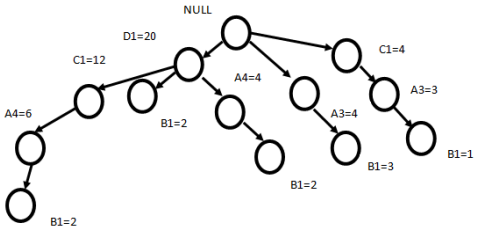
Gambar 4.3 : Hasil Pembentukan *FP-tree* Setelah Pembacaan TID 30

Gambar didapat setelah melakukan TID 30 yang dijumlahkan, yaitu berisi Null-Sistem Informasi (D1) = 20 -IPA (C1) =16 -SMA Negeri (A3) =12 -SMA Swasta (A4) =10-Medan (B1) = 10-IPS (C2) = 9.

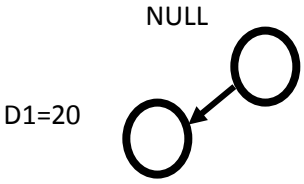
Untuk menemukan *Frequent itemset* dari tabel 4.2, maka perlu ditentukan terlebih dahulu lintasan yang berakhir dengan *support count* terkecil, yaitu C2 yang diikuti dengan A4,A3, C1,D1 dan di akhiri D1. Proses pembentukan masing-masing *node* dapat dilihat pada gambar 4.4.



Gambar 4.4 Lintasan yang Mengandung Simpul C2



Gambar 4.5 Lintasan yang Mengandung Simpul B1



Gambar 4.6 Lintasan yang Mengandung Simpul D1

Setelah mencari *frequent itemset* untuk beberapa akhiran *suffix* maka didapat hasil yang dirangkum dalam tabel berikut:

Tabel 4.3 Daftar *Frequent Itemset* Diurutkan Berdasarkan Hubungan Akhiran

Suffix	*Frequent Itemset*
C2	{C2},{C2,A4}{C2,A3}{C2,D1}{C2,B1,A3},{C2,B1,A4}
B1	{B1},{B1,D1},{B1,A3},{B1,A3,C1},(B1,A4,D1},{B1,A4,C1,D1}
A4	{A4},{A4,C1,D1}

A3	{A3},{A3,C1},{A3,C1,D1}
C1	{C1},{ C1,D1}
D1	{D1}

Dari *frequent itemset* yang didapat dari pembentikan *Fp-Tree* dan *FP-Growth* maka dapat dihitung nilai *Support* dan *Confidence* sebagai berikut:

Support = (Sistem Informasi,IPA,SMA Swasta,Medan) = *Count* (Sistem Informasi,IPA,SMA Swasta,Medan)/Jumlah Transaksi = 1/30.

Support = (Sistem Informasi,IPA,SMA Negeri) = *Count* (Sistem Informasi,IPA,SMA Negeri)/Jumlah Transaksi = 1/30.

Support = (Sistem Informasi, Medan) *Count* (Sistem Informasi, Medan)/Jumlah Transaksi =1/30.

Support = (Sistem Informasi,SMA Swasta, IPS) *Count*(Sistem Informasi,SMA Swasta, IPS)/Jumlah Transaksi = 1/30.

Support = (Sistem Informasi,SMA Swasta,Medan, IPS) *Count* (Sistem Informasi,SMA Swasta, Medan,IPS)/Jumlah Transaksi =1/30.

Support = (SMA Negeri ,IPS) *Count* (SMA Negeri ,IPS) / Jumlah Transaksi = 1/30.

Support = (SMA Negeri ,Medan,IPS) *Count* (SMA Negeri ,Medan,IPS) / Jumlah Transaksi = 1/30.

Support = (IPA,SMA Negeri ,Medan) *Count* (IPA,SMA Negeri ,Medan) / Jumlah Transaksi = 1/30.

Sedangkan untuk *confidence* atau nilai kepercayaannya adalah sebagai berikut :

Confidence = (SMA Negeri,IPS) = Count (SMA Negeri,IPS)/*Count* Medan= 1/9.

Confidence = (Sistem Informasi,SMA Swasta,Medan,IPS) = *Count* (Sistem Informasi,SMA Swasta,Medan,IPS /*Count* Medan = 1/9.

Confidence = (SMA Negeri,Medan,IPS) *Count* (SMA Negeri,Medan,IPS /*Count* Medan =1/9.

Confidence = (Sistem Informasi,SMA Swasta,IPS) *Count*(Sistem Informasi,SMA Swasta,IPS)/*Count* Medan= 1/19.

Confidence = (Sistem Informasi,SMA Swasta,Medan,IPS) *Count* (Sistem Informasi,SMA Swasta,Medan,IPS)/*Count* Medan =1/9.

Confidence = (SMA Negeri, Medan,IPS) *Count* (SMA Negeri, Medan,IPS) / *Count* Medan =1/9.

Confidence = (Sistem Informasi,IPS) *Count* (Sistem Informasi,IPS) / *Count* Medan =1/9.

Confidence = (Sistem Informasi,SMA Swasta,IPS) *Count* (Sistem Informasi,SMA Swasta,IPS) / *Count* Medan =1/9.

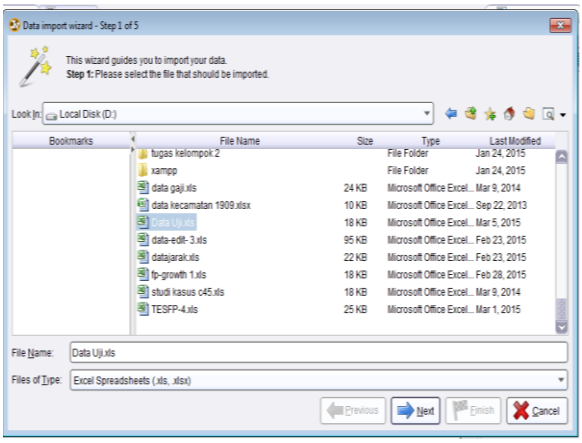
Setelah didapat nilai *support* dan *confidence* dari keseluruhan kombinasi pada data dengan perhitungan *FP-Tree* dan *FP-Growth* maka didapat niai *support* dan *confidence* yang paling tinggi dan akurat yaitu kombinasi. (D1,C1,A3) { Sistem Informasi, IPA, SMA Negeri } yang mempunyai nilai *support* : 5/30 = 0,16 dan nilai *confidence* : 5/20 = 0.25.

5. IMPLEMENTASI SISTEM

Untuk membuktikan kebenaran hasil analisa diperlukan sebuah proses pengujian untuk menguji kebenaran dari hasil pengolahan data yang dikerjakan secara manual pada BAB IV yang telah dikerjakan sebelumnya, untuk proses pengujian tersebut kita dapat menggunakan salah satu *software* aplikasi seperti *Rapidminer,* dengan langkah-langkah sebagai berikut:

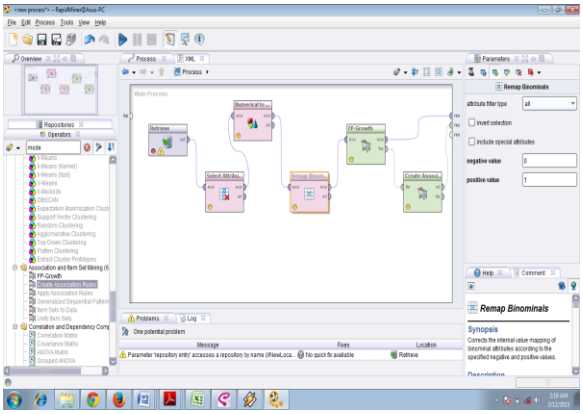
1. Dataset yang terdiri 30 record data dan berisi variabel-variabel atribut Sistem Informasi,IPA,SMA Negeri,SMA

Swasta,Medan, IPS disimpan dalam aplikasi *microsoft excel* dengan nama *file* datauji.xls dan yang akan dicoba menggunakan *software rapidminer* 5.2 untuk melihat hasil sama tidaknya dengan pencarian frequent itemsetnya melalu :



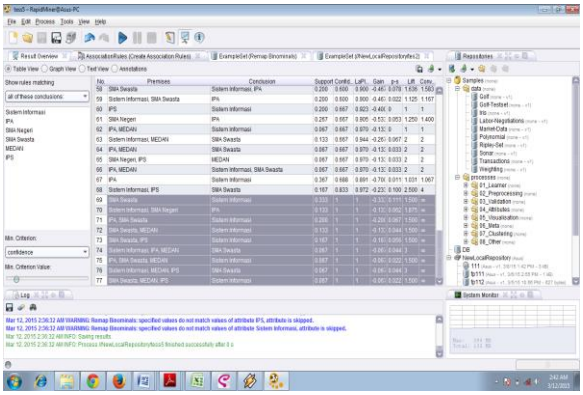
Gambar 5.1 Tampilan *Import Excel Sheet*

Gambar 5.1 adalah proses dari import data uji Xls ke dalam rapidminer



Gambar 5.2 Proses menghubungkan *FP-Growth* ke ras pertama dan *Create Association Rules* ke res kedua

Pada Gamabar 5.2 dijelaskan tentang proses menghubungkan *FP-Growth* ke ras Pertama dan *Create Association Rules* ke ras kedua dan diatur pada minimum *support* dan *minimum confidence*nya.



Gambar 5.20 *Rule valid* jika nilai *Lift Ratio* > 1

Lift Ratio adalah parameter penting selain *support* dan *confidence* dalam *association rule*. *Lift Ratio* mengukur seberapa penting *rule* yang teleh terbentuk berdasarkan nilai *support* dan *confidence*. *Lift Ratio* merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar produk A dibeli bersamaan dengan produk B.

Improvement Ratio dapat dihitung dengan rumus

$$\frac{Support(A \cap B)}{SupportA * SupportB} =$$

Untuk mencari nilai *rule* yang *valid* adalah jika mempunyai nilai *Lift Ratio* > 1 dengan cara *Support Lift Ratio* = *Support* yang mengandung nilai A dan nilai B dibagi *support* A* *Support* B nilai *support* yang mengandung nilai A dan nilai B Adalah hasil dari minimum *support* dibagi dengan *item* (D1,C1,A3) { Sistem Informasi, IPA, SMA Negeri }Pada setiap kemunculan ditransaksi.

=

$$\frac{5/30}{20/30*16/30*12/30} = \frac{0{,}16}{0{,}66*0{,}53*0{,}4} =$$

$$\frac{0{,}16}{0{,}1392} = 1{,}1494$$

Jadi dari hasil pencarian nilai *Lift Ratio* > 1 maka kita bisa menentukan sebuah *rule* yang *valid* dari sekian banyaknya rule yang dijalankan dengan *software Rapidminer*.

Adapun hasil dari rule yang paling mempangaruhi adalah:

Jika dia Asal sekolahnya SMA Swasta, Alamatnya Medan dan Asal Jurusanya IPS maka dia memilih Prodi Sistem Informasi dengan tingkat kepercayaan 100% dan didukung 6% dari data keseluruhan.

Hasil dari rule tersebut yang akan dijadikan target dalam mempromosikan pendidikan.

6. SIMPULAN DAN SARAN

6.1 Simpulan

1. Metode dalam pencarian *Frequent Itemsset* pohon keputusan menggunakan algoritma *FP-Ggrowth* bekerja sangat baik dalam melakukan *Frequent Itemsset* dengan proses pembentukan *FP-Tree* dengan menghasilkan *rule* dari data sampel mahasiswa baru.
2. Penentuan data variabel sangat menentukan tingkat akurasi *FP-Growth* yang dibuat dan besarnya presentase dalam menentukan minum *support* dan minimum *confidence* dipengaruhi oleh data variabel yang digunakan untuk mencari *frequent itemset* yang saling berhubungan untuk menemukan data variabel yang akan dijadikan stretegi dalam promosi pendidikan.
3. Dari penelitian yang dilakukan ada beberapa atribut yang tidak digunakan dalam *rule* yang dihasilkan, sehingga pemilihan atribut di dalam *dataset* sangat penting.
4. Algoritma *FP-Growth* dapat diterapkan untuk mendukung strategi promosi pendidikan pada Perguruan Tinggi. Adapun Informasi yang berkaitan dengan pelaksanaan promosi dapat tersedia dengan cepat, contohnya dalam pemilihan tempat yang akan dijadikan promosi pendidikan, sehingga pihak manajemen dari kampus sendiri dapat melakukan pengambilan keputusan dengan cepat.
5. Pelaksanaan promosi pendidikan pada perguruan tinggi sangat dibantu dengan adanya penerapan algoritma *FP-Growth* ini sehingga diharapkan efektifitas pelaksanaan promosi pendidikan akan dapat semakin ditingkatkan.
6. Untuk menemukan *rule* atau *knowledge* yang sangat berarti dalam menganalisa strategi dalam menentukan promosi pendidikan, dibutuhkan banyak data sampel yang digunakan untuk proses ekstraksi *rule* atau *knowledge* tersebut.

6.2 Saran

Salah satu tujuan penelitian ini adalah agar teknik dalam menentukan strategi promosi pendidikan di Kampus STMIK TRIGUNA DHARMA MEDAN dapat *diimplementasikan* atau diterapkan dengan baik, untuk itu saran yang dapat diperhatikan untuk *pengimplementasian* dari teknik ini adalah:

1. Pada penelitian ini, penulis hanya mencoba salah satu teknik dalam menemukan hubungan beberapa *item* yang digunakan untuk membentuk FP-*Tree* dalam menentukan strategi

promosi pendidikan dengan menggunakan Algorima *FP-Growth.* Untuk menentukan stategi dalam promosi pendidikan yang baik dapat digunakan beberapa atau penggabungan beberapa teknik yang akan digabungkan dalam sistem pengambilan keputusan untuk menentukan stategi yang tepat dalam menentukan stategi promosi pendidikan atau bisa juga menggunakan algoritma *CT-Pro.* Oleh sebab itu perlunya dilakukan penelitian lebih lanjut untuk mendapatkan strategi promosi dan menentukan teori mana yang menghasilkan strategi yang paling baik.

2. Untuk mendapatkan hasil promosi yang baik sangat diperlukan sumber data yang yang lebih banyak dan lebih lengkap.
3. Untuk mengimplementasikan *rule* yang dihasilkan dari teknik dari algoritma *FP-Growth* dapat berkonsultasi dengan tim marketing yang ada di kampus tersebut. Sehingga dengan adanya hubungan algoritma *FP-Growth* dengan pihak manajemen dapat mengembangkan strategi dalam promosi pendidikan sehingga pihak kampus akan lebih maju lagi kedepannya.

7. DAFTAR PUSTAKA

BUULOLO, E. 2013. Implementasi Algoritma Apriori Pada Sistem Persediaan Obat (Studi Kasus : Apotik

Rumah Sakit Estomihi Medan). *Pelita Informatika Budi Darma,* Vol : IV No: 1**,** 71-83.

FADLINA. 2014. Data Mining Untuk Analisa Tingkat Kejahatan Jalanan Dengan Algoritma Association Rule Metode Apriori (Studi Kasus Di Polsekta Medan Sunggal). *Informasi dan Teknologi Ilmiah (INTI),* Vol : III No : 1**,** 144-154.

RIRIANTI. 2014. Implementasi Algoritma FP-GROWTH Pada Aplikasi Prediksi Persedian Sepeda Motor (Studi Kasus PT. Pilar Deli Labumas). *Pelita Informatika Budi Darma,* Vol : VI No: 1.

SENSUSE, G. G. D. D. I. 2012. Penerapan Metode Data Mining Market Basket Analysis Terhadap Data Penjualan Produk Buku Dengan Menggunakan Algoritma Apriori dan Frequent Pattern Growth (FP-GROWTH) : Studi Kasus Percetakan PT. GRAMEDIA. *Jurnal TELEMATIKA MKOM,* Vol.4 No.1**,** 118-132.

KUSRINI dan EMHA TAUFIQ LUTHFI. 2009. “Algoritma Data Minning.” Jakarta : Andi. 10

Int. J. of Advanced Networking and Applications
Volume:01, Issue:06, Pages: 400-404 (2010)

Implementation of Web Usage Mining Using APRIORI and FP Growth Algorithms

B.Santhosh Kumar
Department of Computer Science, C.S.I. College of Engineering, Ketti- 643 215. The Nilgiris.
Email: b.santhoshkumar@csice.edu.in

K.V.Rukmani
Department of Computer Science, C.S.I. College of Engineering, Ketti- 643 215. The Nilgiris.
Email: rukmani_arun@csice.edu.in

ABSTRACT

Web Usage Mining is the application of data mining techniques to discover interesting usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. Usage data captures the identity or origin of Web users along with their browsing behavior at a Web site. Web usage mining itself can be classified further depending on the kind of usage data considered. They are web server data, application server data and application level data. Web server data correspond to the user logs that are collected at Web server. Some of the typical data collected at a Web server include IP addresses, page references, and access time of the users and is the main input to the present Research. This Research work concentrates on web usage mining and in particular focuses on discovering the web usage patterns of websites from the server log files. The comparison of memory usage and time usage is compared using Apriori algorithm and Frequent Pattern Growth algorithm.

Keywords : Apriori, Data cleaning, FP Growth, FP-tree, Web Usage Mining,

Date of Submission: 16, February 2010 Date of Acceptance: 08, A pril 2010

1. INTRODUCTION

The Web is a huge, explosive, diverse, dynamic and mostly unstructured data repository, which supplies incredible amount of information, and also raises the complexity of how to deal with the information from the different perspectives of view, users, web service providers, business analysts. The users want to have the effective search tools to find relevant information easily and precisely. The Web service providers want to find the way to predict the users' behaviors and personalize information to reduce the traffic load and design the Web site suited for the different group of users. The business analysts want to have tools to learn the user/consumers' needs. All of them are expecting tools or techniques to help them satisfy their demands and/or solve the problems encountered on the Web. Therefore, Web mining becomes a popular active area and is taken as the research topic for this investigation.

Web Usage Mining [4], [5] is the application of data mining techniques to discover interesting usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. Usage data captures the identity or origin of Web users along with their browsing behavior at a Web site. Web usage mining itself can be classified further depending on the kind of usage data considered. They are web server data, application server data and application level data. Web server data correspond to the user logs that are collected at Web server. Some of the typical data collected at a Web server include IP addresses, page references, and access time of the users and is the main input to the present Research. This Research work concentrates on web usage mining and in particular focuses on discovering the web usage patterns of websites from the server log files.

1.1 APRIORI ALGORITHM

The present Research work is designed to operate on log files. The algorithm [7] attempts to find subsets which are common to at least a minimum number C (the cutoff, or confidence threshold) of the item sets. The system operates in the following three modules.

- Preprocessing module
- Apriori or FP Growth Algorithm Module
- Association Rule Generation
- Results

The preprocessing module converts the log file, which normally is in ASCII format, into a database like format, which can be processed by the Apriori algorithm.

The second module is performed in two steps.

- Frequent Item set generation
- Rules derivation

1.2 FP GROWTH ALGORITHM:

The FP Growth algorithm operates in the following four modules.

- Preprocessing module
- FP Tree an FP Growth Module
- Association Rule Generation
- Results

The preprocessing modules convert the log file, which normally is in ASCII format, into a database like format, which can be processed by the FP Growth algorithm.

The 2nd module is performed in two steps.

- FP Tree generation
- Applying FP Growth to generate association rules

FP tree is a compact data structure that stores important, crucial and quantitative information about frequent patterns.

The main components of FP tree are:

- It consists of one root labeled as "root", a set of item prefix sub-trees as the children of the root, and a frequent-item header table.
- Each node in the item prefix sub-tree consists of three fields: item-name, count, and node-link, where item-name registers which item this node represents, count registers the number of transactions represented by the portion of the path reaching this node, and node-link links to the next node in the FP-tree carrying the same item-name, or null if there is none.
- Each entry in the frequent-item header table consists of two fields, (1) item-name and (2) head of node-link, which points to the first node in the FP-tree carrying the item-name.

Second, an FP-tree-based pattern-fragment growth mining method is developed, which starts from a frequent length-1 pattern (as an initial suffix pattern), examines only its conditional-pattern base (a "sub-database" which consists of the set of frequent items co-occurring with the suffix pattern), constructs its (conditional) FP-tree, and performs mining recursively with such a tree. The pattern growth is achieved via concatenation of the suffix pattern with the new ones generated from a conditional FP-tree. Since the frequent item set in any transaction is always encoded in the corresponding path of the frequent-pattern trees, pattern growth ensures the completeness of the result.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The Research work was initiated through a system study and analysis phase, where significant study was conducted to understand the existing system. Using Apriori algorithm for weblog mining is a novel technique. The explosive growth of the World Wide Web (WWW) in recent years has turned the web into the largest source of available online data.

- Situations like several unrelated topics in a single web page may lead to confusion and make harder to reach the information that the visitors are looking for.
- The design of the whole site (interface, content, structure, usability, etc.) is one of the most important aspects for any institution that wants to survive in the cyberspace.
- Understand the way user browses the site and find out which is the most frequent used link and pattern of using the features available in the site.

All these information is available online but are hidden for the users. Presently, there is no powerful that can analyze this hidden information and this Research work uses web usage mining (WUM) Apriori based approach for analyzing the visitor browsing behavior.

2.2 LIMITATIONS OF APRIORI ALGORITHM

Apriori algorithm, in spite of being simple, has some limitation. They are,

- It is costly to handle a huge number of candidate sets. For example, if there are 10^4 frequent 1-item sets, the Apriori algorithm will need to generate more than 10^7 length-2 candidates and accumulate and test their occurrence frequencies. Moreover, to discover a frequent pattern of size 100, such as $\{a_1, \ldots, a_{100}\}$, it must generate $2^{100} - 2 \sim 10^{30}$ candidates in total. This is the inherent cost of candidate generation, no matter what implementation technique is applied.
- It is tedious to repeatedly scan the database and check a large set of candidates by pattern matching, which is especially true for mining long patterns.

In order to overcome the drawback inherited in Apriori, an efficient FP-tree based mining method, FP-growth, which contains two phases, where the first phase constructs an FP tree, and the second phase recursively Researches the FP tree and outputs all frequent patterns.

2.3 PROPOSED SYSTEM

The main goal of the proposed system is to identify usage pattern from web log files of a website. Apriori [7] and FP Growth Algorithm is used for this purpose. The main goal of the proposed system is to identify usage pattern from web log files of a website. For this purpose, the usage of apriori and FP Growth algorithms are proposed. Both are influential algorithms for mining frequent item sets for boolean association rules [1], [9]. In computer science and data mining, Apriori is a classic algorithm for learning association rules [10]. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or

details of a website frequentation).

Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k - 1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The key concepts in this algorithm are

- Frequent Item sets: The sets of item which has minimum support (denoted by Lifor ith-Item set).
- Apriori Property: Any subset of frequent item set must be frequent.
- Join Operation: To find L_k, a set of candidate k-item sets is generated by joining L_{k-1} with itself.

The advantages of using apriori algorithm are

- Uses large item set property.
- Easily parallelized
- Easy to implement

2.4 ADVANTAGES OF FP GROWTH ALGORITHM

The major advantages of FP-Growth algorithm is,
- Uses compact data structure
- Eliminates repeated database scan

FP-growth is an order of magnitude faster than other association mining algorithms and is also faster than tree-Researching. The algorithm reduces the total number of candidate item sets by producing a compressed version of the database in terms of an FP-tree. The FP-tree stores relevant information and allows for the efficient discovery of frequent item sets.

The algorithm consists of two steps:
- Compress a large database into a compact, *Frequent-Pattern tree* (FP-tree) structure
  – highly condensed, but complete for frequent pattern mining and avoid costly database scans
- Develop an efficient, FP-tree-based frequent pattern mining method (FP-growth)
  – A divide-and-conquer methodology: decompose mining tasks into smaller ones and avoid candidate generation: sub-database test only

2.5 ADVANTAGE OF FP-TREE STRUCTURE

The most significant advantage of the FP-tree is that the algorithm scans the tree only twice. Apart from this major advantage, the others include,

- Completeness:
  The FP-tree contains all the information related to mining frequent patterns (given the min_support threshold)

- Compactness:
  – The size of the tree is bounded by the occurrences of frequent items
  – The height of the tree is bounded by the maximum number of items in a transaction

Three major steps performed are starting the processing from the end of list L:

- Construct conditional pattern base for each item in the header table
- Construct conditional FP-tree from each conditional pattern base
- Recursively mine conditional FP-trees and grow frequent patterns obtained so far. If the conditional FP-tree contains a single path, simply enumerate all the patterns

3. SYSTEM DESIGN AND DEVELOPMENT

3.1 INPUT DESIGN

The input design is the process of converting user-oriented input to a computer-based format. The goal of the input design is to make the data entry easier, logical and error free. In the present Research work, the input is taken from the web log file. The web log file has the extension .log and contains ASCII characters. A log file is a text file in which every page request made to the web server is recorded. For each request the corresponding log file contains the following information:

IP address of the computer making the request;
- User ID, (this field is not used in most cases);
- date and time of the request;
- a status field indicating if the request was successful;
- size of the file transferred;
- Referring URL, that is, the URL of the page which contains the link that generated the request; name and version of the browser being used.

This information can be used to reconstruct the user navigation sessions within the site from which the log data originates. In an ideal scenario, each user is allocated an unique IP address whenever an access is made to a given web site. Moreover, it is expected that a user visits the site more than once, each time possibly with a different goal in mind. Once the domain-dependent data transformation phase is completed, the resulting transaction data must be formatted to conform to the data model of the appropriate data mining task. For instance,

Volume:01, Issue:06, Pages: 400-404 (2010)

the format of the data for the [2],[12] discovery task may be different than the format necessary for mining sequential patterns. Finally, a query mechanism will allow the user (analyst) to provide more control over the discovery process by specifying various constraints.

2.5 OUTPUT DESIGN

The main objective of any system is the generation of reports. It has various uses. Some of them are,

- For the users, reports provide source of information required.
- They provide permanent hard copy of the results of transactions.

Careful consideration has been given in the designing of the reports as it helps in decision-making process. In the present work, the performance of the system is judged using two metrics. The first one is the amount of memory used and the second one is the time taken for the algorithm to create the association rules [3], [9]. It was found that the FP Growth algorithm is fast and uses only very small amount of memory when compared with Apriori algorithm.

4. CODING STANDARDS

4.1 APRIORI ALGORITHM

In computer science and data mining, Apriori is a classic algorithm for learning association rules [6],[11]. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k - 1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The key concepts in this algorithm are

- Frequent Item sets: The sets of item which has minimum support (denoted by Lifor ith-Item set).
- Apriori Property: Any subset of frequent item set must be frequent.
- Join Operation: To find L_k, a set of candidate k-item sets is generated by joining L_{k-1} with itself.

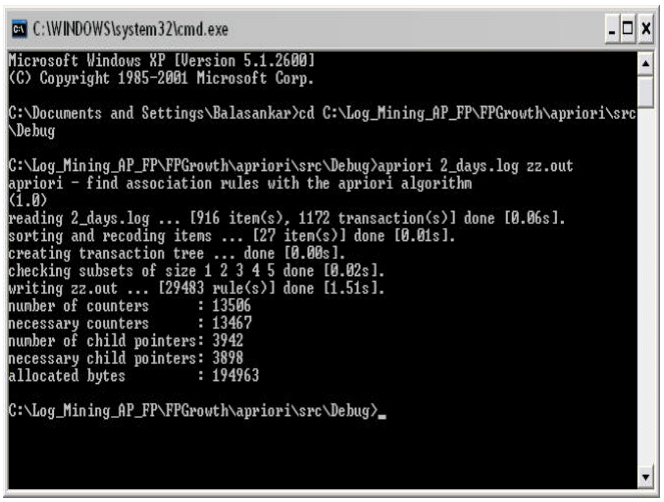
The advantages of using Apriori algorithm are

- Uses large item set property.
- Easily parallelized
- Easy to implement

The Apriori algorithm is an efficient algorithm for finding all frequent item sets. It implements level-wise search using frequent item property and can be additionally optimised. The Apriori algorithm used is given below.

- L_k: Set of frequent item sets of size k (with min support)
- C_k: Set of candidate item set of size k (potentially frequent item sets)

L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) **do**
  C_{k+1} = candidates generated from L_k;
  for each transaction t in database do
    increment the count of all candidates in C_{k+1} that are contained in t
  L_{k+1} = candidates in C_{k+1} with min_support
return $\cup_k L_k$;



4.2 FP GROWTH ALGORITHM

Definition of FP-Tree

FP-tree is a frequent pattern tree consists of one root labeled as "null". It has a set of *item prefix sub trees* as the children of the root, and a *frequent-item header table*. Each node in the *item prefix sub trees* has three fields:

- item-name to register which item this node represents,
- count, the number of transactions represented by the portion of the path reaching this node, and
- Node-link that links to the next node in the FP-tree carrying the same item-name, or null if there is none.

Each entry in the *frequent-item header table* has two fields,

- item-name, and
- head of node-link that point to the first node in the FP-tree carrying the item-name.

5. CONCLUSION

Web usage mining is the application of data mining techniques to discover usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. Web usage mining consists of three phases, namely preprocessing, pattern discovery, and pattern analysis. This Research work implements each of these phases. One of the algorithms which is very simple to use and easy to implement is the Apriori algorithm. This algorithm is used in the present Research work to generate association rules that associates the usage pattern of the clients for a particular website. The output of the system was in terms of memory usage and speed of producing association rules.

The main drawback of Apriori algorithm is that the candidate set generation is costly, especially if a large number of patterns and/or long patterns exist. The main drawback of FP-growth algorithm is the explosive quantity of lacks a good candidate generation method. Future research can combine FP-Tree with Apriori candidate generation method to solve the disadvantages of both apriori and FP-growth. In future the algorithm can be extended to web content mining, web structure mining, etc. The work can also be extended to extract information from image files.

6. ACKNOWLEDGEMENTS

"Implementation of web usage mining using Apriori and FP growth algorithms" research is supported by the management of ***C.S.I College of Engineering, Ketti***, The Nilgiris, whose support we are pleased to acknowledge. We are also grateful to our colleagues in CSICE for useful discussions, and thank to our beloved CSICE students.

REFERENCES

[1] Kotsiantis S, Kanellopoulos D., *Association Rules Mining: A Recent Overview, GESTS International Transactions on Computer Science and Engineering*, Vol.32 (1), 2006, pp. 71-82

[2] Agrawal R, Srikant R., "*Fast Algorithms for Mining Association Rules*", VLDB. Sep 12-15 1994, Chile, 487-99, pdf, ISBN 1-55860-153-8.

[3] Mannila H, Toivonen H, Verkamo A I., "*Efficient algorithms for discovering association rules." AAAI Workshop on Knowledge Discovery in Databases* (SIGKDD). July 1994, Seattle, 181-92.

[4] Tan, P. N., M. Steinbach, V. Kumar, "*Introduction to Data Mining"*, Addison-Wesley, 2005, 769pp.

[5] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques With Java Implementation*, 2nd ed. San Mateo, CA: Morgan Kaufmann, 2005.

[6] P. Becuzzi, M. Coppola, and M. Vanneschi, "*Mining of Association Rules in Very Large Databases: A Structured Parallel Approach,"* Proc. Europar-99, vol. 1685, pp. 1441-1450, Aug. 1999.

[7] R. Jin and G. Agrawal, *"An Efficient Implementation of Apriori Association Mining on Cluster of SMPs,"* Proc. Workshop High Performance Data Mining (IPDPS 2001), Apr. 2001.

[8] J. Han and M. Kamber, *"Data Mining: Concepts and Techniques"* .Morgan Kaufmann Publishers, 2000.

[9] E.-H. Han, G. Karypis, and V. Kumar, *"Scalable Parallel Data mining for Association Rules,"* Proc. ACM SIGMOD 1997, May 1997.

[10] E-H. Han, G. Karypis, and V. Kumar, *"Scalable Parallel Data mining for Association Rules,"* IEEE Trans. Data and Knowledge Eng.,vol. 12, no. 3, May/June 2000.

[11] H. Cokrowijoyo, D. Taniar, *A framework for mining association rules in Data Warehouses,* Proc. IDEAL 2004, Lecture Notes in Computer Science, vol. 3177, Springer, Berlin, 2004, pp. 159–165.

[12] L. Dehaspe, L. Raedt, *Mining association rules in multiple relations*, Proc. ILP'97, Lecture Notes in Computer Science, vol. 1297, Springer, Berlin, 1997, pp. 125–132.

Authors Biography

B. Santhosh Kumar received Bachelor degree in Computer Science & Engg from Bharathiar University and Master of Engineering in Computer Science & Engg from Anna University Trichy. He has published several papers in Data mining. Currently he is working as a lecturer- Computer Science & Engg in CSI College of Engg, Ketti.

K.V. Rukmani received Master of Computer Application from Mother Teresa Womens' University and Master of Engineering in Computer Science & Engg from Anna University Coimbatore. She has published several papers in Data mining. Currently she is working as an Asst Professor and HoD in charge- Computer Science & Engg in CSI College of Engg, Ketti.

Int. J. of Advanced Networking and Applications
Volume:01, Issue:06, Pages: 400-404 (2010)

Implementation of Web Usage Mining Using APRIORI and FP Growth Algorithms

B.Santhosh Kumar
Department of Computer Science, C.S.I. College of Engineering, Ketti- 643 215. The Nilgiris.
Email: b.santhoshkumar@csice.edu.in

K.V.Rukmani
Department of Computer Science, C.S.I. College of Engineering, Ketti- 643 215. The Nilgiris.
Email: rukmani_arun@csice.edu.in

ABSTRACT

Web Usage Mining is the application of data mining techniques to discover interesting usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. Usage data captures the identity or origin of Web users along with their browsing behavior at a Web site. Web usage mining itself can be classified further depending on the kind of usage data considered. They are web server data, application server data and application level data. Web server data correspond to the user logs that are collected at Web server. Some of the typical data collected at a Web server include IP addresses, page references, and access time of the users and is the main input to the present Research. This Research work concentrates on web usage mining and in particular focuses on discovering the web usage patterns of websites from the server log files. The comparison of memory usage and time usage is compared using Apriori algorithm and Frequent Pattern Growth algorithm.

Keywords : Apriori, Data cleaning, FP Growth, FP-tree, Web Usage Mining,

Date of Submission: 16, February 2010 Date of Acceptance: 08, A pril 2010

1. INTRODUCTION

The Web is a huge, explosive, diverse, dynamic and mostly unstructured data repository, which supplies incredible amount of information, and also raises the complexity of how to deal with the information from the different perspectives of view, users, web service providers, business analysts. The users want to have the effective search tools to find relevant information easily and precisely. The Web service providers want to find the way to predict the users' behaviors and personalize information to reduce the traffic load and design the Web site suited for the different group of users. The business analysts want to have tools to learn the user/consumers' needs. All of them are expecting tools or techniques to help them satisfy their demands and/or solve the problems encountered on the Web. Therefore, Web mining becomes a popular active area and is taken as the research topic for this investigation.

Web Usage Mining [4], [5] is the application of data mining techniques to discover interesting usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. Usage data captures the identity or origin of Web users along with their browsing behavior at a Web site. Web usage mining itself can be classified further depending on the kind of usage data considered. They are web server data, application server data and application level data. Web server data correspond to the user logs that are collected at Web server. Some of the typical data collected at a Web server include IP addresses, page references, and access time of the users and is the main input to the present Research. This Research work concentrates on web usage mining and in particular focuses on discovering the web usage patterns of websites from the server log files.

1.1 APRIORI ALGORITHM

The present Research work is designed to operate on log files. The algorithm [7] attempts to find subsets which are common to at least a minimum number C (the cutoff, or confidence threshold) of the item sets. The system operates in the following three modules.

- Preprocessing module
- Apriori or FP Growth Algorithm Module
- Association Rule Generation
- Results

The preprocessing module converts the log file, which normally is in ASCII format, into a database like format, which can be processed by the Apriori algorithm.

The second module is performed in two steps.

- Frequent Item set generation
- Rules derivation

1.2 FP GROWTH ALGORITHM:

The FP Growth algorithm operates in the following four modules.

- Preprocessing module
- FP Tree an FP Growth Module
- Association Rule Generation
- Results

The preprocessing modules convert the log file, which normally is in ASCII format, into a database like format, which can be processed by the FP Growth algorithm.

The 2nd module is performed in two steps.

- FP Tree generation
- Applying FP Growth to generate association rules

FP tree is a compact data structure that stores important, crucial and quantitative information about frequent patterns.

The main components of FP tree are:

- It consists of one root labeled as "root", a set of item prefix sub-trees as the children of the root, and a frequent-item header table.
- Each node in the item prefix sub-tree consists of three fields: item-name, count, and node-link, where item-name registers which item this node represents, count registers the number of transactions represented by the portion of the path reaching this node, and node-link links to the next node in the FP-tree carrying the same item-name, or null if there is none.
- Each entry in the frequent-item header table consists of two fields, (1) item-name and (2) head of node-link, which points to the first node in the FP-tree carrying the item-name.

Second, an FP-tree-based pattern-fragment growth mining method is developed, which starts from a frequent length-1 pattern (as an initial suffix pattern), examines only its conditional-pattern base (a "sub-database" which consists of the set of frequent items co-occurring with the suffix pattern), constructs its (conditional) FP-tree, and performs mining recursively with such a tree. The pattern growth is achieved via concatenation of the suffix pattern with the new ones generated from a conditional FP-tree. Since the frequent item set in any transaction is always encoded in the corresponding path of the frequent-pattern trees, pattern growth ensures the completeness of the result.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The Research work was initiated through a system study and analysis phase, where significant study was conducted to understand the existing system. Using Apriori algorithm for weblog mining is a novel technique. The explosive growth of the World Wide Web (WWW) in recent years has turned the web into the largest source of available online data.

- Situations like several unrelated topics in a single web page may lead to confusion and make harder to reach the information that the visitors are looking for.
- The design of the whole site (interface, content, structure, usability, etc.) is one of the most important aspects for any institution that wants to survive in the cyberspace.
- Understand the way user browses the site and find out which is the most frequent used link and pattern of using the features available in the site.

All these information is available online but are hidden for the users. Presently, there is no powerful that can analyze this hidden information and this Research work uses web usage mining (WUM) Apriori based approach for analyzing the visitor browsing behavior.

2.2 LIMITATIONS OF APRIORI ALGORITHM

Apriori algorithm, in spite of being simple, has some limitation. They are,

- It is costly to handle a huge number of candidate sets. For example, if there are 10^4 frequent 1-item sets, the Apriori algorithm will need to generate more than 10^7 length-2 candidates and accumulate and test their occurrence frequencies. Moreover, to discover a frequent pattern of size 100, such as $\{a_1, \ldots, a_{100}\}$, it must generate $2^{100} - 2 \sim 10^{30}$ candidates in total. This is the inherent cost of candidate generation, no matter what implementation technique is applied.
- It is tedious to repeatedly scan the database and check a large set of candidates by pattern matching, which is especially true for mining long patterns.

In order to overcome the drawback inherited in Apriori, an efficient FP-tree based mining method, FP-growth, which contains two phases, where the first phase constructs an FP tree, and the second phase recursively Researches the FP tree and outputs all frequent patterns.

2.3 PROPOSED SYSTEM

The main goal of the proposed system is to identify usage pattern from web log files of a website. Apriori [7] and FP Growth Algorithm is used for this purpose. The main goal of the proposed system is to identify usage pattern from web log files of a website. For this purpose, the usage of apriori and FP Growth algorithms are proposed. Both are influential algorithms for mining frequent item sets for boolean association rules [1], [9]. In computer science and data mining, Apriori is a classic algorithm for learning association rules [10]. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or

details of a website frequentation).

Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k - 1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The key concepts in this algorithm are

- Frequent Item sets: The sets of item which has minimum support (denoted by Lifor ith-Item set).
- Apriori Property: Any subset of frequent item set must be frequent.
- Join Operation: To find L_k, a set of candidate k-item sets is generated by joining L_{k-1} with itself.

The advantages of using apriori algorithm are

- Uses large item set property.
- Easily parallelized
- Easy to implement

2.4 ADVANTAGES OF FP GROWTH ALGORITHM

The major advantages of FP-Growth algorithm is,
- Uses compact data structure
- Eliminates repeated database scan

FP-growth is an order of magnitude faster than other association mining algorithms and is also faster than tree-Researching. The algorithm reduces the total number of candidate item sets by producing a compressed version of the database in terms of an FP-tree. The FP-tree stores relevant information and allows for the efficient discovery of frequent item sets.

The algorithm consists of two steps:
- Compress a large database into a compact, *Frequent-Pattern tree* (FP-tree) structure
  – highly condensed, but complete for frequent pattern mining and avoid costly database scans
- Develop an efficient, FP-tree-based frequent pattern mining method (FP-growth)
  – A divide-and-conquer methodology: decompose mining tasks into smaller ones and avoid candidate generation: sub-database test only

2.5 ADVANTAGE OF FP-TREE STRUCTURE

The most significant advantage of the FP-tree is that the algorithm scans the tree only twice. Apart from this major advantage, the others include,

- Completeness:
  The FP-tree contains all the information related to mining frequent patterns (given the min_support threshold)

- Compactness:
  – The size of the tree is bounded by the occurrences of frequent items
  – The height of the tree is bounded by the maximum number of items in a transaction

Three major steps performed are starting the processing from the end of list L:

- Construct conditional pattern base for each item in the header table
- Construct conditional FP-tree from each conditional pattern base
- Recursively mine conditional FP-trees and grow frequent patterns obtained so far. If the conditional FP-tree contains a single path, simply enumerate all the patterns

3. SYSTEM DESIGN AND DEVELOPMENT

3.1 INPUT DESIGN

The input design is the process of converting user-oriented input to a computer-based format. The goal of the input design is to make the data entry easier, logical and error free. In the present Research work, the input is taken from the web log file. The web log file has the extension .log and contains ASCII characters. A log file is a text file in which every page request made to the web server is recorded. For each request the corresponding log file contains the following information:

IP address of the computer making the request;
- User ID, (this field is not used in most cases);
- date and time of the request;
- a status field indicating if the request was successful;
- size of the file transferred;
- Referring URL, that is, the URL of the page which contains the link that generated the request; name and version of the browser being used.

This information can be used to reconstruct the user navigation sessions within the site from which the log data originates. In an ideal scenario, each user is allocated an unique IP address whenever an access is made to a given web site. Moreover, it is expected that a user visits the site more than once, each time possibly with a different goal in mind. Once the domain-dependent data transformation phase is completed, the resulting transaction data must be formatted to conform to the data model of the appropriate data mining task. For instance,

Volume:01, Issue:06, Pages: 400-404 (2010)

the format of the data for the [2],[12] discovery task may be different than the format necessary for mining sequential patterns. Finally, a query mechanism will allow the user (analyst) to provide more control over the discovery process by specifying various constraints.

2.5 OUTPUT DESIGN

The main objective of any system is the generation of reports. It has various uses. Some of them are,

- For the users, reports provide source of information required.
- They provide permanent hard copy of the results of transactions.

Careful consideration has been given in the designing of the reports as it helps in decision-making process. In the present work, the performance of the system is judged using two metrics. The first one is the amount of memory used and the second one is the time taken for the algorithm to create the association rules [3], [9]. It was found that the FP Growth algorithm is fast and uses only very small amount of memory when compared with Apriori algorithm.

4. CODING STANDARDS

4.1 APRIORI ALGORITHM

In computer science and data mining, Apriori is a classic algorithm for learning association rules [6],[11]. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k - 1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The key concepts in this algorithm are

- Frequent Item sets: The sets of item which has minimum support (denoted by Lifor ith-Item set).
- Apriori Property: Any subset of frequent item set must be frequent.
- Join Operation: To find L_k, a set of candidate k-item sets is generated by joining L_{k-1} with itself.

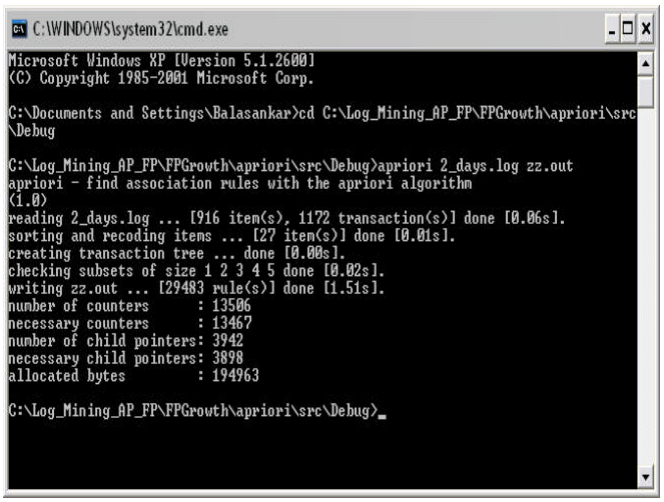
The advantages of using Apriori algorithm are

- Uses large item set property.
- Easily parallelized
- Easy to implement

The Apriori algorithm is an efficient algorithm for finding all frequent item sets. It implements level-wise search using frequent item property and can be additionally optimised. The Apriori algorithm used is given below.

- L_k: Set of frequent item sets of size k (with min support)
- C_k: Set of candidate item set of size k (potentially frequent item sets)

L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) **do**
  C_{k+1} = candidates generated from L_k;
  for each transaction t in database do
    increment the count of all candidates in C_{k+1} that are contained in t
  L_{k+1} = candidates in C_{k+1} with min_support
return $\cup_k L_k$;



4.2 FP GROWTH ALGORITHM

Definition of FP-Tree

FP-tree is a frequent pattern tree consists of one root labeled as "null". It has a set of *item prefix sub trees* as the children of the root, and a *frequent-item header table*. Each node in the *item prefix sub trees* has three fields:

- item-name to register which item this node represents,
- count, the number of transactions represented by the portion of the path reaching this node, and
- Node-link that links to the next node in the FP-tree carrying the same item-name, or null if there is none.

Each entry in the *frequent-item header table* has two fields,

- item-name, and
- head of node-link that point to the first node in the FP-tree carrying the item-name.

5. CONCLUSION

Web usage mining is the application of data mining techniques to discover usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. Web usage mining consists of three phases, namely preprocessing, pattern discovery, and pattern analysis. This Research work implements each of these phases. One of the algorithms which is very simple to use and easy to implement is the Apriori algorithm. This algorithm is used in the present Research work to generate association rules that associates the usage pattern of the clients for a particular website. The output of the system was in terms of memory usage and speed of producing association rules.

The main drawback of Apriori algorithm is that the candidate set generation is costly, especially if a large number of patterns and/or long patterns exist. The main drawback of FP-growth algorithm is the explosive quantity of lacks a good candidate generation method. Future research can combine FP-Tree with Apriori candidate generation method to solve the disadvantages of both apriori and FP-growth. In future the algorithm can be extended to web content mining, web structure mining, etc. The work can also be extended to extract information from image files.

6. ACKNOWLEDGEMENTS

"Implementation of web usage mining using Apriori and FP growth algorithms" research is supported by the management of ***C.S.I College of Engineering, Ketti***, The Nilgiris, whose support we are pleased to acknowledge. We are also grateful to our colleagues in CSICE for useful discussions, and thank to our beloved CSICE students.

REFERENCES

[1] Kotsiantis S, Kanellopoulos D., *Association Rules Mining: A Recent Overview, GESTS International Transactions on Computer Science and Engineering*, Vol.32 (1), 2006, pp. 71-82

[2] Agrawal R, Srikant R., "*Fast Algorithms for Mining Association Rules*", VLDB. Sep 12-15 1994, Chile, 487-99, pdf, ISBN 1-55860-153-8.

[3] Mannila H, Toivonen H, Verkamo A I., "*Efficient algorithms for discovering association rules." AAAI Workshop on Knowledge Discovery in Databases* (SIGKDD). July 1994, Seattle, 181-92.

[4] Tan, P. N., M. Steinbach, V. Kumar, "*Introduction to Data Mining"*, Addison-Wesley, 2005, 769pp.

[5] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques With Java Implementation*, 2nd ed. San Mateo, CA: Morgan Kaufmann, 2005.

[6] P. Becuzzi, M. Coppola, and M. Vanneschi, "*Mining of Association Rules in Very Large Databases: A Structured Parallel Approach,"* Proc. Europar-99, vol. 1685, pp. 1441-1450, Aug. 1999.

[7] R. Jin and G. Agrawal, *"An Efficient Implementation of Apriori Association Mining on Cluster of SMPs,"* Proc. Workshop High Performance Data Mining (IPDPS 2001), Apr. 2001.

[8] J. Han and M. Kamber, *"Data Mining: Concepts and Techniques"* .Morgan Kaufmann Publishers, 2000.

[9] E.-H. Han, G. Karypis, and V. Kumar, *"Scalable Parallel Data mining for Association Rules,"* Proc. ACM SIGMOD 1997, May 1997.

[10] E-H. Han, G. Karypis, and V. Kumar, *"Scalable Parallel Data mining for Association Rules,"* IEEE Trans. Data and Knowledge Eng.,vol. 12, no. 3, May/June 2000.

[11] H. Cokrowijoyo, D. Taniar, *A framework for mining association rules in Data Warehouses,* Proc. IDEAL 2004, Lecture Notes in Computer Science, vol. 3177, Springer, Berlin, 2004, pp. 159–165.

[12] L. Dehaspe, L. Raedt, *Mining association rules in multiple relations*, Proc. ILP'97, Lecture Notes in Computer Science, vol. 1297, Springer, Berlin, 1997, pp. 125–132.

Authors Biography

B. Santhosh Kumar received Bachelor degree in Computer Science & Engg from Bharathiar University and Master of Engineering in Computer Science & Engg from Anna University Trichy. He has published several papers in Data mining. Currently he is working as a lecturer- Computer Science & Engg in CSI College of Engg, Ketti.

K.V. Rukmani received Master of Computer Application from Mother Teresa Womens' University and Master of Engineering in Computer Science & Engg from Anna University Coimbatore. She has published several papers in Data mining. Currently she is working as an Asst Professor and HoD in charge- Computer Science & Engg in CSI College of Engg, Ketti.

IMPLEMENTASI ALGORITMA FP – GROWTH DAN ASSOCIATION RULE PADA PEMILIHAN BIDANG MINAT PROGRAM STUDI TEKNIK INFORMATIKA

Nama Mahasiswa **:Tri Rahayu Wulandari**
NIM **:1210651078**
Jurusan **:Teknik Informatika**
Dosen Pembimbing 1 **:Lutfi Ali Muharrom, S. Si, M.Si**
Dosen Pembimbing 2 **: Daryanto S.Kom, M.Kom**

Abstrak

Pada saat ini, mahasiswa di Jurusan Teknik Informatika Universitas Muhammadiyah Jember belum memiliki sistem untuk membantu mahasiswa menentukan bidang minat. Pemilihan bidang minat ini sangat diperlukan karena sangat berkaitan dengan pengambilan Tugas Akhir dari mahasiswa itu sendiri. Sistem ini akan dibangun berdasarkan hasil analisa terhadap sejumlah data – data sebelumnya. Algoritma FP – Growth adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sekumpulan data. Algoritma *FP – Growth generate candidate* tidak dilakukan karena *FP - Growth* menggunakan konsep pembangunan *tree* dalam pencarian *frequent itemset.*

Dari paparan di atas diharapkan dengan adanya model sistem pendukung keputusan yang dimilki oleh Jurusan dapat membantu untuk menentukan pola nilai histori matakuliah dalam penentuan matakuliah pilihan yang akan ditawarkan dengan menerapkan teknik *data mining* dengan menggunakan *Association Rule* dan Algoritma *FP – Growth.*

Kata Kunci : Algoritma *Fp – growth, association rule mining, Fp – Tree*

Abstrak

At this time, a student in the Department of Information Engineering University of Muhammadiyah Jember not have a system to help students determine their areas of interest. Selection of areas of interest is very necessary because it is related to decision-Final of the students themselves. The system will be built based on the analysis of some data - previous data. Algorithm FP - Growth is one of the alternative algorithm that can be used to specify the data set that appears most frequently (frequent itemset) in a set of data. Algorithm FP - Growth generate candidate was not done because of FP - Growth uses the concept of development in the search for frequent itemset tree.

From the above explanation is expected by the model of decision support system that is owned by the Department can help to determine the pattern of historical value in determining the course that will be offered optional subjects by applying data mining techniques using Association Rule and algorithm FP - Growth.

Keywords: Algorithm *Fp - growth, association rule mining, fp - Tree,*

1.1 Latar Belakang

Algoritma FP – Growth adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sekumpulan data. Algoritma *FP – Growth generate candidate* tidak dilakukan karena *FP - Growth* menggunakan konsep pembangunan *tree* dalam pencarian *frequent itemset.*

Dalam model ini, hasil rekomendasi penetuan matakuliah pilihannya didapat dengan menggunakan metode *data mining* yaitu *Association Rule* dengan pendekatan algoritma *FP – Growth*. Untuk mendapatkan pola ini digunakan pendekatan *FP – Growth* untuk mempercepat mendapatkan *ekstrak frequent itemset*. Pada akhirnya model sistem pendukung keputusan ini dapat digunakan untuk mendapatkan rekomendasi penentuan bidang minat berdasarkan data acuan yang sesuai dengan kebijakan yang dikeluarkan pihak jurusan.

Pada saat ini, mahasiswa di Jurusan Teknik Informatika Universitas Muhammadiyah Jember belum memiliki sistem untuk membantu mahasiswa menentukan bidang minat. Pemilihan bidang minat ini sangat diperlukan karena sangat berkaitan dengan pengambilan Tugas Akhir dari mahasiswa itu sendiri. Sistem ini akan dibangun berdasarkan hasil analisa terhadap seumlah data – data sebelumnya

1.2 Tujuan

Menghasilkan sebuah model sistem pendukung keputusan dalam penentuan matakuliah pilihan di Jurusan Teknik Informatika dengan data acuan dari kebijakan yang dinamis, dan membuat sistem pencarian pola asosiasi konsentrasi dimana akan ditemukan aturan - aturan (*rule*).

2.1 Penggalian Data

Penggalian data (data mining) adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan didalam basis data. Penggalian data adalah tentang memecahkan suatu masalah dengan menganalisis data yang telah ada, dan sudah di definiskan sebagai proses pola dalam data, dimana pola yang didapat harus memiliki beberapa keuntungang (Written dan Frank,2005)

2.2 Association Rule

Association rule merupakan suatu proses pada *data mining* untuk menentukan semua aturan *assosiatif* yang memenuhi syarat minimum untuk *support* (*minsup)* dan *confidence* (*mincof*) pada sebuah database. Menurut Agrawal dan Srikant dalam (Ridowati, 2003), masalah dalam pencarian seluruh *association rule* dapat dibagi menjadi dua, yaitu bagaimana menemukan seluruh barang – barang (*itemset*) yang memiliki nilai penunjang (*support)* di atas nilai penunjang terkecil yang telah ditentukan. Nilai penunjang untuk sebuah barang adalah jumalah transaksi yang terdapat dalam *itemset*, *itemset* dengan nilai terkecil disebut *large Itemset.* Masalah yang kedua adalah bagaimana menggunakan large itemsetbuntuk menghasilkan aturan – aturan yang diperlukan.

Aturan yang hendak dihasilkan adalah dalam bentuk $a \rightarrow (I–a)$, yaitu perbandingan antara *support* (I) dengan *support* (a) haruslah lebih besar dari *mincof* (*minimum confidence*). Misakan $[I] = \{I_1, I_2, \ldots I_m\}$ adalah himpunan literal yang disebut barang. Himpunan *item – item* disebut dengan *Itemset.* D adalah himpunan transaksi, dimana setiap transaksi T itemset sehingga TE, I.

3.1 Algoritma FP - Growth

Algoritma Fp – Growth menggunakan konsep pembangunan tree yang biasa disebut FP – *Tree*, dalam pencarian *Frequent Itemset* bukan menggunakan generate candidate seperti yang dilakukan pada algoritma apriori. Dengan menggunakan konsep tersebut proses algoritma fp – growth menjadi lebih cepat dari pada algoritma apriori. Metode yang digunakan FP – growth ada tiga tahapan utama, yakni : 1 tahapan pembangkitan *Condational Pattern Base*, 2 tahapan pembangkitan *Condational FP – Tree* dan ke 3 Pencarian *Frequent Itemset.*

Analisis dari data keranjang dapat digunakan untuk mendapatkan aturan asosiasi dari kombinasi IPK mahasiswa per KBK dengan bidang minat yang diambil oleh mahasiswa seperti contoh tabel dibawah ini :

Tabel 1 Data mahasiswa

ID	KBJ	SBC	RPL	BM
0910651010	3,20	3,39	3,21	RPL
0910651011	2,79	3,05	3,00	RPL
0910651012	3,16	3,16	3,42	KBJ
0910651025	2,93	3,14	3,35	KBJ
0910651038	2,93	3,01	3,07	SBC
0910651040	3,00	3,13	3,28	SBC
0910651041	3,13	3,07	3,50	KBJ
0910651045	2,90	3,13	3,50	RPL
0910651050	2,93	3,16	3,42	SBC
0910651104	2,95	3,58	2,85	RPL

Untuk memudahkan analisis, maka IPK masing – masing KBK dibuat menjadi 3 interval :

IPK menjadi 3 interval :

IPK KBK KBJ
(minimal – 2,98) = KBJ 1
(3,00 – 3,19) = KBJ 2
(maximal – 3,20) = KBJ 3
IPK KBK SBC
(minimal - 2,98) = SBC 1
(3,00 – 3,38) = SBC 2
(maximal – 3,39) = SBC 3
IPK KBK RPL
(minimal – 3,00) = RPL 1
(3,01 – 3,41) = RPL 2
(maximal – 3,42) = RPL 3

Tabel 2 Data transaksi Mentah

ID	KBJ	SBC	RPL	BM
1010	KBJ-2	SBC-3	RPL-2	RPL
1011	KBJ-1	SBC-2	RPL-2	RPL
1012	KBJ-2	SBC-2	RPL-3	KBJ
1025	KBJ-1	SBC-2	RPL-2	KBJ
1038	KBJ-1	SBC-2	RPL-2	SBC
1040	KBJ-2	SBC-2	RPL-2	SBC
1041	KBJ-2	SBC-2	RPL-2	KBJ
1045	KBJ-1	SBC-2	RPL-2	RPL
1050	KBJ-1	SBC-2	RPL-3	SBC
1104	KBJ-1	SBC-2	RPL-1	RPL

Berikut adalah proses pencarian Frequent Itemset. Langkah pertama adalah menentukan *minimum support* dan *minimum confidence (Minsup),* yaitu dengan *minsup* (0,3) dan *mincof* (0,9)

Tabel 3 Kemunculan Item

Itemset	*Support*
KBJ-2	4
SBC-3	1
RPL-2	7
KBJ	3
KBJ-1	6
SBC-2	9
RPL-1	1
RPL	4
KBJ-3	0

SBC-1	0
RPL-3	2
SBC	3

Setelah diperoleh *Frequent List*, kemudian ubah item sesuai frequent list dengan menghilangkan item SBC-3 , RPL-1, KBJ-3, SBC-1, RPL-3. Dan kemudian di urutkan dari yang terbesar pada setiap item transaksi.

Tabel 4 *Frequent List*

Itemset	*Support*
SBC-2	9
RPL-2	7
KBJ-1	6
KBJ-2	4
RPL	4
KBJ	3
SBC	3

Tabel 5 transaksi yang telah disesuaikan dengan *frequent list*

ID	ITEM
0910651010	RPL-2,KBJ-2,RPL
0910651011	SBC-2,RPL-2,KBJ-1,RPL
0910651012	SBC-2,KBJ-2,KBJ
0910651025	SBC-2,RPL-2,KBJ-1,KBJ
0910651038	SBC-2,RPL-2,KBJ-1,SBC
0910651040	SBC-2,RPL-2,KBJ-2,SBC
0910651041	SBC-2,RPL-2,KBJ-2,KBJ
0910651045	SBC-2,RPL-2,KBJ-1,RPL
0910651050	SBC-2,KBJ-1,SBC
0910651104	SBC-2,KBJ-1,RPL

Pembuatan FP – Tree diawali dari TID 1 sampai TID 10.

Sehingga kita menemukan daftar *Frequent Itemset* berdasarkan hubungan akhiran atau *Suffix.*

Tabel 6 *Suffix* atau Akhiran

Suffix	**Frequent Itemset**
{SBC}	{KBJ-1,SBC}, {SBC-2,SBC},{KBJ-2,SBC},{SBC-2,SBC}
{KBJ}	{KBJ-2,KBJ},{RPL-2,KBJ-2,KBJ},{SBC-2,KBJ-2,KBJ},{RPL-2,KBJ-1,KBJ},{SBC-2,KBJ-1,KBJ},{SBC-2,KBJ}
{RPL}	{KBJ-2,RPL},{RPL-2,KBJ-2,RPL},{KBJ-1,RPL},{RPL-2,KBJ-1,RPL},{SBC-2,KBJ-1,RPL},{SBC-2,RPL}
{KBJ-2}	{RPL-2,KBJ-2},{SBC-2,RPL-2,KBJ-2},{SBC-2,KBJ-2}
{KBJ-1}	{RPL-2,KBJ-1},{SBC-2,KBJ-1},{SBC-2,RPL-2,KBJ-1},{SBC-2,KBJ-1}
{RPL-2}	{SBC-2,RPL-2}
{SBC-2}	{SBC-2}

Proses Pencarian *Association Rule*

Proses pencarian association rule ini memanfaatkan minimum confidence yang telah ditentukan ,yaitu 0,9 atau 90%, dimana S→ $(I - s)$

R1 = IPK KBK RPL = (RPL-2) →IPK KBK KBJ = (KBJ-2)

Confidence = Support Count {IPK KBK RPL = (RPL – 2) ; IPK KBK KBJ (KBJ – 2) }/ *Support Count* {IPK KBK RPL = (RPL – 2)} = 4/7 = 0,6

R2 = IPK KBK RPL = (RPL – 2) → IPK KBK RPL

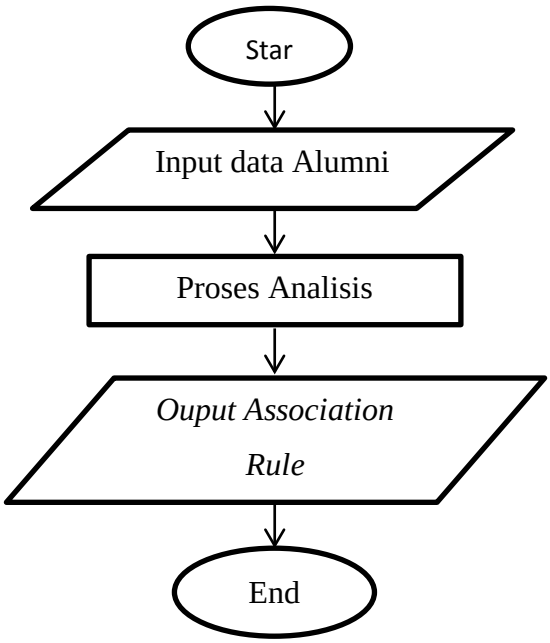
Confidence = Support Count {IPK KBK RPL = (RPL -2) ; RPL} / Support Count {IPK KBK RPL = (RPL -2) = 4/7 = 0,6

R3 = IPK KBK KBJ = (KBJ – 2) → RPL

Confidence = Support Count {IPK KBK KBJ = (KBJ – 2) ; RPL} / Support Count {IPK KBK KBJ = (KBJ -2) = 4/4 = 1

Jadi contoh rule yang terpilih adalah R3 yang ditandai dengan cetakan tebal, rule yang tidak terpilih atau ditolak dikarenakan tidak memenuhi syarat *minimum confidence.*

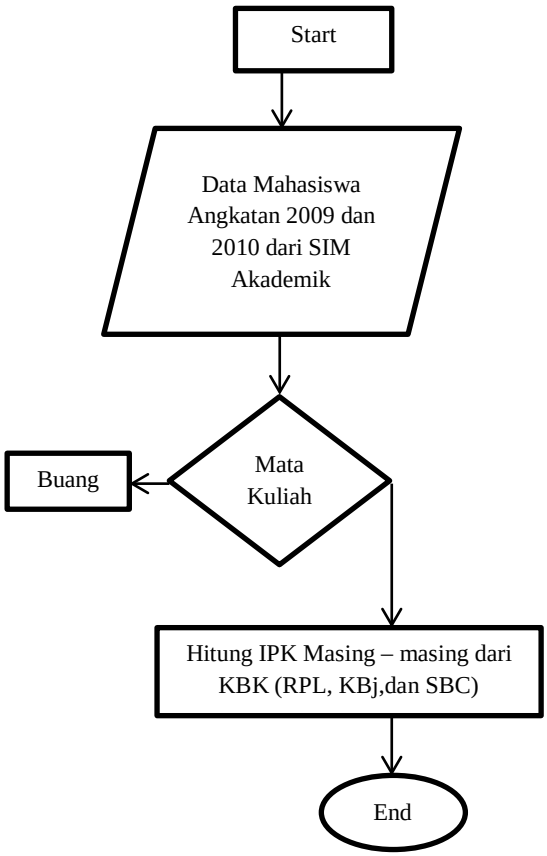
3.2 Flowchart Sistem



Sistem bidang minat ini menggunakn algoritma *FP – Growth* dalam menentukan bidang minat mahasiswa. Langkah – langkah dalam memproses dengan menggunkan algoritma *FP – Growth* ini adalah, yang pertama, mencari *rule* terlebih dahulu, yaitu dengan memsaukkan input data alumni yang telah terpilih, kemudian algoritma *FP – Growth* ini akan digunakn untuk menganalisis data. Saat memasukkan

data mahasiswa yang melalui tahap pro – processing atau dari data mentah sampai rule yang terpilih maka saat itu pula algoritma *FP – Growth* memulai melakukan pekerjaan dengan 2 tahapan, yaitu dengan melakukan pembentukan *frequent itemset* dan proses pembentukan *Association Rule*, sehingga keluarlah *Ouput Association Rule*.

3.3Perancangan Data

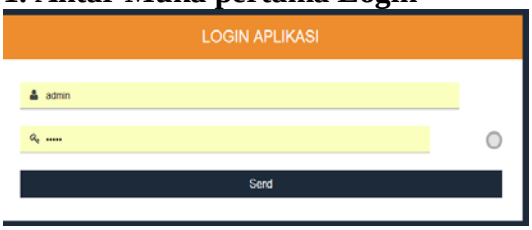


Berikut adalah penjelasan dari langkah – langkah perancangan data dia atas :

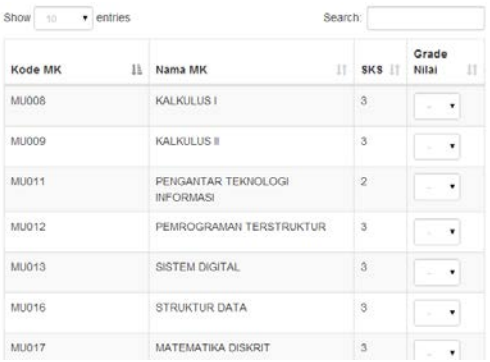
1. Pertama kita mengambil data mahasiswa angkatan tahun 2009 dan 2010 dari SIM akademik, kemudian data tersebut tidak dapat langsung digunakan karena kita harus terlebih dahulu memisah antara matakuliah umum dan matakuliah yang diwajib, karena kita hanya memerlukan matakuliah wajib saja,
2. Selanjutnya, setelah selesai memilih matakuliah yang diwajibkan kita akan memulai menghitung IPK dari KBK RPL, KBK KBJ, dan KBK dari SBC.
3. Kemudian ouput yang dihasilkan adalah ID dari mahasiswa dengan NIM 0910651010 mendapatkan IPK dari KBK RPL (3,21), IPK dari KBK KBJ (3,20) , dan IPK dari KBK SBC (3,39).

4.1 Pengujian

1. Antar Muka pertama Login



2.Antar muka Input data dengan seluruh Matakuliah yang telah ditempuh baik mata kuliah yang terdapat pada kelompok SBC,KBJ, maupun RPL yang menjadi acuan dalam penentuan penjurusan.



3.**Proses Antarmuka Kombinasi IPK**

PROSES KOMBINASI

No	NIM	Nama Mahasiswa	KBJ	SBC	RPL	HASIL KOMBINASI IPK
1	910651001	FARIDA RAHAYU NINGTIAS	KBJ 3	SBC 3	RPL 3	SBC
2	910651002	Nur Arif Fitriadi	KBJ 2	SBC 2	RPL 3	KBJ
3	910651003	Zuhri Nurhuda	KBJ 3	SBC 2	RPL 2	KBJ
4	910651004	RESARDI ESA PRASETYO	KBJ 3	SBC 2	RPL 2	KBJ
5	910651009	Bertha Nindri Widianarta	KBJ 3	SBC 3	RPL 2	SBC

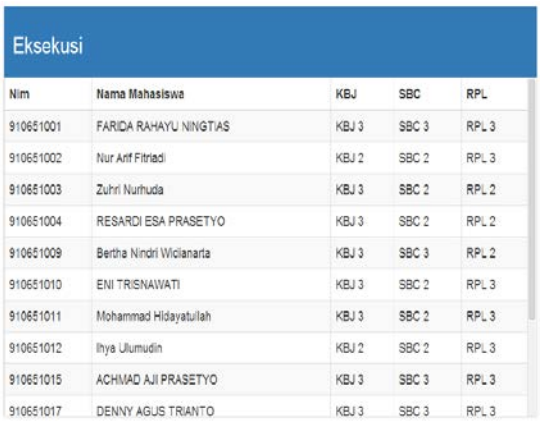
4.Proses antar muka Frekuensi itemset

Merupakan pencarian frekuensi berdasarkan minsuuport dan minconfidence yang telah diinputkan.

Frequesi

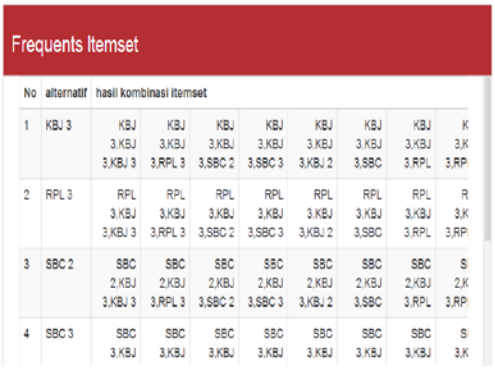
No	Itemset	Frequensi
1	KBJ 3	8
2	RPL 3	8
3	SBC 2	7
4	SBC 3	7
5	KBJ 2	6
6	SBC	5
7	RPL	4
8	RPL 1	3
9	RPL 2	3

Proses Eksekusi Itemset


Eksekusi

Nim	Nama Mahasiswa	KBJ	SBC	RPL
910651001	FARIDA RAHAYU NINGTIAS	KBJ 3	SBC 3	RPL 3
910651002	Nur Arif Fitriadi	KBJ 2	SBC 2	RPL 3
910651003	Zuhri Nurhuda	KBJ 3	SBC 2	RPL 2
910651004	RESARDI ESA PRASETYO	KBJ 3	SBC 2	RPL 2
910651009	Bertha Nindri Widianarta	KBJ 3	SBC 3	RPL 2
910651010	ENI TRISNAWATI	KBJ 3	SBC 2	RPL 3
910651011	Mohammad Hidayatullah	KBJ 3	SBC 2	RPL 3
910651012	Ihya Ulumudin	KBJ 2	SBC 2	RPL 3
910651015	ACHMAD AJI PRASETYO	KBJ 3	SBC 3	RPL 3
910651017	DENNY AGUS TRIANTO	KBJ 3	SBC 3	RPL 3

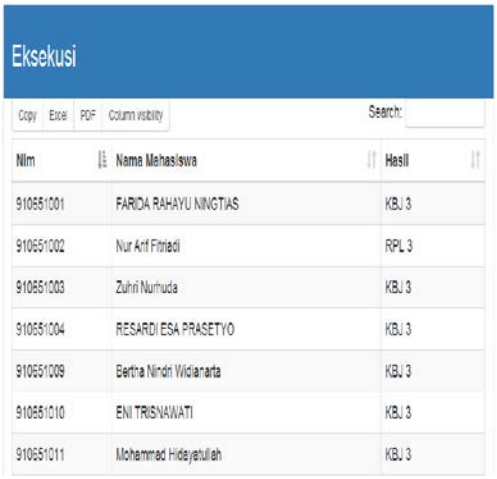
Hasil Suffix dari Frequent Itemset


Frequents Itemset

No	alternatif	hasil kombinasi itemset
1	KBJ 3	KBJ 3,KBJ 3 / KBJ 3,RPL 3 / KBJ 3,SBC 2 / KBJ 3,SBC 3 / KBJ 3,KBJ 2 / KBJ 3,SBC / KBJ 3,RPL / K 3,RP
2	RPL 3	RPL 3,KBJ 3 / RPL 3,RPL 3 / RPL 3,SBC 2 / RPL 3,SBC 3 / RPL 3,KBJ 2 / RPL 3,SBC / RPL 3,RPL / R 3,RP
3	SBC 2	SBC 2,KBJ 3,KBJ 3 / SBC 2,KBJ 3,RPL 3 / SBC 2,KBJ 3,SBC 2 / SBC 2,KBJ 3,SBC 3 / SBC 2,KBJ 3,KBJ 2 / SBC 2,KBJ 3,SBC / SBC 2,KBJ 3,RPL / S 2,K 3,RP
4	SBC 3	SBC 3,KBJ / SBC 3,KBJ / SBC 3,KBJ / SBC 3,KBJ / SBC 3,KBJ / SBC 3,KBJ / SBC 3,KBJ / S 3,K

Setelah melalui proses perhitungan dengan penerapan itemset (*association rule* dengan memanfaatkan *minimum confidence* yang telah di tentukan), kemudian didapat hasil akhir (yang terdapat pada menu proses eksekusi). Hasil yang dimaksud adalah sebagai berikut:

Hasil Rule yang terpilih


Eksekusi

Copy Excel PDF Column visibility Search:

Nim	Nama Mahasiswa	Hasil
910651001	FARIDA RAHAYU NINGTIAS	KBJ 3
910651002	Nur Arif Fitriadi	RPL 3
910651003	Zuhri Nurhuda	KBJ 3
910651004	RESARDI ESA PRASETYO	KBJ 3
910651009	Bertha Nindri Widianarta	KBJ 3
910651010	ENI TRISNAWATI	KBJ 3
910651011	Mohammad Hidayatullah	KBJ 3

5.1 Kesimpulan

Dari analisis dan pengujian yang telah dilakukan, maka dapat disimpulkan bahwa metode algoritma FP – Growth dapat digunakan untuk membantu para mahasiswa dalam menentukan bidang minat mereka dengan melihat nilai IPK

masing- masing mahasiswa tersebut dengan menerapkan itemset *association rule* dengan memanfaatkan *minimum confidence* yang telah di tentukan, sehingga didapat hasil akhir yang terdapat pada menu proses eksekusi.

5.2 Saran

Saran – saran untuk pengembangan selanjutnya yaitu, peneliti selanjutnya dapat membandingkan metode algoritma FP – Growth dengan metode lain seperti Forward, Channing bayes, atau Fuzzy.

Daftar Pustaka

David Samuel, *Penerapan struktur FP – Tree dan Algoritma FP – Growth dalam Optimasi Penentuan Frequent Itemset*. Institut Teknologi Bandung, 2008.

Erwin, " Analisis Market Basket dengan Algoritma Apriori dan FP – Growth " , Juli, 2009

Han, J,. Pei,. J., & Yin, Yiwen. (2004) *Mining Frequent Patterns without Candidate Generation*. Simon Fraser Universit. Netherlands: Kluwer Academic.

Kusrini and Emha Taufiq Lutfhi, Algoritma Data Mining, 1st ed., Theresia Ari Prabawati, Ed. Yogyakarta: ANDI, 2009

Santoso, Budi. 2007. Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis. GrahaIlmu. Yogyakarta.

DETECTION OF PTERYGIUM DISEASE USING FORWARD CHAINING AND VIOLA JONES ALGORITHM

Umu Hanifah, Dr. Purba Daru Kusuma S.T., M.T, Casi Setianingsih S.T,M.T
Computer Engineering Study, Faculty of Electrical & Communication Engineering – Telkom Univesity
JL. Telekomunikasi No. 1, Dayeuh Kolot, Bandung, Jawa Barat, 40257 Indonesia

***Abstract*— Eyesight is one of the most important senses for human life. Because only with our eyes, we can see and know the situations and conditions that occur around us. If there are problems or disorders that happen in our eyes, then we will feel uncomfortable and there are several diseases that can reduce the quality of vision and can cause blindness.**

In this project the author will be make an application to detect Pterygium eye disease based on the early symptoms that have been felt by the patient and find out how severely the patient affected by Pterygium disease with different levels. The stages used to determine the level of Pterygium disease is by filling all the symptoms by the patient in the application using Forward Chaining method and using an image segmentation process with Viola Jones Algorithm.

The results of using the Viola Jones algorithm that have been processed using this application have an accuracy rate of 76% by testing 50 images and the results of the images detected are 38 images, in addition there are some images that are not detected and there are some images that are not detected.

***Keywords*— Pterygium, Algoritma Viola Jones, Forward Chaining**

I. INTRODUCTION

The number of human populations worldwide has continued to grow from 6 to 7.9 billion by 2020, with 6 billion people in Africa and Asia[1]. Indonesia occupies the second highest position after Ethiopia for the world's blindness rate, reported at the Asia Pacific Academy of Opthalmology meeting in Sydney 2010. With the blindness rate of Indonesia above 1% makes the number of blindness in Indonesia is not only a health problem, but has become a problem social[2].

According to dr.Johan Hutauruk, eye specialist from the Jakarta Eye Center emphasizes that there are five eye diseases that are commonly experienced by the people of Indonesia, and one of the five diseases there is an eye disease called Pterygium. Pterygium itself is a disease with eye conditions characterized by the growth of the membrane covering the white part of the eyeball. Prevalence of Pterygium in accordance with the Clinical Management Guide Perdami, Pterygium incidence is quite high in Indonesia, on the equator reaches up to 13.1%[3].

From the background the author makes a system useful for detecting pterygium disease in the eyes of a person who will be made through the decision-making process by using expert system taken by a person or several experts with Forward Chaining method and apply digital image processing with Viola-Jones Algorithm. With this application is expected to facilitate the doctors at health facilities level 1 to be able to find out more about the level of Pterigium disease suffered by the patients and because in Indonesia no one has made this application for this disease and the doctor still uses it manually.

II. SYSTEM OVERVIEW

System overview in this research is to make a web-based applications that can detect Pterygium disease in the eyes of patient useing Viola Jones method and Expert System method. This application uses Python Programing language that has been programmed to determine the extent of eye damage caused by Pterygium disease. This application uses image to run the program as an input and to classify how severe the damage rate suffered by the user by using Forward Chaining method. The system to be built is divided into two parts: training and testing.

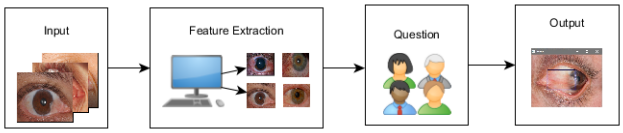


Figure. 1 Overview of The System

In Figure 1 is the overview system that starting from the collection of images containing the object to be detected. This process uses images sourced from the internet and has been validated by an eye infection specialist, Susi Heryati, dr. SpM (K). Then the system performs feature extraction to find out the special features contained in the image, then each user answered all the questions on the expert system to be asked about what is felt by the user, then the results ultimately a detectable image of where the pterygium is located and the classification statement of the severity of pterygium disease. Process flow image processing is presented in the form of flowchart as follows:

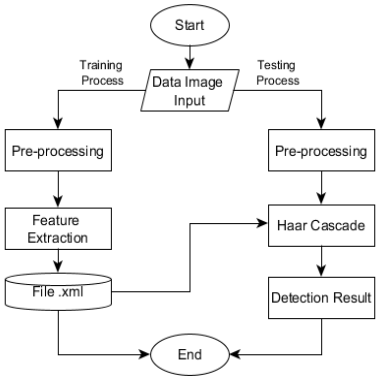


Figure. 2 Flowchart The System

In Figure 2 is a process for knowing the object that is the process by which the system can know the object in the image that we will detect. As for a brief description of the above process as follows:

1. Input image, is the image that users will select in the gallery that has been previously provided.
2. Pre-processing, digital image which then has gone through the previous process and then converted into two-color image with grayscaling process.
3. Feature Extraction (Viola Jones), to get the characteristic of the image we will detect.
4. Detecting result, the final result of the previously processed image and the location of pterygium in the image.

Based on the research, after the image is processed the user must fill in some questions to be asked about the symptoms felt about this disease and the disease is categorized into 3 parts: Level 1, Level 2, and Level 3.

III. VIOLA JONES

The Viola Jones Algorithm Method is a method proposed by Paul Viola and Michael Jones in 2001 and widely used for object processing because it has an efficient algorithm and a high degree of accuracy compared to other object detection methods[4]. There are four main keys in the method of detecting Viola Jones objects, namely:

A. Haar Like Feature

Haar-like features process images in boxes, where in a box there are several pixels. The packing is then processed and a different value (threshold) indicates the dark and bright area by subtracting the pixel value on the white area with the pixel value in the black area.

The first step of this algorithm is to convert the input image into an integral image. This is done by making each pixel equal to the entire sum of all pixels above and to the left of the concerned pixel. By sum of all pixels inside any given rectangle can be calculated using only four values[5]. Sum of the rectangle ABCD = D - (B + C) + A

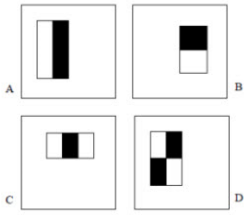


Figure. 4 Viola Jones Haar like features.

Seen in the Figure 4 that this algorithm has two features consisting of two rectangles (A and B), one feature with three rectangles (C), and one feature with rectangles (D). These features were created by Viola and Jones by modifying the Haar wavelets method, so the features are called Haar-like features.

B. Integral Image

To simplify the process of calculating feature values, the Viola-Jones algorithm uses a medium of Integral Image. Integral imagery is used as a quick and effective way to calculate an image whose value per pixel is the sum of the upper left to right bottom pixel values.

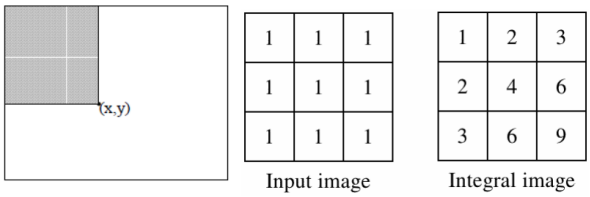


Figure. 5 Viola Jones Haar like features.

In Figure 5 is the calculation of the number of pixels in a square, can be calculated by adding the pixel value on each square. Each value in the integral image has been added to the previous square pixel value. The process of searching the value of this feature is done from the top left of the image to the bottom right corner with a shift of Δx and Δy[6].

C. AdaBoost Algorithm

In support of this goal, the weak learning algorithm is designed to select the single rectangle feature which best separates the positive and negative examples (this is similar to the approach in the domain of image database retrieval). For each feature, the weak learner determines the optimal threshold classification function, such that the minimum number of examples are misclassified[7]. Basically Adaboost is used to reduce errors in the learning process of the algorithm. The algorithm is compiled using equations:

- Given example images
  (x1, y1),.....,(xn, yn) (1)
  where yi = 0.1 for negative data and positive data respectively.

- Initialize the weight of
  $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for yi = 0.1 (2)
  respectively, where m and l are the number of negative and positive respectively.

- For t = 1,...,T:
  1. Normalize the weights,
     $$W_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}} \quad (3)$$
  2. For each feature a j classifier h j is restricted to use a single feature. Errors evaluated with respect to:
     $$W_{t,\epsilon_j} = \sum_{i=1} w_i \left| h_j(x_i) - y_i \right| \quad (4)$$

- Choose classifier, ht with the lowest error ϵ_t. Update weight:

$$W_{t,\epsilon_j} = \sum_{i=1} w_i \left| h_j(x_i) - y_i \right| \quad (5)$$

Where ei = 0 if the data xi is correctly classified, ei = 1 if otherwise and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$

- The final strong classifier is:

$$F(x) = \sum_{t=1}^{T} \alpha_t h_t(x); \quad (6)$$

where $\alpha_t = \log \frac{1}{\beta_t}$; $\alpha_j = \log \frac{1}{\beta_j}$

where:
ht (x) = weak or basic classifier
α = learning rate (learning rate)
F (x) = strong or final classifier

D. Cascade Classifier

The Cascade of Classifier combination is the last stage in the Viola Jones method. This method is a way to detect objects based on classification, where the collection of classification functions are arranged gradually to classify an image or image pieces as objects or non objects[8].

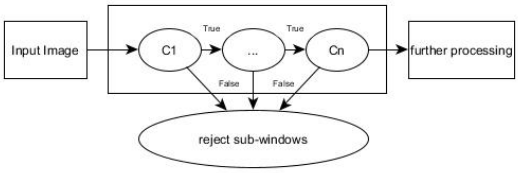


Figure. 6 Schematic depiction of a the detection cascade. A series of classifiers are applied to every sub-window.

Stages in the cascade are constructed by training classifiers using AdaBoost and then adjusting the threshold to minimize false negatives. Note that the default AdaBoost threshold is designed to yield a low error rate on the training data[8]. Here's a training process to get the special features:

a. Training

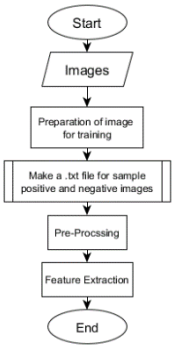


Figure. 7 Training Flowchart

Steps used to detect object is through the process of training, where it can be seen in the picture above, with the explanation as follows:

1. Preparing data training by collecting 113 positive images that measure 300x200 pixels containing objects to detect. Collecting 587 negative images that are 320x240 pixels contains a background image where the desired object is not found in the set of images.
2. After all the images are collected, the image must be grouped into 2 parts, positive images and negative images. Then we make all the images into .txt so that the application can read all existing images by doing the process of selecting objects that they want to detect using object_marker.py manually for positive images and using the help of the OpenCV library for negative images.
3. Extract the feature using the OpenCV library, namely opencv_traincascade.exe on the terminal.
4. The result of the cascade sequence that has been obtained in the convert in .xml format which is then used for image testing process to be able to detect pterygium membrane in the image.

b. Testing

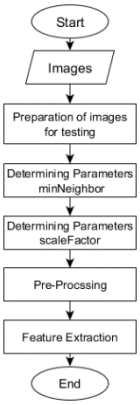


Figure. 8 Testing Flowchart

Steps used to detect object is through the process of testing, where it can be seen in the picture above, with the explanation as follows:

1. Taking a sample image of 50 of the remainder of each image used for the total of 113.
2. Determine the parameters to be used, there are 2 parameters, scaleFactor and minNeighbor. Both of these parameters are the parameters of successful adaboost detection.
3. Matching feature values from each image using the integral image method so that the calculation process in finding value matches between image testing and the results of selected features can be faster.
4. Select objects as well as determine box detection with adaboost and produce output in the form of pterygium disease detection.

As for things to note in the classification method is to set the parameters as follows.

a. minNeighbor

The parameters that determine how many neighbors each candidate rectangle should maintain it because there are too many object that detected, we have to using this parameter to reduce the adjacency between the boxes obtained on each image entered.

b. scaleFactor
Parameter specifying how much the image size is reduced at each image scale. In short. Your model has a fixed size defined during training. This means that this size of face is detected in the image if occuring. However, by rescaling the input image, you can resize a larger face towards a smaller one, making it detectable for the algorithm.

IV. EXPERT SYSTEM

Expert systems are computer-based systems that use knowledge, facts, and reasoning techniques to solve problems designed to model the ability to resolve problems like an expert. Expert is someone who has special knowledge, experience, and methods that can be used as a reference in solving a problem. This expert system can also help the activities of experts as experienced assistants and have experienced assistants and have the required knowledge[9].

A. Forward Chaining

The Forward haining method is a search method or forward-tracking technique that begins with existing information and merging rules to produce a conclusion or goal. This method takes into consideration the facts which then lead to a conclusion based on these facts[10]. Start with a set of known facts and apply the rules to generate new facts whose premises correspond to known facts, and continue the process until they reach a predetermined goal, or until no further facts can be derived according to known premises / facts.

B. Certainty Factor

In the application of expert system there is a method to solve the problem of data uncertainty, one of the methods used is certainty factor (certainty factor). Certainty Factor (CF) is one of the techniques used to overcome uncertainty in decision making. This method has its own belief value not only its premises which have a value of confidence[11].

In expressing the degree of confidence is used a value called Certainy Factor (CF) to assume the degree of confidence of an expert against a data. Here are the basic formulations of Certainy Factor:

$$CF[h,e] = MB[h,e] – MD[h,e] \quad (7)$$

Where CF is factor of certainty, is MB [h, e] = is measure of belief, measure of belief or belief level against the hypothesis (h), if given the evidence (e) between 0 and 1 and MD [h, e] = measure of disbelief, measure of distrust or degree of confidence in the hypothesis (h), if given evidence (e) between 0 and 1. Here is a description of some combination of Certainty Factor against various conditions:

The method used in this research is certainty factor method. In the face of a problem often found answers that do not have full certainty. Certanity factor (CF) shows the size of certainty to a fact or rule. Table 1 is a symptom of pterygium disease obtained from an expert and converted to a CF value:

Table 1: CF values of symptoms of pterygium disease

Symtomps	CF Value
Red eyes	0,5
Watery eyes	0,2
Sensitive to light	0,8
Lots of exposure to ultraviolet light (like from the sun)	0,3
Work outdoors (outdoor)	0,3
Feels like something in it	0,9
Vision decreases	0,8
There is a thin membrane on the eye	0,4

Calculation of CF values on the system will be done after the user selects or inputs the symptoms, then after that will enter the calculation phase of CF value. In order for the system to conclude good results it takes the value of the evidence to categorize the level of expert confidence.

Table 2: Evidence Value of Confidence Level

CF Values	Conclusions
-1.0	Definitely not
-0.8	Almost certainly not
-0.6	Most likely not
-0.4	Probably not
-0.2 - 0.2	Do not know
0.4	Maybe
0.6	Most likely
0.8	Almost certain
1.0	Certainly

The user must answer the questions provided to prove that the user is exposed to pterygium disease and the level suffered. There are several questions that will be asked about the perceived symptoms are:

Table 3: Table of Symptoms

No.	Symptomps	The Answer
1.	Do your eyes often look red?	YES/NO
2.	Do your eyes often runny?	YES/NO
3.	Are your eyes sensitive to light (glare)?	YES/NO
4.	Are your eyes often directly exposed to ultraviolet light?	YES/NO

5.	Do you work outdoors?	YES/NO
6.	Do your eyes feel like there is something in it?	YES/NO
7.	Do you feel your vision decrease?	YES/NO
8.	Are there soft membranes in your eyes?	YES/NO

After the user answers all the questions that have been asked, here are some symptoms that conclude that the user is exposed to pterygium disease based on the severity suffered:

Table 4: Pterigium Disease Decision Table

No.	Symptomps ID	Symptomps	Level 1	Level 2	Level 3
1.	G01	Red eyes	✓	✓	✓
2.	G02	Watery eyes	✓	✓	✓
3.	G03	Sensitive to light		✓	✓
4.	G04	Lots of exposure to ultraviolet light (like from the sun)	✓	✓	✓
5.	G05	Work outdoors (outdoor)	✓	✓	✓
6.	G06	Feels like something in it		✓	✓
7.	G07	Vision decreases			✓
8	G08	There is a thin membrane on the eye	✓		

In the table above the authors categorize pterigium disease into 3 parts based on the severity of the suffered, the higher the pterigium level itself the more severe the illness in suffering. Here is a system flowchart to determine the severity suffered by the user by using forward chaining method and calculation of certainty factor:

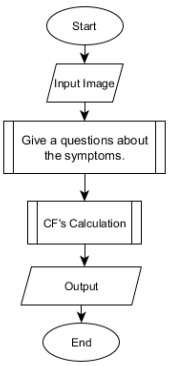


Figure. 9 Expert System Flowchart

According to the picture above, here is an explanation of expert system flowchart:

1. The first stage is the user enters the image to be selected on the gallery that has been provided.
2. Then the system will detect an image that has been entered, if the image is detected, the user must fill in some questions that have been submitted based on symptoms of symptoms of pterygium. Users only select Yes / No on apps as per what is felt about this disease. This process yields a conclusion about the patient's severe degree of pterygium disease.
3. Then if the user has filled in all the questions provided, the system will calculate based on the rules rules that have been made.
4. The end result included is a detectable picture of pterygium disease with the severity and value of certainty about the disease.

V. EXPERIMENTAL RESULT

There are several tests to be performed on this final project, such as accuracy testing to determine the effect of parameter settings on the performance of the detection system. Each image will be tested based on these parameters and the best performance results will be used to test the system as a whole. There will be some testing on the system to be done to find out what makes the app work best. The tests to be run are as follows:

a. Testing Detection System
b. Expert System Testing

Scenarios made in testing the detection system is to test the test data of 50 of 113 training data and data testing. The tests were performed on several value combinations of two different parameters for the detection system. These parameters are the factor values for enlarging the scanning window that is the scaleFactor and the minimum number of members a group of positive-affection detection windows is minNeighbor. The parameter values are used as follows:

- *scaleFactor* : { 1.1, 1.2, 1.3, 1.4 }
- *minNeighbor* : { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }

Table 5: Experiments on a scale of 1.1

minNeighbor	scaleFactor	Object Detected
1	**1.1**	0
2		6
3		11
4		16
5		24
6		5
7		6
8		11
9		16
10		**17**

Table 6: Experiments on a scale of 1.2

minNeighbor	scaleFactor	Object Detected
1	**1.2**	0
2		0

minNeighbor	scaleFactor	Object Detected
3	1.2	2
4		8
5		11
6		19
7		21
8		21
9		24
10		**27**

Table 7: Experiments on a scale of 1.3

minNeighbor	scaleFactor	Object Detected
1	1.3	0
2		2
3		6
4		9
5		16
6		22
7		22
8		22
9		25
10		**27**

Table 8: Experiments on a scale of 1.4

minNeighbor	scaleFactor	Object Detected
1	1.4	0
2		6
3		11
4		16
5		24
6		31
7		29
8		26
9		31
10		**38**

In table 8 above the author test on parameter 1.4. The results in the figure above indicate that in contrast to scaleFactor 1.4. In the scaleFactor 1.4 and minNeighbor 10 parameters the number of detected objects is 38 images of all the tested images as much as 50.

$$Accuracy = \frac{Amount\ of\ data}{Total\ data\ amount}\ X\ 100\%$$

$$= \frac{38}{50}\ X\ 100\%$$

$$= 76\%$$

From the test results to find the best values for minNeighbor and scaleFactor parameter with 4 times the experiment for 50 each image testing, then got the accuracy value 76% with minNeighbor 10 and scaleFactor 1.4. This parameter will be used further because according to the author this parameter becomes the most accurate parameter of the level detection done on all images that have been tested.

After obtaining a suitable value for the parameters used in the detection of pterygium disease in the picture, the authors make an expert system to increase the level of precision on the types of levels found in pterygium disease. There are 4 scenarios that have been made before, and the final result obtained as follows:

1. Scenario 1

In this scenario the patient does not choose any symptoms that are felt, so the system calculates and the final results obtained are as follows:

Table 9: The result of scenario 1

Condition	CF Point	Result
Pterygium disease was not detected	0	[ ] Accepted [ ] Not Accepted

From table 9 above if it can be concluded the system has taken the decision that the above input parameters produce condition “Pterygium disease was not detected”

2. Scenario 2

The following is the input of symptoms selected on the system:

a. Eyes red
b. Watery eyes
c. Lots of exposure to ultraviolet light (like from the sun)
d. Work outdoors

Table 10: The result of scenario 2

Condition	CF Point	Result
Level 1 Pterygium	0.49	[ ] Accepted [ ] Not Accepted

From table 10 above if using manual calculations found the value of CF for it can be concluded that Forward Chaining has taken the decision that the above input parameters produce conditions “Most likely the patient is exposed to level 1 pterygium”

3. Scenario 3

The following is the input of symptoms selected on the system:

a. Eyes red
b. Watery eyes
c. Sensitive to light
d. Feels like something in it

Table 11: The result of scenario 3

Condition	CF Point	Result
Level 2 Pterygium	0.39	[ ] Accepted [ ] Not Accepted

From table 11 above if using manual calculations found the value of CF for it can be concluded that Forward Chaining has taken the decision that the above input parameters produce conditions "Maybe the patient is exposed to level 2 pterygium"

4. Scenario 4

The following is the input of symptoms selected on the system:

a. Watery eyes
b. Work outdoors
c. Lots of exposure to ultraviolet light (like from the sun)
d. Visions decreases
e. Feels like something in it
f. There is a thin membrane on the eye

Table 12: The result of scenario 4

Condition	CF Point	Result
Level 3 Pterygium	0.47	[ ] Accepted [ ] Not Accepted

From table 11 above if using manual calculations found the value of CF for it can be concluded that Forward Chaining has taken the decision that the above input parameters produce conditions "Most likely the patient is exposed to level 1 pterygium 3"

VI. CONCLUSION

1. The accuracy of the detection system has been made as much as 76%, probably due to lack of training data used so that there are limitations of the system to detect pterigium disease that has various forms.
2. The accuracy of the detection system for the unhealthy image testing process (healthy eye) is 100% of the total of 30 images with the same parameters used as the testing process on the pterygium eye. This indicates that the correct system only detects pterigium membranes in the eye image.Visualization results show that the features used on each stage is different.
3. Testing on expert systems as much as 100% because it has been proven using manual calculations with the appropriate results.
4. Survey results obtained to 3 physicians at a level facility agreed that they needed this pterygium application.

REFERENCES

[1] S. West and A. Sommer, "Prevention of blindness and priorities for the future," vol. 79, no. 3, 2001.

[2] J. Chui, M. T. Coroneo, L. T. Tat, R. Crouch, D. Wakefield, and N. Di Girolamo, "Ophthalmic pterygium: A stem cell disorder with premalignant features," *Am. J. Pathol.*, vol. 178, no. 2, pp. 817–827, 2011.

[3] Errly, U. A. Mulyani, and D. Susilowati, "Distribusi Dan Karakteristik Pterigium Di Indonesia," *Bul. Penelit. Sist. Kesehat.*, vol. 14, no. 23, p. 85, 2011.

[4] M. Nasrun and C. Setianingsih, "Lie Detector With Eye Movement And Eye Blinks Analysis With Viola-Jones Algorithm And Fuzzy Logic Method."

[5] I. Khan, H. Abdullah, M. Shamian, and B. Zainal, "Efficient Eyes and Mouth Detection Algorithm Using Combination of Viola Jones and Skin Color Pixel Detection," *2013 Int. J. Eng.*, vol. 3, no. 4, pp. 51–60, 2013.

[6] D. Dan, R. Fitur, and M. Pada, "Citra Wajah Menggunakan Haar Cascade," pp. 298–305, 2015.

[7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. 2001 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition. CVPR 2001*, vol. 1, p. I-511-I-518, 2001.

[8] R. N. Perdana, B. Irawan, C. Setianingsih, F. Teknik, and U. Telkom, "Perancangan Pintu Pintar Untuk Deteksi Wajah Nyata Berbasis Pengolahan Citra Digital Menggunakan Deteksi Gerak Fisiologis Design Of Smartdoor For Live Face Detection Based On Image Processing Kata kunci : image processing , face detection , raspberry pi K," vol. 4, no. 3, pp. 4169–4176, 2017.

[9] M. Marimin, "Teori dan Aplikasi Sistem Pakar dalam Teknologi Manajerial," no. January 2009, 2007.

[10] A. Al-Ajlan, "The Comparison between Forward and Backward Chaining," *Int. J. Mach. Learn. Comput.*, vol. 5, no. 2, pp. 106–113, 2015.

[11] S. H. Stepahnie Halim, "Penerapan Metode Certainty Factor dalam Sistem Pakar Pendeteksi Resiko Osteoporosis dan Osteoarthritis," *Issn 2355-3286*, vol. VII, no. 2, pp. 59–69, 2015.

Jurnal Sistem Informasi (Journal of Information Systems). 2/13 (2017), 118-124
DOI: http://dx.doi.org/10.21609/jsi.v13i2.551

IMPLEMENTASI ALGORITMA *FP-GROWTH* DALAM PENENTUAN POLA HUBUNGAN KECELAKAAN LALU LINTAS

Rizky Fitria[1], Warnia Nengsih[2], dan Dini Hidayatul Qudsi[3]

[1,2,3] Program Studi Sistem Informasi, Politeknik Caltex Riau, Jl. Umban Sari (Patin) No. 1 Rumbai, Pekanbaru, Riau, 28265, Indonesia

Email: [1]rizky13si@mahasiswa.pcr.ac.id, [2]warnia@pcr.ac.id, [3]dinihq@pcr.ac.id

Abstrak

Tingginya jumlah kecelakaan yang terjadi menjadi alasan penting bagi pihak Satuan Lalu Lintas, salah satunya Polresta Pekanbaru untuk mengetahui hubungan antara factor-faktor penyebab kecelakaan. Hubungan tersebut dipelajari dengan menggunakan teknik *data mining*. Teknik ini bertujuan untuk menemukan informasi berupa pola yang dapat menjadi acuan dalam pengambilan kebijakan di Satuan Lalu Lintas Polresta Pekanbaru. Teknik *data mining* yang digunakan yaitu teknik *association rule* dengan Algoritma *Fp-growth*. Algoritma ini menerapkan struktur *data tree* untuk mengetahui pola kecelakaan lalu lintas. Pola tersebut ditentukan oleh dua parameter, yaitu *support* (nilai penunjang) dan *confidence* (nilai kepastian). Sistem dibuat berbasis *desktop* dengan bahasa pemograman Visual Basic.Net. Sistem ini menghasilkan pola kecelakaan yang sering terjadi. Berdasarkan pengujian *lift ratio,* didapatkan pola kecelakaan yang paling sering terjadi, yaitu faktor kecelakaan dengan jenis_luka adalah luka ringan, jenis_jalan adalah jalan arteri, waktu adalah padat kendaraan, dan jenis_kelamin adalah perempuan dengan nilai *lift ratio* 1.20%. Sedangkan hasil pengujian skala Likert didapatkan bahwa 88.09% pengguna merasa dimudahkan dengan sistem ini dan dapat membantu mereka dalam mengetahui pola kecelakaan yang sering terjadi.

Kata kunci: *kecelakaan lalu lintas, associattion rule, Fp-growth, visual basic.net*

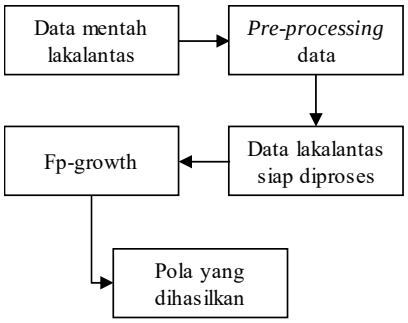
Abstract

The high number of accidents that happened is an important reason for the Traffic Unit, such as for Polresta Pekanbaru to determine the relationship between the factors causing the accident. The relationship is studied using data *mining* technique. This technique aims to find information in the form of patterns that can be a reference for Pekanbaru Police Traffic Unit in making policy. Data *mining* technique used in this research is association *rule* technique with *Fp-growth* Algorithm. This algorithm applies data *tree* structure to learn the traffic accident pattern. The pattern is determined by two parameters, namely support (the value of support) and confidence (the value of certainty). The system is developed as a desktop-based system using Visual Basic.Net programming language. This system generates *frequent* crash patterns. Based on the lift ratio test, the most *frequent* accident pattern, ie accident factor with *jenis_luka* is minor injury, *jenis_jalan* is arterial road, waktu is high density of vehicles, and *jenis_kelamin* is women with lift ratio value 1.20%. While Likert scale test results obtained that 88.09% of users feel facilitated with this system that it can help them in knowing the pattern of accidents that often occur.

Keywords: *traffic accident, association rule, Fp-growth, visual basic.net*

1. Pendahuluan

Kecelakaan lalu lintas (lakalantas) adalah suatu peristiwa yang tidak diduga dan tidak disengaja yang melibatkan kendaraan atau pengguna jalan lain yang mengakibatkan korban manusia dan kerugian harta benda [1]. Hal ini dapat kita lihat dalam kehidupan sehari-hari, terjadinya lakalantas mengakibatkan kerugian yang sangat besar dan bahkan dapat menghilangkan nyawa seseorang. Kejadian ini menjadi hal yang mengerikan yang terjadi di beberapa negara yang umumnya adalah negara-negara berkembang, dimana urusan transportasi masih dalam tahap pembangunan dan perbaikan. Pada saat ini, India merupakan negara dengan jumlah kematian terbanyak akibat kecelakaan lalu lintas, sedangkan Indonesia menempati urutan kelima (berdasarkan data WHO tahun 2011 [2]). Sementara, menurut Badan Intelijen Negara, kecelakaan lalu lintas di Indonesia menjadi pembunuh terbesar ketiga, dibawah penyakit jantung koroner dan TBC [3]. Bahkan, kecelakaan lalu lintas menjadi penyebab



Gambar 1. Diagram blok sistem secara garis besar

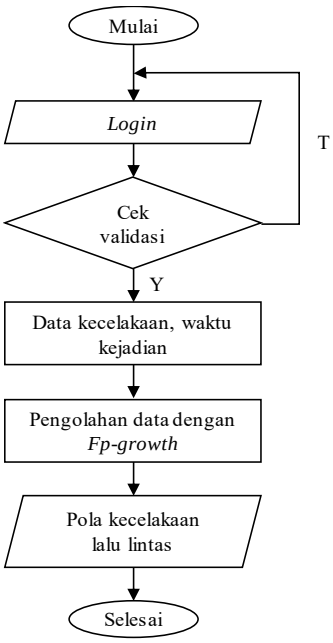
utama kematian anak-anak di dunia, dengan rentang usia 10-24 tahun. Terdapat sekitar 400.000 korban dengan usia dibawah 25 tahun yang meninggal di jalan raya, dengan rata rata angka kematian 1000 orang setiap harinya.

Untuk mendapatkan pola kecelakaan lalu lintas, diperlukan suatu teknik *data mining* yaitu teknik *association rule* untuk menemukan pola hubungan ketekaitan suatu *item*. Algoritma yang digunakan dari teknik *association rule* ini adalah adalah algoritma *Fp-growth. Fp-growth* adalah sebuah metode dalam *data mining* untuk mencari *frequent itemset* tanpa menggunakan *candidate generation. Fp-growth* ini merupakan pengembangan dari algoritma apriori. Kekurangan dari algoritma apriori diperbaiki dengan menghilangkan *candidate generation,* karena *Fp-growth* menggunakan konsep pembangunan *tree* dalam pencarian *frequent itemset.* Hal tersebut membuat algoritma *Fp-growth* lebih cepat dibandingkan algoritma apriori.

Pada penelitian ini, peneliti telah menganalisis hubungan kecelakaan lalu lintas di Kota Pekanbaru dengan variabel-variabel tertentu. Variabel yang akan diteliti antara lain yaitu waktu kejadian, jenis pekerjaan, usia, jenis kelamin, jenis kecelakaan dan jenis luka, jenis kendaraan dan jenis jalan. Penelitian ini menggunakan teknik *Association Rule* yaitu *Fp-growth* untuk mengetahui pola yang terbentuk dari faktor-faktor penyebab terjadinya kecelakaan. Penelitian ini diharapkan dapat membantu sistem yang ada untuk menghasilkan pengetahuan baru yang lebih bermanfaat bagi pihak Satuan Lalu Lintas Polresta Pekanbaru.

Penelitian Terdahulu

Penelitian yang dilakukan oleh [4] membahas tentang implementasi algoritma *Fp-growth* untuk menganalisis data transaksi yang terjadi disebuah toko kue yaitu L'Cheese pekanbaru. Pada pengimplementasiannya, peneliti menerapkan algoritma *Fp-growth* untuk menganalisis pola pembelian pelanggan dengan menganalisis penjualannya. Hasilnya



Gambar 2. Alur kerja sistem secara umum

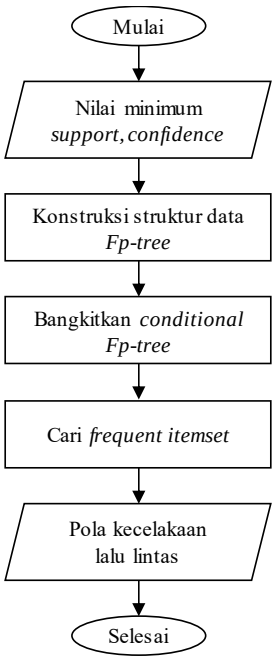
berupa *item* yang paling banyak terjual dan keterkaitan antara dua *item* atau lebih yang nantinya akan berdampak pada penjualan kue tersebut.

Penelitian yang dilakukan oleh [5] menganalisis terhadap penjualan sepeda motor untuk mengetahui sepeda motor mana yang lebih banyak dijual, dan berapa banyak persediaan yang diperlukan perusahaan untuk menyediakan sepeda motor. Untuk mengimplementasikan algoritma *Fp-growth* pada penelitian ini, dilihat dari banyaknya *item* yang terjual, lalu membuat rangkaian *tree* dengan *fp-tree* untuk mengetahui banyaknya *frequent itemset* yang terjadi.

Peneliti [6] melakukan penelitian untuk mengetahui pola hubungan yang sering terbentuk dari faktor-faktor penyebab terjadinya kecelakaan di jalan raya khususnya pada Kabupaten Sleman. Penelitian ini menggunakan metode *association rule* algoritma *apriori.* Faktor yang diambil dalam penelitian ini adalah waktu, profesi, usia, jenis kelamin, dan Surat Izin Mengemudi (SIM).

2. Method

Penelitian yang dibahas pada artikel ini merupakan penelitian untuk membangun sebuah aplikasi *desktop* yang akan menganalisis pola hubungan kecelakaan lalu lintas menggunakan metode *association rule* yaitu algoritma *Fp-growth.* Faktor yang diambil dalam penelitian ini adalah waktu kejadian, jenis pekerjaan, usia, jenis kelamin, jenis kecelakaan, jenis luka, jenis kendaraan dan jenis jalan.



Gambar 3. Cara kerja dari proses algoritma *fp-growth*

Teknik *Association Rule*

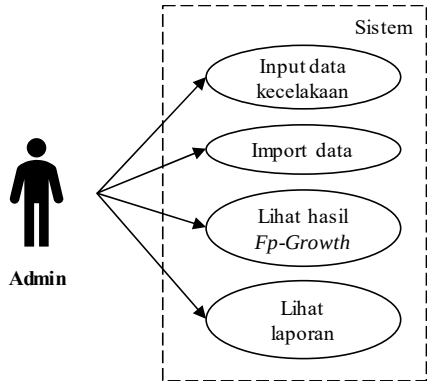
Analisis asosiasi adalah teknik *data mining* untuk menemukan aturan asosiatif antara kombinasi suatu *item*. Contoh aturan asosiatif dari analisis pembelian di suatu pasar swalayan adalah seperti berapa besar kemungkinan seorang pelanggan membeli roti bersamaan dengan susu [4]. *Association Rule* ditentukan oleh dua parameter, yaitu: 1) *Support* adalah suatu ukuran yang menunjukkan seberapa besar tingkat dominasi/persentase suatu kombinasi *item* dari keseluruhan transaksi, 2) *Confidence* adalah suatu ukuran yang menunjukkan kuatnya hubungan antar *item* dalam aturan asosiatif. Metodologi yang mendasari metode ini adalah analisis pola frekuensi tinggi dan pembentukan aturan asosiatif.

Analisis Pola Frekuensi Tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam *database*. Nilai *support* sebuah *item* diperoleh dengan persamaan(1). Sedangkan nilai *support* dari 2 *item* diperoleh dari persamaan(2).

$$Support(A) = \frac{\sum Transaksi\ Mengandung\ A}{\sum Transaksi} x100\% \quad (1)$$

$$Support(A,B) = P(A \cap B) = \frac{\sum Transaksi\ Mengandung\ A\ dan\ B}{\sum Transaksi} x100\% \quad (2)$$



Gambar 4. *Use case diagram* untuk pengguna "Admin"

Pembentukan Aturan Asosiatif

Setelah semua pola frekuensi tertinggi di*item*ukan, baru dicari aturan *asosiatif* yang memenuhi sarat minimum untuk *confidence* dengan menghitung *confidence* aturan *asosiatif* A > B nilai *confidence* dari aturan A > B diperoleh dari persamaan(3).

$$Confidence = P(\frac{B}{A}) = \frac{\sum Transaksi\ Mengandung\ A\ dan\ B}{\sum Transaksi\ Mengandung\ A} \quad (3)$$

Algoritma *Fp-growth*

Algoritma *Fp-growth* merupakan salah satu algoritma dari teknik *association rule* yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul *(frequent itemset)* dalam sebuah kumpulan data [7].

Pengembangan dari algoritma apriori ini terletak dalam *scanning database* dan akurasi *rules*-nya. *Fp-growth* lebih memberikan keuntungan karena hanya dilakukan satu atau dua kali saja *scanning database* sedangkan apriori perlu melakukan *scanning database* berulang ulang. Pada apriori akurasi *rules*-nya lebih tinggi daripada *Fp-growth.* Namun, karena *scanning* dilakukan secara berulang, kecepatannya jadi lebih lambat. Walaupun begitu, kedua algoritma ini memiliki tujuan yang sama yaitu menentukan *frequent itemset* [7].

Algoritma *Fp-growth* memiliki tiga tahapan utama [8], yaitu: 1) Tahap pembangkitan *Conditional Pattern Base*, merupakan sub-*database* yang berisi *prefix path* dan *suffix pattern* (pola akhiran); 2) Tahap pembangkitan *Conditional fp-tree*, dimana pada tahap ini *support count* dari setiap *item* untuk *conditional pattern base* dijumlahkan; 3) Tahap pencarian *Frequent Itemset*, merupakan lintasan tunggal (*single path*), kemudian didapatkan *frequent item*set dengan melakukan kombinasi *item* untuk *conditional fp-tree.*

TABEL 1.
HASIL UJI COBA NILAI SUPPORT

No.	Tahun	
	2013-2013	2013-2015
1	0.18	0.01
2	0.32	0.1
3	0.38	0.01
4	0.1	0.02
5	0.06	0.02
6	0.12	0.07
7	0.01	
8	0.06	
9	0.04	
10	0.01	
11	0.02	
12	0.07	
13	0.02	
14	0.16	
15	0.06	
16	0.15	
17	0.35	
18	0.54	
Rata-rata	0.15	0.04

TABEL 2.
HASIL UJI COBA NILAI CONFIDENCE

No.	Tahun	
	2013-2013	2013-2015
1	0.82	0.75
2	0.78	0.31
3	0.63	0.70
4	0.28	0.78
5	0.33	0.61
6	0.36	1
7	0.23	
8	0.32	
9	0.26	
10	0.80	
11	0.33	
12	0.55	
13	1	
14	0.88	
15	0.72	
16	0.55	
17	0.85	
18	0.83	
Rata-rata	0.58	0.69

Lift ratio

Lift ratio digunakan untuk mengukur seberapa penting *rule* yang telah terbentuk berdasarkan nilai *support* dan *confidence*. *Lift ratio* adalah perbandingan antara *confidence* dengan nilai *benchmark confidence*. *Benchmark confidence* adalah perbandingan antara jumlah semua *item consequent* terhadap total jumlah transaksi [3]. Rumus *benchmark confidence* dan *lift ratio* dapat dilihat pada persamaan(4) dan persamaan(5).

$$Benchmark\ Confidence = \frac{Nc}{N} \quad (4)$$

$$Lift\ ratio = \frac{Confidence(A,C)}{benchmark\ confidence(A,C)} \quad (5)$$

dimana Nc adalah jumlah transaksi dengan *item* dalam *consequent,* dan N adalah jumlah transaksi *database.* Apabila nilai *lift ratio* lebih besar dari 1, maka menunjukkan adanya manfaat dari aturan tersebut. Lebih tinggi nilai *lift ratio* maka lebih besar kekuatan asosiasinya.

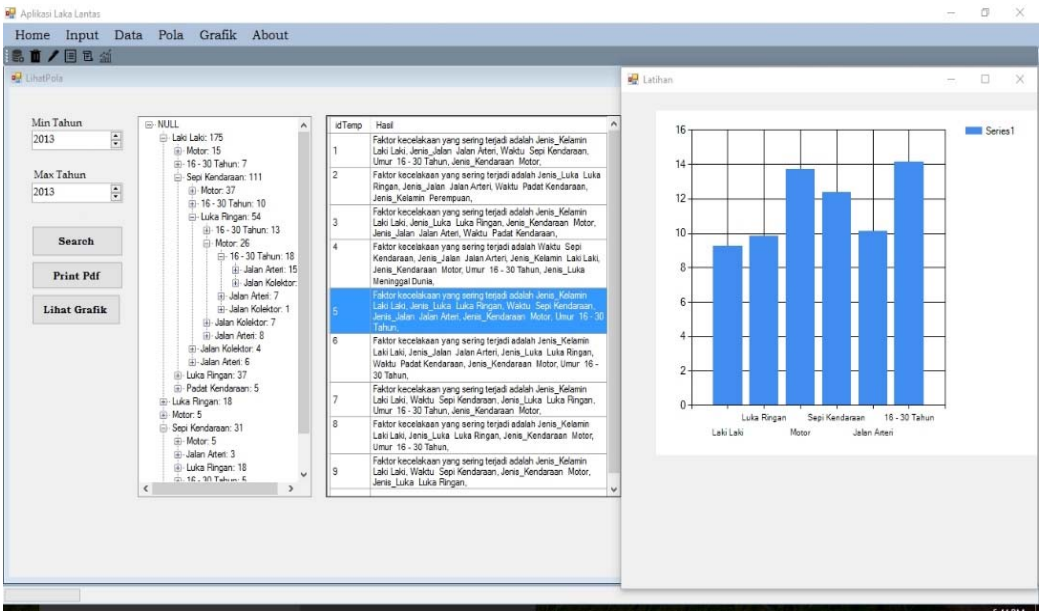
Blok Diagram Sistem

Gambar 1 merupakan diagram blok sistem secara garis besar. Data kecelakaan lalu lintas yang diperoleh dilakukan proses *pre-processing* terlebih dahulu sehingga menghasilkan data yang siap digunakan untuk proses *mining*. Data-data yang telah melewati proses tersebut akan di-*mining*-kan dengan algoritma *Fp-growth* untuk menghasilkan sebuah pola hubungan kecelakaan lalu lintas.

Flowchart

Pada Gambar 2 menjelaskan alur kerja sistem secara umum. Untuk dapat mengakses sistem, admin harus *login* terlebih dahulu kedalam sistem untuk dapat dicek kevalidannya. Setelah berhasil *login*, admin dapat meng-*import* data kecelakaan ke sistem. Sebelum proses pengolahan *Fp-growth*, admin harus memilih rentang waktu kejadian kecelakaan terlebih dahulu. Setelah itu, dilakukan proses *Fp-growth* pada data yang telah dipilih lalu sistem pun akan menampilkan pola yang dihasilkan.

Pada Gambar 3 menjelaskan cara kerja dari proses algoritma *Fp-growth.* Sebelumnya tentukan nilai minimum *support* dan nilai minimum *confidence*. Selanjutnya bangun *tree* dengan cara menentukan frekuensi kemunculan setiap *item* beserta prioritasnya untuk kemudian dipetakan kedalam *tree.* Setelah pembentukan *fp-tree* bangkitkan *conditional pattern base* dengan cara melakukan pembacaan pada *tree* dengan awalan *path* dari bawah ke atas. Kemudian dilakukan pembangkitan *conditional fp-tree* dimana pada tahap ini *support count* untuk tiap *item* pada setiap *conditional pattern base* akan dijumlahkan. *Item* yang memiliki jumlah *support count* lebih besar dengan *minimum support* akan dibangkitkan menjadi sebuah *tree* yang disebut *conditional fp-tree.* Tahapan selanjutnya yaitu pencarian *frequent itemset* dengan cara melakukan subsets dari *conditional fp-tree* terhadap *item* sehinga menghasilkan *frequent itemset.* Apabila nilai *confidence* yang dihasilkan pada subset yang terbentuk lebih besar dari nilai minimum *confidence* yang ditentukan maka *rule* tersebut dapat digunakan.



Gambar 5. Halaman untuk melihat pola

Use Case Diagram

Gambar 4 menunjukkan bahwa admin dapat melakukan semua fungsi yang ada pada sistem. Sebelum menjalankan semua fungsinya, admin harus *login* terlebih dahulu ke sistem.

3. Hasil dan Analisis

Tampilan Antarmuka

Gambar 5 merupakan tampilan dari sistem yang dibangun. Halaman untuk melihat pola tersebut akan menampilkan *tree* dan pola sesuai dengan tahun yang dipilih oleh pengguna berdasarkan algoritma *Fp-growth*.

Pengujian dan Analisis

Pengujian dilakukan untuk mengukur tingkat keakurasian dari metode *Fp-growth*. Pengujian ini menggunakan *lift ratio*, akan tetapi sebelumnya dilakukan pengujian dalam menentukan nilai minimum *support* dan *confidence* terbaik.

Pengujian Minimum Support dan Minimum Confidence

Pengujian nilai *minimum support* ini digunakan untuk mengetahui nilai *minimum support* yang tepat yang akan digunakan dalam sistem. Nilai *minimum support* itu sendiri merupakan syarat minimum jumlah kemunculan *item* secara bersamaan dalam keseluruhan transaksi. Pengujian ini dilakukan dengan menguji coba data dengan nilai *minimum support* terendah yaitu 0.01. Berikut hasil pengujian nilai *minimum support*.

Berdasarkan Tabel 1 hasil yang didapat melalui pengujian *minimum support* pada dataset 2013-2013 didapatkan 18 pola dengan nilai rata rata *support* yaitu 0.15 sedangkan pada dataset 2013-2015 yaitu 0.04. Sehingga peneliti mengambil nilai *support* terkecil yaitu 0.04 sebagai nilai *support* yang akan digunakan pada sistem penetuan pola kecelakaan ini. Pemilihan nilai *minimum support* yang paling kecil dikarenakan nilai *minimum support* adalah nilai acuan dalam perhitungan sebuah *itemset*. Hanya *itemset* yang nilai *support* nya sama atau lebih besar dari nilai *minimum support* saja yang akan dijadikan sebuah pola. Hal ini juga dikarenakan jika menggunakan nilai *minimum support* 0.15 maka pada *dataset* 2013-2015 tidak akan ada pola yang berhasil terseleksi dikarenakan tidak adanya pola yang memiliki nilai *support* 0.15 atau lebih, sehingga peneliti menggunakan nilai 0.4 sebagai nilai *minimum support* yang tepat.

Berdasarkan Tabel 2 hasil yang didapat melalui pengujian *minimum confidence* pada dataset 2013-2013 dihasilkan 18 pola dengan nilai rata rata *confidence* sebesar 0.58. Sedangkan pada dataset 2013-2015, dihasilkan 6 pola dengan nilai rata-rata yaitu 0.69. Berdasarkan hasil tersebut, peneliti mengambil nilai *confidence* terkecil yaitu 0.58 yang kemudian dibulatkan menjadi 0.6 sebagai nilai *confidence* yang akan digunakan pada sistem penentuan pola kecelakaan ini.

TABEL 3.
HASIL PENGUJIAN *LIFT RATIO*

No	Pola	Lift
1	Faktor kecelakaan dengan Jenis_Kelamin Laki Laki, Jenis_Jalan Jalan Arteri, Waktu Sepi Kendaraan, Umur 16 - 30 Tahun	1.05
2	Faktor kecelakaan dengan Jenis_Kelamin Laki Laki, Jenis_Jalan Jalan Arteri, Umur 16 - 30 Tahun, Jenis_Luka Luka Ringan, Tempat_Kejadian Jl Yos Sudarso	1.02
3	Faktor kecelakaan dengan Jenis_Kelamin Laki Laki, Jenis_Jalan Jalan Arteri, Umur 16 - 30 Tahun, Tempat_Kejadian Jl Yos Sudarso	1.06

Pengujian Lift ratio

Berdasarkan uji coba dataset yang telah dilakukan didapatkan nilai minimum *support* yaitu 0.04 dan minimum *confidence* yaitu 0.6. Pengujian selanjutnya yaitu *lift ratio* yang bertujuan untuk mengetahui kuat tidaknya aturan asosiasi yang telah terbentuk dengan persamaan rumus *lift ratio* pada persamaan(5). Hasilnya adalah analisis *lift ratio* untuk data 2013-2013 (1 tahun), dan 2013-2015 (3 tahun) yang direpresentasikan sebagai persamaan(6).

$$Lift\ (x \rightarrow Y) = \frac{Confidence\ (X \rightarrow Y)}{Supp\ (Y)} \qquad (6)$$

dimana, nilai Supp (Y) didapat dari persamaan(7).

$$Supp\ (Y) = \frac{jumlah\ Y}{jumlah\ data\ keseluruhan} \qquad (7)$$

Adapun nilai dari pengujian *lift ratio* dapat dilihat pada Tabel 3. Berdasarkan pengujian *lift ratio* dapat diambil analisis bahwa nilai *support* dan nilai *confidence* akan mempengaruhi besarnya nilai *lift ratio*. Semakin rendah nilai *support* dan semakin tinggi nilai *confidence* maka nilai *lift ratio* akan semakin besar. Begitu pula dengan semakin tinggi nilai *support* dan semakin rendah nilai *confidence* maka nilai *lift ratio* akan semakin kecil. Besar kecilnya nilai *lift ratio* menunjukkan kuatnya hubungan dalam aturan asosiasi tersebut. Hal ini ditunjukkan dengan jika nilai *lift ratio* lebih besar dari 1 berarti hubungan antar *item* pada pola yang dihasilkan semakin kuat.

Pola yang memiliki nilai *lift ratio* tertinggi pada dataset 2013-2013 adalah kecelakaan dengan jenis luka yaitu luka ringan pada jenis jalan arteri di waktu sepi kendaraan dengan korban berjenis kelamin perempuan yang memiliki nilai *lift ratio* 1.19. Kemudian pada dataset 2013-2015 nilai *lift ratio* tertinggi yaitu pada kecelakaan dengan jenis kelamin korban laki-laki pada jalan Yos Sudarso yang merupakan kategori jalan arteri dengan rentang umur korban 16-30 tahun dengan nilai *lift ratio* 1.06.

4. Kesimpulan

Berdasarkan uraian yang telah dijelaskan, maka dapat diambil kesimpulan bahwa aplikasi penentuan pola kecelakaan lalu lintas dengan algoritma *Fp-growth* yang dibangun dapat berjalan dengan baik dan sesuai dengan tujuan yang diharapkan. Meskipun demikian, aplikasi ini hanya membantu pihak terkait dalam membuat keputusan untuk meminimalisir terjadinya kecelakaan, bukan untuk membuat keputusan dalam membuat kebijakan.

Berdasarkan pengujian nilai *minimum support* dan pengujian nilai *minimum confidence* didapatkan nilai *minimum support* yang tepat yaitu 0.04 dan *nilai confidence* yang tepat adalah 0.6. Besar kecilnya nilai *minimum support* dan *confidence* ini nantinya akan mempengaruhi jumlah pola yang dihasilkan.

Berdasarkan pengujian *lift ratio* didapatkan kesimpulan bahwa semakin tinggi nilai *support* dan semakin tinggi nilai *confidence* maka nilai *lift ratio* yang dihasilkan akan semakin kecil. Besar kecilnya nilai *lift ratio* menunjukkan kuatnya hubungan dalam aturan asosiasi tersebut. Jika nilai *lift ratio* lebih besar dari 1 berarti pola yang dihasilkan semakin kuat hubungan antar *item*-nya.

Adapun saran yang dapat diberikan untuk penelitian selanjutnya yaitu dapat menambahkan atribut yang lain yang memiliki kategorinya jangan terlalu banyak seperti tempat kejadian kecelakaan serta menggunakan dataset yang jauh lebih besar dari sekarang, misalnya data yang diolah dari tahun 2011 sampai saat ini 2017. Dan juga penelitian selanjutnya dapat menerapkan algoritma asosiasi yang berbeda seperti *eclat*.

Referensi

[1] Ririanti. "Implementasi Algoritma FP-Growth pada Aplikasi Prediksi Persediaan Sepeda Motor (Studi Kasus PT. Pilar Deli Labumas)," *Pelita Informatika Budi Darma*, vol. 6, pp. 139-144, 2014.

[2] World Health Organization. (2017). Road traffic injuries. [online] Available at: http://www.who.int/violence_injury_prevention/road_traffic/en/ [Diakses pada 25 Okt. 2017].

[3] Bin.go.id. (2017). Kecelakaan Lalu Lintas Menjadi Pembunuh Terbesar Ketiga | Badan Intelijen Negara Republik Indonesia. [online] Available at: http://www.bin.go.id/awas/detil/197/4/21/03/2013/kecelakaan-lalu-lintas-pembunuh-terbesar-ketiga [Diakses pada 25 Okt. 2017].

[4] Fitriany Sitanggang, L., "Aplikasi Data mining Association Rule dengan Algoritma FP-Growth Untuk Mengenali Pola Pembelian

Pelanggan pada Toko Kue (Studi Kasus: L'Cheese Factory)," Politeknik Caltex Riau, 2014.

[5] Hakim, L., "Penentuan Pola Kecelakaan Lalu Lintas Menggunakan Metode *Association Rule* dengan Algoritma Apriori," *University Research Colloquium,* pp. 73-81, 2015.

[6] Todingrara, M., "Tinjauan Kriminologis terhadap Pelanggaran Lalu Lintas yang Menimbulkan Kecelakaan Berakibat Kematian (Studi Kasus Polres Tana Toraja Tahun 2009-2012)," Universitas Hasanuddin, 2013.

[7] Arifin, R. N., "Implementasi Agoritma *Frequent Pattern Growth* (FP-Growth) Menentukan Asosiasi antar Produk (Studi Kasus: Nadiamart)," *Jurnal Teknik ITS*, pp. 68-76, 2015.

[8] Fitriyani, "Implementasi Algoritma FP-Growth Menggunakan *Association Rule* pada *Market Base Analysis*," *Jurnal Informatika*, vol. 2, pp. 296-305, 2015.

IMPLEMENTASI ALGORITMA FP-GROWTH MENGGUNAKAN *ASSOCIATION RULE* PADA *MARKET BASKET ANALYSIS*

Fitriyani
Fakultas Teknik, Universitas BSI Bandung
Jalan Sekolah Internasional No. 1-6, Bandung 40282, Indonesia
fitriyani.fyn@bsi.ac.id

Abstract - *The set of data can be processed into information or useful knowledge, one of the data that can be processed is data purchases by consumers. However, large data processing will take a long time in the process. So that these data require appropriate methods in the process. The method is often used in data processing transactions are Apriori, but a great deal less precise data using Apriori because in the process repeatedly scanning the database (candidate set generation). In this study using the FP-Growth method for determining frequent itemset with structure of FP-Tree and Association Rule to determine support and confidence in the transaction data so that the results can be known relationships between an item with other items that are frequently purchased by consumers.*

Keywords : Apriori, FP-Growth, *Association Rule*, *Transaction*, *Frequent Itemset*.

Abstrak - Himpunan data yang besar dapat diolah menjadi informasi atau pengetahuan yang bermanfaat, salah satu data yang dapat diolah adalah data transaksi pembelian barang oleh konsumen. Akan tetapi pemrosesan data yang besar akan membutuhkan waktu yang lama dalam prosesnya. Sehingga data tersebut membutuhkan metode yang tepat dalam proses pengolahannya. Metode yang sering digunakan dalam pengolahan data transaksi adalah Apriori, akan tetapi data yang besar kurang tepat menggunakan Apriori karena dalam prosesnya melakukan *scanning* berulang kali pada database (*candidate set generation*). Dalam penelitian ini menggunakan metode FP-Growth untuk menentukan *frequent itemset* dengan struktur FP-Tree dan *Association Rule* untuk menentukan *support* dan *confidence* pada data transaksi sehingga hasilnya dapat diketahui hubungan-hubungan antara suatu barang dengan barang lainnya yang sering dibeli oleh konsumen.

Kata Kunci : Apriori, FP-Growth, *Association Rule*, *Transaction*, *Frequent Itemset*.

PENDAHULUAN

Database penjualan mempunyai data transaksi yang sangat besar jika setiap harinya ada 100 transaksi, dalam sebulan ada sekitar 3.000 transaksi dan setahun menjadi sekitar 36.000 transaksi, kemudian dalam waktu dua tahun data tersebut sudah menjadi 72.000. Data transaksi atau *database* penjualan yang sangat besar, apakah dimanfaatkan lagi atau hanya disimpan sebagai arsip? Jika hanya disimpan tanpa digunakan lagi data ini akan terus menumpuk sehingga untuk memelihara data tersebut membutuhkan biaya pemeliharaan dan ini akan masuk kedalam biaya tambahan perusahaan sehingga perusahaan harus menyediakan biaya untuk pemeliharaan data tersebut.

Himpunan data penjualan yang dimiliki sebenarnya dapat diolah menggunakan data mining untuk melihat pola pembelian pelanggan, dengan data mining untuk data yang besar tidak akan terbuang begitu saja dan dapat

bermanfaat sehingga dapat memberikan keuntungan kepada perusahaan. Data mining dasarnya adalah untuk membangun program komputer yang menyaring *database* otomatis, mencari keteraturan atau pola. Pola yang kuat dan jika ditemukan adanya kemungkinan-kemungkinan akan menggeneralisasi untuk membuat prediksi yang akurat pada data masa depan. Akan tetapi akan ada masalah ketika banyak pola yang tidak menarik (Witten, dkk: 2011). Data mining sebagai proses untuk mendapatkan informasi yang berguna dari gudang basis data yang besar dan dapat diartikan sebagai pengekstrakan informasi baru yang diambil dari bongkahan data besar yang membantu dalam pengambilan keputusan. Istilah data mining kadang disebut juga *knowledge discovery* (Prasetyo: 2012).Mesin pencari google memanfaatkan data mining dalam mengolah data permintaan informasi untuk para browser. Seperti contoh orang yang sedang sakit Flu ketika mencari informasi tentang penyakit ini, itu akan membedakan pencarian informasi pada orang yang hanya ingin mengetahui apa yang dimaksud dengan penyakit Flu. Dari sekian banyak permintaan informasi mengenai apapun, Google memanfaatkan data tersebut untuk pengetahuan yang bermanfaat, sehingga dapat memenuhi tantangan global saat ini.

Perusahan retail seperti supermarket mempunyai data yang begitu besar meliputi data penjualan, data penjualan yang berisi ratusan bahkan ribuan data transaksi dapat diolah dengan data mining sehingga data tersebut tidak hanya menjadi data arsip, tetapi dapat memberikan pengetahuan tentang pola pikir konsumen, seperti tipe-tipe pembeli, hubungan antar item yang dibeli. Seperti tipe pembeli ibu rumah tangga ketika belanja kebutuhan, sebagian besar mereka membeli bahan pokok. Tetapi berbeda untuk tipe pembeli pelajar, ketika masuk tahunn ajaran baru mereka akan membeli kebutuhan untuk belajar, dan sebagainya. Kesimpulannya adalah market basket merupakan pengolahan data keranjang belanja konsumen.

Contoh ibu rumah tangga sering membeli Gula kemudian Kopi, jarang sekali setelah membeli Gula akan membeli Kemeja dan Pulpen atau contoh lain yang paling umum di toko retail besar yang ada di US, Wal-Mart adalah pembelian Beer dan Diapers yang saling berhubungan, beer adalah minuman beralkohol dan diapers adalah merk popok bayi (Aprilla, dkk: 2013).

Dalam observasi diatas diketahui bahwa setiap hari jumat malam sekitar jam enam dan tujuh sore, laki-laki dewasa akan membeli beer dan popok secara bersamaan, para istri mereka akan meminta suaminya untuk membeli popok sedangkan para suami akan menghabiskan waktu hari minggu dengan meminum beer. Dari observasi tersebut memberikan keputusan pada pihak manajemen untuk display produk beer dan bersebelahan dengan produk diapers.Pola yang sudah terbentuk pada keranjang belanja atau market basket dapat memberikan keputusan untuk display produk, promosi diskon produk, strategi penjualan untuk barang yang kurang menarik minat beli konsumen dan sebagainya, pola ini disebut dengan analisis asosiasi (*analysis association*).

Data keranjang belanja dapat diubah menjadi biner untuk pengolahan data tersebut, dimana angka biner adalah berisi 1 (ya atau true) dan 0 (tidak atau false), pada data keranjang belanja untuk pembelian sebuah itemset X akan diberi angka 1, sedangkan tidak membeli itemset X akan diberi angka 0. *Association rule* adalah pernyataan implikasi $X \rightarrow Y$, dimana X dan Y merupakan itemset yang lepas (disjoint), sehingga memenuhi persyaratan $X \cap Y = \{\}$. Kekuatan aturan asosiasi dapat diukur dengan *support* dan *confidence*. *Support* digunakan untuk menentukan seberapa banyak aturan dapat diterapkan pada set data, sedangkan *confidence* digunakan untuk menentukan seberapa sering item di dalam Y muncul dalam transaksi yang berisi X (Prasetyo: 2012).

Rumus untuk mencari *support* dan *confidence* :

$$\text{Support } (X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \ldots.(1)$$

$$\text{Confidence } (X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} ..(2)$$

Contoh jika diketahui X = {Gula, Kopi} dan Y = {Aqua}, jumlah transaksi yang berisi X ∪ Y = {Gula, Kopi, Aqua} ada 1 dan total jumlah transaksi N ada 12 sehingga support untuk aturan X → Y menjadi (X → Y) = 1/12 = 0.083. Sedangkan untuk mencari aturan confidence adalah dengan membagi jumlah X ∪ Y = {Gula, Kopi, Aqua} dengan jumlah support untuk X = {Gula, Kopi}, $\sigma(X \cup Y)$ = 1 dan $\sigma(X)$ = 5 sehingga confidence (X → Y) = 1/5 = 0.2.

Hasil support dan confidence pada data mining untuk market basket dapat dibatasi dengan diterapkan association rule, dimana minsup dan minconf dapat diatur sesuai dengan kebutuhan. Minsup adalah ambang batas support, sedangkan minconf adalah ambang batas confidence. Sebagai contoh minsup 50% dan minconf 50% sehingga hasil dari pengolahan data yang akan tampil yang hanya mempunyai nilai support dan confidence 50% atau 0.5, data yang mempunyai nilai dibawah 50% atau 0.5 akan dibuang.

Algoritma dalam penelitian ini menggunakan algoritma FP-Growth yang merupakan salah satu algoritma untuk menentukan himpunan data yang sering muncul pada keranjang belanja konsumen atau dapat disebut dengan frequent itemset. Algoritma lainnya yang sering digunakan untuk menentukan frequent itemset adalah algoritma Apriori, paradigma Apriori yang dikembangkan oleh Agrawal dan Srikan (1994) adalah anti-monotone Apriori Heuristic dimana jika itemset pada panjang pola k *infrequent* (tidak sering muncul) dalam set data maka semua subset (bagian) dari itemset pasti juga *infrequent* sehingga kandidat yang diperiksa menjadi berkurang. Dasar algoritma Apriori adalah mencari himpunan kandidat dengan panjang (k+1) dari semua pola frequent dengan panjang k, kemudian kandidat yang memenuhi akan dicocokkan jumlah kemunculan pola tersebut dengan yang ada dalam database. Sehingga algoritma Apriori akan melakukan *scanning* database berulang-ulang, untuk data relatif kecil apriori dapat diterapkan untuk menentukan *frequent itemset*, akan tetapi untuk data yang besar algoritma Apriori akan membutuhkan waktu yang lama dalam menemukan pola (Samuel: 2008). Berbeda dengan algoritma FP-Growth yang hanya memerlukan dua kali scanning database dalam menghasilkan frequent itemset.

Algoritma FP-Growth adalah sebuah metode dalam data mining untuk mencari *frequnet itemset* tanpa menggunakan *candidate generation*. Pembangunan data menggunakan struktur FP-Tree untuk mengolah database transaksi (Jiawei, dkk: 2012).FP-Growth menggunakan strategi *divide-conquer* sehingga algoritma ini hanya membutuhkan dua kali *scanning database*. Contoh I={i_1, i_2, I_3, …, I_n} adalah sekumpulan dari data item dan basis data transaksi T={t_1, t_2, t_3, …, t_n}, setiap transaksi t_1 berisi bagian dari item I. *Scanning* pertama dilakukan ketika *support count* (jumlah transaksi yang mengandung itemset tertentu) sudah ditentukan diawal, contohnya t_1 memiliki item i_1, i_2 kemudian tentukan *support count*ξ = 2. Untuk menentukan support adalah berdasarkan kebutuhan karena *support* ini akan berpengaruh pada hasil FP-Tree yang dibuat dan hasil *support* dan *confidence* pada *association rule*. Semakin *support count*ξ bernilai tinggi 100% atau 10 maka akan sedikit node pada FP-Tree dan *support-confidence* akan sedikit menampilkan hasil pola hubungan antara item.

Kekurangan dalam algoritma Apriori diperbaiki oleh algoritma FP-Growth dengan menghilangkan *candidate generation*, karena algoritma FP-Growth menggunakan konseppembangunan tree dalam

pencarian frequent itemset. Hal tersebut yang membuat agoritma FP-Growth lebih cepat daripada algoritma Apriori dalam pencarian *frequent itemset*

Algoritma FP-Growth menggunakan struktur data tree yang disebut dengan FP-Tree. Dalam penggalian itemset yang frequent menggunakan tiga langkah sebagai berikut :

1. Tahap pembangkitan*Conditional Pattern Base*, tahap ini merupakan subdatabse yang berisi prefix path dan *suffix pattern* (pola akhiran).
2. Tahap pembangkitan Conditional FP-Tree, pada tahap ini *support count* dari setiap item untuk *conditional pattern base* dijumlahkan.
3. Tahap pencarian *Frequent Itemset*, tahap ini merupakan lintasan tunggal (*single path*), kemudian didapatkan *frequent itemset* (Samuel: 2008).

dengan melakukan kombinasi item untuk conditional FP-Tree.

METODE PENELITIAN

Penelitian ini menggunakan dataset transaksi atau market basket dengan jumlah 12 transaksi dengan berbagai macam item yang dibeli oleh konsumen. Dataset tersebut akan diolah untuk mencari frequent itemset menggunakan metode algoritma FP- Growth dengan struktur pembangunan FP-Tree, kemudian dicari pola-pola belanja konsumen dengan menggunakan *association rule* untuk mencari *support* dan *confidence* pada item yang saling berhubungan.

PEMBAHASAN

Pembahasan dalam penelitian ini adalah dengan menerapkan FP-Growth pada data transaksi dan menggunakan asssociation rule. Untuk minsup dan minconf diatur menjadi 50% atau 0.5 sehingga data support dan confidence yang tampil hanya yang memenuhi persyaratan tersebut.

FP-Tree

Dalam penelitian ini menggunakan sample data transaksi sebanyak 12 transaksi dengan berbagai macam item yang dibeli oleh konsumen. Adapun data tersebut sebagai berikut :

Tabel 1
Sampel Data Transaksi

TID	*Priority Item*
1	Gula, Kopi, Sampo, Sabun
2	Gula, Kopi, Sprei, Popok, Celana, Boneka, Kemeja
3	Sprei, Popok, Boneka
4	Gula, Aqua
5	Aqua, Sampo, Popok
6	Gula, Kopi, Sabun
7	Sprei, Celana, Boneka
8	Aqua, Sprei, Sampo, Popol, Sabun, Kemeja
9	Gula, Kopi, Celana
10	Aqua, Sampo
11	Gula, Kopi, Aqua
12	Sprei, Sampo, Sabun

Tentukan frekuensi kemunculan setiap item yang ada pada transaksi dan berikan prioritas untuk kemunculan item yang paling tinggi dengan rincian sebagai berikut :

Tabel 2
Frekuensi dan prioritas pada tiap item

Item	*Frequent*	*Priority*
Gula	6	1
Kopi	5	2
Aqua	5	3
Popok	4	6
Sprei	5	4
Sabun	4	7
Sampo	5	5
Kemeja	2	10
Celana	3	8
Boneka	3	9

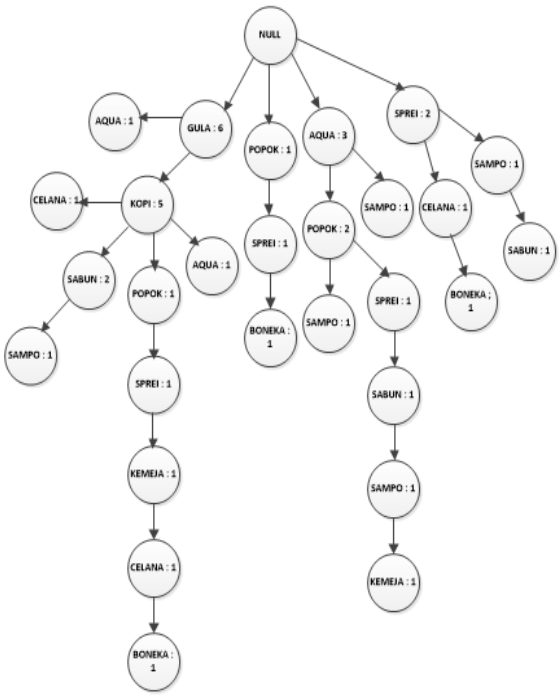
Setelah dilakukan *scanning* data untuk menentukan *frekuensi* dan prioritas didapat bahwa item produk Gula memiliki nilai *frequent* paling tinggi dengan urutan prioritas pertama, kemudian Kopi urutan kedua, Aqua urutan ketiga, Sprei urutan keempat, Sampo urutan kelima, Popok urutan keenam, Sabun urutan ketujuh, Celana urutan kedelapan, Boneka urutan kesembilan dan terakhir Kemeja urutan kesepuluh. Langkah selanjutnya adalah menyesuaikan urutan pada tiap transaksi sesuai dengan *frequent* paling tinggi. Seperti pada tabel dibawah ini :

Tabel 3
Item Yang Sudah Diurutkan
Berdasarkan Prioritas

TID	*Priority Item*
1	Gula, Kopi, Sampo, Sabun
2	Gula, Kopi, Sprei, Popok, Celana, Boneka, Kemeja
3	Sprei, Popok, Boneka
4	Gula, Aqua
5	Aqua, Sampo, Popok
6	Gula, Kopi, Sabun
7	Sprei, Celana, Boneka
8	Aqua, Sprei, Sampo, Popol, Sabun, Kemeja
9	Gula, Kopi, Celana
10	Aqua, Sampo
11	Gula, Kopi, Aqua
12	Sprei, Sampo, Sabun

Pada transaksi pertama item yang dibeli dengan urutan Gula, Kopi, Sabun, Sampo terlihat pada tabel 1, sedangkan setelah dilakukan scanning pertama dan diurutkan bedasarkan *frequent* tertinggi dapat diurutkan untuk transaksi pertama menjadi Gula, Kopi, Sampo, Sabun karena produk Sampo lebih tinggi *frequentnya* dibandingkan dengan produk Sabun. Terlihat pada tabel 3, untuk transaksi kedua dengan urutan Gula, Kopi, Popok, Sprei, Kemeja, Celana, Boneka menjadi Gula, Kopi, Sprei, Popok, Celana, Boneka, Kemeja karena produk Sprei lebih tinggi*frequentnya* dibandingkan dengan produk Popok, kemudian produk Celana lebih tinggi nilai *frequentnya* daripada produk Kemeja dan produk Boneka lebih tinggi nilai *frequentnya* daripada produk Celana serta produk Kemeja lebih tinggi nilai *frequentnya* daripada produk Boneka dan seterusnya.

Gambar dibawah ini memberikan ilustrasi mengenai pembentukan FP-Tree untuk data transaksi pada tabel 1.



Gambar 1
Ilustrasi Pembentukan FP-Tree

Ilustrasi pembentukan FP-Tree dapat dijelaskan sebagai berikut :

1. Pada gambar diatas adalah *scanning* kedua kemudian pembentukan FP-Tree setelah pengurutan item pada transaksi pertama, pembentukan FP-Tree dimulai dari node pertama diberi label Null selanjutnya node kedua diberi label Gula dan support count = 1 karena transaksi yang melaluilintasannya berjumlah 1. Node selanjutnya adalah Kopi dengan *support count* 1, Kopi berada setelah node Gula, kemudian

Sabun berada setelah node Kopi dan Sampo berada setelah node Sabun.

2. Setelah lintasan pertama selesai, digambarkan lintasan kedua dengan urutan Gula dan Kopi dengan *support count* = 2, kemudian setelah Kopi ada node baru dengan label Popok, Sprei, Kemeja, Celana, Boneka.
3. Lintasan kedua selesai, dilanjutkan dengan lintasan ketiga. Berbeda dengan lintasan pertama dan kedua, lintasan ketiga diawali dengan Popok dimana Popok merupakan node baru sehingga mempunyai urutan Null → Popok → Sprei → Boneka.
4. Selanjutnya lintasan keempat merupakan node yang sudah ada yaitu Gula sehingga *support count* dari Gula menjadi 3, kemudian didalam node Gula ada node Aqua dengan *support count* 1.
5. Proses selanjutanya adalah lintasan kelima yang diawali dengan node Aqua, Aqua setelah node Null dilanjutkan node Popok dan Sampo.
6. Lintasan selanjutnya kembali pada node Gula dengan *support count* 4, Kopi dengan *support count* 3 dan Sabun dengan *support count* 2.
7. Node Sprei, Celana dan Boneka berada pada node awal setelah node Null.
8. Lintasan kedelapan diawali dengan node Aqua, node Aqua sudah ada sebelumnya sehingga hanya menambahkan *support count* menjadi 2 dan berikutnya adalah node Popok dengan *support count* 2, kemudian selanjutnya node Sprei, Sabun, Sampo dan Kemeja setelah node Popok.
9. Node Gula bertambah *support count* sehingga menjadi 5 dan node Kopi menjadi 4 serta dalam node Kopi bertambah node Celana.
10. Lintasan kesepuluh node Aqua bertambah 1 untuk *support count* sehingga menjadi 3 dan mempunyai node turunan dengan nama Sampo.
11. Berikutnya node Gula dan Kopi bertambah lagi untuk *support count*, kemudian node Kopi mempunyai node turunan Aqua dengan *support count* 1.
12. Lintasan yang terakhir adalah node Sprei dengan *support count* menjadi 2, berikutnya node Sampo dan Sabun.

Penerapan FP-Growth

Setelah tahap pembentukan FP-Tree, selanjutnya adalah penerapan algoritma FP-Growth untuk mencari *frequent itemset* yang signifikan. Seperti pada penjelasan sebelumnya, penerapan algoritma FP-Growth mengurutkan tiga langkah seperti : *conditional pattern base*, *conditional FP-Tree* dan *frequent pattern generated*. Berikut tabel yang menunjukkan masing-masing dari langkah penerapan algoritma FP-Growth:

Tabel 4
Conditional Pattern Base

Conditional Pattern Base
Kemeja : {{GL, KP, SR, PP, CL, BN :1}, {AQ, SR, SP, PP, SB :1}}
Boneka : {{GL, KP, SR, PP, CL :1}, {SR, CL:1}, {SR,PP}}
Celana : {{GL, KP, SR, PP:1}, {GL, KP :1}, {SR:1}}
Sabun : {{GL, KP, SP :1}, {GL, KP :1}, {AQ, SR, SP, PP :1}, {SR, SP :1}}
Popok : {{GL, KP, SR :1}, {SR :1}, {AQ, SP :1}, {AQ, SR, SP :1}}
Sampo : {{GL, KP :1}, {AQ :2}, {AQ, SR :1}, {SR :1}}
Sprei : {{GL, KP :1}, {AQ :1}}
Aqua : {{GL :1}, {GL, KP :1}}
Kopi : {GL : 5}

Tabel 5
Conditional FP-Tree

Conditional FP Tree
Kemeja : (GL :1, KP : 1), (AQ : 1, SR :1)
Boneka : (GL :1, KP : 1), (SR :2)
Celana : (GL : 2, KP :2), (SR :1)
Sabun : (GL :2, KP :2), (SR : 1), (AQ :1)
Popok : (SR :1, AQ :2)
Sampo : (GL :1, KP :1),
Sprei : (GL : 1, KP :2), (SR :1)
Aqua : (GL :2, KP :2)
Kopi : (GL :1)

Tabel 6
Frequent Pattern Generated

Pattern Generated
Kemeja : {GL, KM : 1}, {KP, KM : 1}, {AQ, KM : 1}
Boneka : {GL, BN : 1}, {SR, BN : 1}
Celana : {GL, CL : 2}, {KP, CL : 2}
Sabun : {GL, SB : 2}, {KP, SB : 2}, {SR, SB : 1}, {AQ, SB : 1}
Popok : {GL, PP : 1}, {KP, PP : 1}, {SR, PP : 1}, {AQ, PP :1}
Sampo : {GL, SP : 1}, {KP, SP : 1}, {AQ, SP : 2}, {SR, SP : 1}
Sprei : {GL, SR : 1}, {KP, SR : 1}, {AQ, SR : 1}
Aqua : {GL, AQ : 2}, {KP, AQ : 2}
Kopi : {GL, KP : 5}

Penerapan*Association Rule*

Setelah pembahasan FP-Tree dan FP-Growth selanjutnya adalah penerapan *association rule* pada market basket.

Untuk mencari *association rule* adalah dengan mengelompokkan sejumlah item yang disebut dengan itemset. Itemset adalah kumpulan item di keranjang belanja dala jumlah kosong atau lebih (Prasetyo: 2012). Pada tabel dibawah ini adalah gambaran dari 1-itemset

Tabel 7
Itemset dengan 1-itemset

Frequent 2 Item	Gula	Kopi	Aqua	Popok	Sprei	Sabun	Sampo	Kemeja	Celana	Boneka
Gula	6	5	2	1	1	2	1	1	2	1
Kopi	5	5	1	1	1	2	1	1	2	1
Aqua	2	1	5	2	1	1	3	1	0	0
Popok	1	1	2	4	3	1	2	2	1	2
Sprei	1	1	1	3	5	2	2	2	2	3
Sabun	2	2	1	1	2	4	3	1	0	0
Sampo	1	1	3	2	2	3	5	1	0	0
Kemeja	1	1	1	2	2	1	1	2	1	1
Celana	2	2	0	1	2	0	0	1	3	2
Boneka	1	1	0	2	3	0	0	1	2	3

Untuk menghitung 2-itemset adalah dengan mencari produk yang terdiri dari 2 produk pada data transaksi, contoh yang pertama adalah produk {Gula, Gula} berjumlah 6 karena pembelian produk Gula ada 6 pada data transaksi, kemudian produk {Gula, Kopi} berjumlah 5 karena pada data transaksi ada 5 pembelian produk Gula dan Kopi, selanjutnya produk {Gula, Aqua} berjumlah 2 karena ada 2 transaksi pembelian produk tersebut dan seterusnya sampai semua data transaksi dihitung.

Tabel 8
Itemset dengan 2-itemset.

Customer	Gula	Kopi	Aqua	Popok	Sprei	Sabun	Sampo	Kemeja	Celana	Boneka
1	1	1	0	0	0	1	1	0	0	0
2	1	1	0	1	1	0	0	1	1	1
3	0	0	0	1	1	0	0	0	0	1
4	1	0	1	0	0	0	0	0	0	0
5	0	0	1	1	0	0	1	0	0	0
6	1	1	0	0	0	1	0	0	0	0
7	0	0	0	0	1	0	0	0	1	1
8	0	0	1	1	1	1	1	1	0	0
9	1	1	0	0	0	0	0	0	1	0
10	0	0	1	0	0	0	1	0	0	0
11	1	1	1	0	0	0	0	0	0	0
12	0	0	0	0	1	1	1	0	0	0

Data selanjutnya adalah meghitung *support* dari masing-masing item, pada tabel dibawah ini adalah perhitungan *support* untuk 2-itemset menggunakan rumus support (1) dan rumus *confidence* (2). Contoh produk {Gula, Kopi}, untuk mencari support adalah dengan menghitung transaksi yang pembeliannya berisi produk {Gula, Kopi}. Data tersebut ada sebanyak 5, kemudian dibagi jumlah transaksi 12 sehingga 5 dibagi 12 adalah 0.4167. Sedangkan menghitung confidence adalah dengan menghitung jumlah pembelian produk {Gula, Kopi} sebanyak 5 dibagi dengan jumlah pembelian Gula sebanyak 6 sehingga 5 dibagi 6 adalah 0.833 dan seterusnya sampai semua hubungan ada nilai *support* dan *confidence*.

Tabel 9
Support 2-itemset

Support (2 Items)	Gula	Kopi	Aqua	Popok	Sprei	Sabun	Sampo	Kemeja	Celana	Boneka
Gula	0,5	0,417	0,17	0,083	0,083	0,167	0,0833	0,0833	0,1667	0,08333
Kopi	0,42	0,417	0,08	0,083	0,083	0,167	0,0833	0,0833	0,1667	0,08333
Aqua	0,17	0,083	0,42	0,167	0,083	0,083	0,25	0,0833	0	0
Popok	0,08	0,083	0,17	0,333	0,25	0,083	0,1667	0,1667	0,0833	0,16667
Sprei	0,08	0,083	0,08	0,25	0,417	0,167	0,1667	0,1667	0,1667	0,25
Sabun	0,17	0,167	0,08	0,083	0,167	0,333	0,25	0,0833	0	0
Sampo	0,08	0,083	0,25	0,167	0,167	0,25	0,4167	0,0833	0	0
Kemeja	0,08	0,083	0,08	0,167	0,167	0,083	0,0833	0,1667	0,0833	0,08333
Celana	0,17	0,167	0	0,083	0,167	0	0	0,0833	0,25	0,16667
Boneka	0,08	0,083	0	0,167	0,25	0	0	0,0833	0,1667	0,25

Berikutnya adalah *confidence* untuk 2-itemset.

Tabel 10
Confidence 2-items

Confidence	Gula	Kopi	Aqua	Popok	Sprei	Sabun	Sampo	Kemeja	Celana	Boneka
Gula	1	0,833	0,33	0,167	0,167	0,333	0,1667	0,1667	0,3333	0,16667
Kopi	1	1	0,2	0,2	0,2	0,4	0,2	0,2	0,4	0,2
Aqua	0,4	0,2	1	0,4	0,2	0,2	0,6	0,2	0	0
Popok	0,25	0,25	0,5	1	0,75	0,25	0,5	0,5	0,25	0,5
Sprei	0,2	0,2	0,2	0,6	1	0,4	0,4	0,4	0,4	0,6
Sabun	0,5	0,5	0,25	0,25	0,5	1	0,75	0,25	0	0
Sampo	0,2	0,2	0,6	0,4	0,4	0,6	1	0,2	0	0
Kemeja	0,5	0,5	0,5	1	1	0,5	0,5	1	0,5	0,5
Celana	0,67	0,667	0	0,333	0,667	0	0	0,3333	1	0,66667
Boneka	0,33	0,333	0	0,667	1	0	0	0,3333	0,6667	1

PENUTUP

Kesimpulan

Penerapan algoritma FP-Growth dalam penelitian ini dapat diambil kesimpulan sebagai berikut :

1. Algoritma FP-Growth lebih efisien dalam memproses market basket atau data keranjang belanja konsumen.

2. Menggunakan *association rule* dapat mengetahui rule-rulebelanja konsumen seperti hubungan-hubungan tiap produk yang dibeli oleh konsumen dan mempengaruhi tipe-tipe konsumen dengan produk yang dibeli

3. *Association Rule* dapat memberikan keputusan pada manajemen untuk display produk, promosi, strategi pemasaran dan lain-lain

REFERENSI

Aprilla, D. C. (2013). *Data Mining dengan Rapid Miner*. Jakarta.

Jiawei han, M. K. (2012). *Data Mining Concepts and Techniques.* Waltham: Elsevier.

Prasetyo, E. (2012). *Data Mining Konsep dan Aplikasi Menggunakan MATLAB.* Yogyakarta: Penerbit ANDI Yogyakarta.

Samuel, D. (2008). Penerapan Struktur FP-Tree dan Algoritma FP-Growth dalam Optimasi Penentuan Frequent Itemset. 1-6.

Witten, I. H. (2011). *Data Mining Practical Machine Learning Tools and Techniques.* Burlington: Elsevier.

International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12 – No. 14, July 2018 – www.ijais.org

Efficient Mining of Frequent Itemsets using Improved FP-Growth Algorithm

Abdulkader M. Al-Badani
Faculty of Computer Science & Information Systems, Thamar University, Yemen

Basheer M. Al-Maqaleh
Faculty of Computer Science & Information Systems, Thamar University, Yemen

ABSTRACT
Frequent itemsets are itemsets that appear frequently in a dataset. Finding frequent itemsets plays an important role in association rules mining, correlations, and many other interesting relationships among data. Frequent itemset mining has been an active research area and a large number of algorithms have been developed. FP- Growth algorithm is currently one of the best approaches to frequent itemsets mining. It constructs a tree structure from transaction dataset and recursively traverse this tree to extract frequent itemsets in a depth first search manner. Also, it takes time to build an FP-tree, suffers from the increasing size of FP-tree and generating large number of frequent itemsets. In this paper, an improved frequent itemsets mining algorithm based on FP-Growth algorithm is proposed. The proposed algorithm uses a two dimensional array structure called Ordered Frequent Itemsets Matrix (OFIM) to construct a highly compact FP-tree. It greatly circumvents repeated scanning of datasets and it reduces the computational time, and reduces the number of frequent items that are generated obtaining significantly improved performance for FP-tree based algorithms.

General Terms
Data Mining, Association Rule, Frequent Itemsets Mining.

Keywords
FP-Growth Algorithm, Aprioiri Algorithm, FP-tree, Support Count, Ordered Frequent Itemset Matrix

1. INTRODUCTION
Data mining is the process of extracting useful information from huge amount of data stored in the databases [1]. Frequent itemset mining is one of the classical problems in the most of the data mining applications [2]. Frequent itemsets are common in real-life data, such as sets of items bought together in a super store. For example, a set of items, such as milk and coffee, which appear frequently together in a transaction dataset, is a frequent itemset . The frequent itemset can be defined as follows[1]:-

Let I={I1, I2, I3,...,In} be a collection of items. D is the set of transactions in the database, where each transaction T is a set of items, so I includes T. For any transaction A which is included in I, if and only if the transaction T concludes A, A can be called the item set. The support count of item set A is the number of the transaction which includes A in the database D. If the support count of itemset A is greater than or equal to a given support count, A can be called the frequent itemset, and the given support count is the minimum support count (*minsup*).

There are a large number of algorithms to extract frequent itemsets. Two main algorithms are Apriori [3], and FP-Growth [4]. Apriori is a typical algorithm for frequent itemset mining and association rule discovery over transactional databases. The frequent itemsets find out by Apriori can be used to find out association rules, which highlight common trends in the databases. It suffers from the large I/O cost caused by multiple database scans [1].

FP-Growth is the first successful tree-based algorithm of mining the frequent itemsets [4]. It works in a divide and conquers way that considerately reduces the size of the subsequent conditional FP-tree . It requires two scans of the datasets. The FP-tree is a compressed representation of the transactions. However, a compact representation does not reduce the potential combinatorial number of candidate itemsets, which is the bottleneck of FP-Growth [5]. Also, the large databases structure does not fit into main memory due to the very large size tree that might be generated [6]. So, in the proposed algorithm a new two-dimensional array structure, called Ordered Frequent Itemsets Matrix (OFIM), based on FP-Growth algorithm is used. This new structure compacts a transactional database to facilitate a suitable environment for efficient frequent itemsets mining.

The rest of paper is organized as follows: Section 2 presents related work. The original FP-Growth algorithm is presented in Section 3. The OFIM is elaborated in detail in Section 4. The proposed algorithm is presented in Section 5. The experimental results and discussions are described in Section 6. Conclusion and future work are given in Section 7.

2. RELATED WORK
This section presents some of the existing algorithms related to frequent itemsets mining. Many algorithms related to frequent itemsets mining are presented in [2,3,4,7].
An improved of FP-Growth algorithm for mining description-oriented rules is introduced in [8]. They have proposed new modification for description of gene groups using Gene Ontology (GO) based FP-Growth algorithm and the results show that the new algorithm allows generating rules faster. An association rule mining using new FP-Linked list algorithm is presented in [9]. It has proposed a new frequent pattern mining algorithm based on FP- Growth idea which is using a bit matric and a linked list structure to extract frequent patterns. A combined approach of frequent pattern growth and decision tree of Infrequent Weighted Itemset (IWI) mining are suggested in [10]. In their paper, two novel quality measures are proposed to test the IWI mining process. Efficient algorithms to find frequent itemsets using data mining are proposed in [11]. These algorithms are proposed to achieve privacy, utility and efficiency frequent itemsets mining, which is based on the frequent pattern growth algorithm. An improved algorithm of frequent itemsets mining is developed in [12]. It has proposed a non-recursive more efficient FPNR-growth algorithm which improves the time and space performance. A new hybrid frequent pattern-

Apriori (FP-AP) algorithm of high utility item set mining is developed in [13]. It has presented FP-AP algorithm, which is the combination of frequent pattern and Apriori algorithm. FP is proposed to split the longer transaction rather than truncate it and also to find the high profitable item with privacy to that item set without redundancy. A survey on FP-Growth tree using association rule mining is presented in [14]. The researchers introduced a new technique which extracts all the frequent itemsets without the generation of the conditional FP-trees. A new algorithm CT-PRO which uses the Compressed FP-tree is developed in [15]. The experimental results of this algorithm are much more efficient in terms of performance. An improved FP-tree algorithm with relationship technique for refined result of association rule mining is proposed in [16]. It has adapted the same idea for identifying frequent item set with large database. An efficient and updatable item-item frequency matrix for frequent itemset generation is introduced in [17]. This matrix maps the data between the items in the form of square matrix.

The most contribution of this work is a novel algorithm that uses OFIM structure to greatly compact structure of FP-tree, and then improve the performance of the algorithms operating on FP-trees.

3. FP-GROWTH ALGORITHM

FP-Growth algorithm [4] mines the complete set of frequent itemsets without generating the candidate. It retains the itemsets association by compressing the database representing frequent items into a frequent pattern tree, or FP-tree. It only needs two dataset scans when mining all frequent itemsets [2]. The first scan counts the number of occurrences of each item. The second scan constructs the initial FP-tree which contains all frequency information of the original dataset. Mining the dataset then becomes mining the FP-tree. The pseudo-code of the FP-Growth algorithm of a transaction database is given below [4,6].

Input: A transaction database DB, and a *minsup* threshold ξ.
Output: FP-tree.
Procedure:-

Step-1: Scan the transactional database and find support count for each item.

Step-2: If support(item) < *minsup*, discard the item.

Step-3: Construct a header table called I-list to store the sorted of frequent item-sets in descending order based on its support and node link.

Step-4: Initially, construct FP-Growth tree. In the first step, it creates the root of an FP-Growth tree and labels it as "null". And read the item in each transaction and create branch for each transaction. If each node has shared a common prefix, increment by 1 otherwise create a new node.

Step-5: In header table, each item points to its corresponding occurrences in the tree through a single link list, which is represented by dotted lines.

Step-6: Construct the mine FP-tree is which called FP-Growth tree.

The FP-tree has a header table associated with it. Single items and their counts are stored in the header table in decreasing order of their frequency[4]. Table 1 shows an example of a transactional dataset and Figure 1 shows the FP-tree, which generated by FP-Growth algorithm from this dataset.

Table 1: A dataset with nine transactions.

TID	List of items
T1	I1,I2,I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

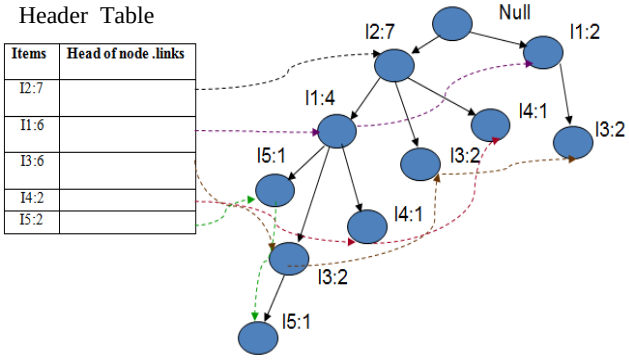


Figure 1: An Example of FP-tree(*minsup*=50%).

The generated frequent itemsets are shown in Table 2.

Table 2: The discovered frequent itemsets by FP-Growth algorithm

TID	Conditional FP-tree	Frequent itemsets
I5	<I2:2,I1:2>	{I2,I5:2}, {I1,I5:2}, {I2,I1,I5:2}
I4	<I2:2>	{I2,I4:2}
I3	<I2:4,I1:2>,<I1:2>	{I2,I3:4},{I1,I3:4}, {I2,I1,I3:2}
I1	<I2:4>	{I2,I1:4}

4. ORDERED FREQUENT ITEMSET MATRIX

OFIM is a two-dimensional array used to summarize the transactional database and contains all frequent itemsets, which they are sorted in support descending order. The OFIM is N×M, where N is the number of transactions and M is the longest number of ordered frequent items. The proposed algorithm scans the transactional dataset to drive a list of frequent items, in which items ordered in frequency descending order. This ordering is important since the construction of OFIM will follow this order. For any transaction, the candidate itemsets which have occurrence frequencies greater than or equal to the *minsup* threshold are added to the list of frequent itemsets, these lists are called Ordered Frequent Itemsets Lists (OFILs). Rest of the non-frequent candidate itemsets are discarded. The frequent items in each transaction are listed in this ordering in the rightmost column of Table 3.

Table 3: Transactional dataset with OFILs.

TID	List of items	OFILs.
T1	I1,I2,I5	I2,I1,I5
T2	I2,I4	I2,I4
T3	I2,I3	I2,I3
T4	I1,I2,I4	I2,I1,I4
T5	I1,I3	I1.I3
T6	I2,I3	I2,I3
T7	I1,I3	I1,I3
T8	I1,I2,I3,I5	I2,I1,I3,I5
T9	I1,I2,I3	I2,I1,I3

Notice that the frequent items in the transaction is ordered according to their order in the list of frequent items. Referring to the Table 3, the OFIL of transaction I1,I2,I5 is I2,I1,I5. An empty OFIM having N×M is initialized with "0" values. The matrix generation process reads the OFILs list by list. The process extracts items from each list. It then adds the items to the rows and to the corresponding columns of the matrix one by one. The process repeats for each list in the OFILs. Table 4 describes the complete OFIM after reading all the OFILs, which are given in Table 3.

Table 4. The OFIM.

T1	I2	I1	I5	0
T2	I2	I4	0	0
T3	I2	I3	0	0
T4	I2	I1	I4	0
T5	I1	I3	0	0
T6	I2	I3	0	0
T7	I1	I3	0	0
T8	I2	I1	I3	I5
T9	I2	I1	I3	0

5. THE PROPOSED ALGORITHM

Traditional FP-Growth algorithm takes time to construct FP-tree, discovers large number of frequent itemsets, and suffers from the increasing size of FP-tree, which may not fit in the main memory [4,5,6]. The process of discovering frequent itemsets takes the OFIM and a *minsup* threshold as input. The proposed algorithm scans every column in OFIM to compute the support of each different items, and the other (previous) columns are used to distinguish the node's parent node of current column. By this way, we can insert one level of nodes into FP-tree at a time, we cannot compute frequent items one by one in order to save the information of the current nodes and their parent nodes. Also, if any infrequent, say x item is found in any column then there is no any other item after x in the same row is frequent. So, the proposed algorithm deletes this row which greatly reduces the search space and then saves execution time. By using OFIM into the process of tree construction, the expensive frequent items scans in the proposed algorithm are saved. Also, more frequently occurring items are arranged at the top of the FP-tree and thus are more likely to be shared. This indicates that FP-tree structure is usually highly compact. This shortens the time of tree construction and decreasing the size of FP-tree, therefore the performance is much more better than the FP-Growth algorithm. The detailed descriptions of the proposed algorithm are as follows :-

Input : A transaction dataset and a *minsup* threshold.
Output: FP-tree.
1- Scan the transaction database once. Collect F, the set of frequent items F, and their supports. Sort F in support descending order as OFIL, the list of ordered frequent items. All infrequent itemsets are deleted in this step.

2- Create OFIM. For each row corresponds to the OFIL, all ordered frequent items in OFIL are entered item-by-item into the corresponding columns.

3- Create the root of an FP-tree, T, and label it as "null". Let column number in OFIM be j.

4- For (j=1; j<=M; j++)
{
If j=1 Then Do
{
Collect the set of frequent items and their supports, and sort items according to descending support count. Let the result be [f: n | OFIL] , where f is the first frequent item in OFIL, and n is the count. Insert these nodes as the root's child nodes into the FP-tree. Frequent items, f , are processed according to their order.
}
Else Do
{
Contrast both the current column (j) and the previous columns precede it, compare the set of frequent items and collect their supports. Let the result be [p, f: n | OFIL] where p is the parent frequent items of the previous columns, and f is the current frequent item of column (j). Link the nodes with the same item-name via the node-link structure. This means inserting [f: n] as the child nodes of p into the FP-tree and letting their node-link be linked to the nodes with the same item-name via the node-link structure.
}
}

The proposed algorithm mined the FP-tree as follows : start from each frequent length 1 pattern as an initial suffix pattern, construct its conditional pattern base which consists of the set of prefix path in the FP-tree co-occurring with the suffix pattern, then construct its conditional FP-tree and perform mining recursively on such a tree. The pattern growth is a achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree [4]. The FP-tree generated by the proposed algorithm of transactional dataset in Table 1 is shown in Figure 2.

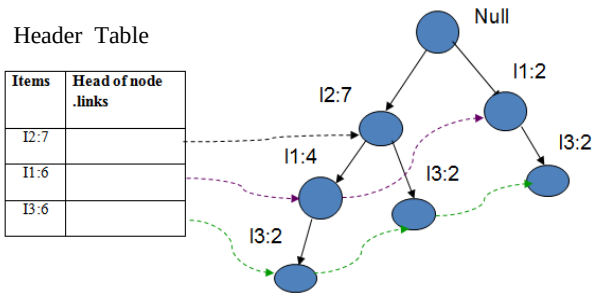


Figure 2: Construction of FP-tree using the proposed algorithm (*minsup*=50%)

The generated frequent itemsets are shown in Table 5.

Table 5: The generated frequent itemsets by the proposed algorithm

TID	Conditional FP-tree	Frequent itemsets
I3	<I2:4,I1:2>,<I1:2>	{I2,I3:4}, {I1,I3:4}, {I2,I1,I3:2}
I1	<I2:4>	{I2,I1:4}

Figures 1 and 2 show that the proposed algorithm constructs small size of FP-tree than FP-Growth algorithm. Also, Tables 1 and 2 indicate that the FP-Growth algorithm generates larger number of frequent itemsets than the proposed algorithm.

6. EXPERIMENTAL RESULTS AND DISCUSSIONS

The performance of the proposed algorithm is validated on real-world datasets obtained from UCI Machine Learning Repository which is a collection of widely used benchmark and real-world datasets for data mining and KDD community [18]. The performance of the proposed algorithm is evaluated and compared with well-known FP-Growth algorithm in terms of time required to find frequent itemsets and the number of discovered frequent itemsets from the specified datasets. All experiments are performed on a laptop with C++, 32 bit Windows 7, 4GB RAM and 2.2 GHZ Intel core (TM) Duo CPU. Table 6 shows some statistical information about the datasets used in this comparative study.

Table 6: Characteristics of the test datasets.

Datasets	Size	#Transactions
Grocery	0.56MB	10800
Mushroom	0.12MB	8124
Car	0.055MB	1728

The comparison performance of the proposed algorithm and FP-Growth algorithm on these three datasets are demonstrated below :-

6.1 Experiment One

This experiment was carried out on the Grocery dataset. It contains one month of real-world point-of- sale transactions data from a typical local grocery outlet. In order to efficiently evaluate the performance of the proposed algorithm over original FP-Growth algorithm, experiments have been conducted and compared several times with different values of *minsup*. The obtained results based on the execution time required to find the frequent itemsets and the number of frequent itemsets of various *minsup* values such as 10%, 20% , 30%, and 50% are shown in Table 7.

Table 7: Comparison results for the Grocery dataset with various *minsup* thresholds.

No.	*minsup*	Execution time per milliseconds (ms)		# Discovered Frequent itemsets	
		FP-Growth	Proposed algorithm	FP-Growth	Proposed algorithm
1	10%	800	600	13	12
2	20%	700	400	11	6
3	30%	600	300	6	3
4	50%	400	100	3	1

In general, when the values of *minsup* increase, the execution time and the number of discovered frequent itemsets decrease in both algorithms.

It is observed that the execution time of the proposed algorithm consumes less time compared to the FP-Growth algorithm, even though it varies the *minsup* threshold value. Figure 3 shows the performance of two algorithms according to the execution time for 4 different *minsup* thresholds. It clearly illustrates that the proposed algorithm outperforms than FP-Growth algorithm. The reason behind that the FP-Growth needs to construct a large number of conditional sub-trees and then generates a large number of frequent itemsets, it is not only time consuming but also high memory cost.

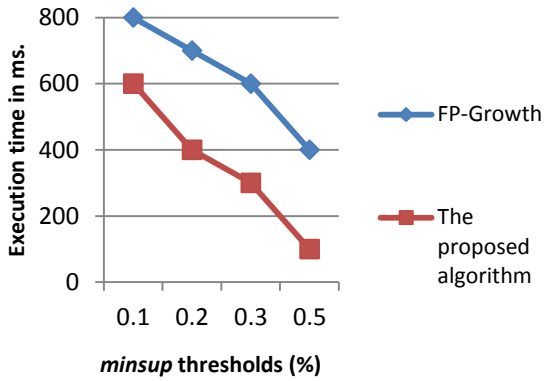


Figure 3: Comparing the results of the execution time and the *minsup* thresholds for the Grocery dataset.

6.2 Experiment Two

Mushroom dataset was used for this experiment. This dataset has 8124 records, with 23 attributes that contain 119 item. Table 8 shows the execution time and the number of discovered frequent itemsets of the FP-Growth and the proposed algorithm of various *minsup* thresholds such as 10% , 20% , 30%, and 50%.

Table 8: Comparison results for the Mushroom dataset with various *minsup* thresholds.

No.	*minsup*	Execution time per milliseconds (ms)		# Discovered Frequent itemsets	
		FP-Growth	Proposed algorithm	FP-Growth	Proposed algorithm
1	10%	590	300	20	19
2	20%	560	260	18	7
3	30%	550	250	17	5
4	50%	530	240	13	3

Table 8 indicates that the execution time of the proposed algorithm are approximately equal even though it varies the *minsup* threshold of the Mushroom dataset and it gives less mining time and a small number of the generated frequent itemsets than the FP-Growth algorithm. The proposed algorithm scales much better than the FP-Growth. This is because as the *minsup* threshold goes down, the number of the frequent itemsets increase dramatically. The candidate sets that the FP-Growth must handle become large, and the pattern matching with a lot of candidates by searching through the FP-tree becomes very expensive. Figure 4 shows the comparison results between the two algorithms according to the execution time with 4 different *minsup* thresholds.

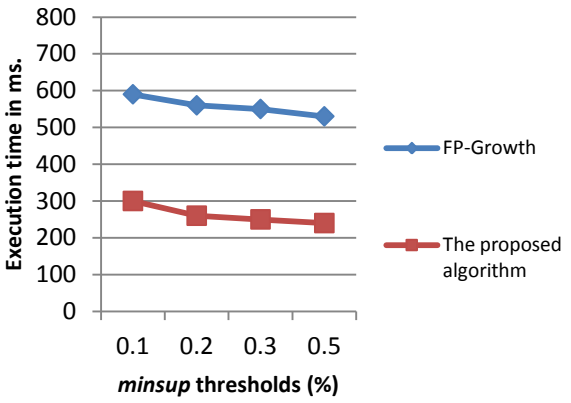


Figure 4: Comparing the results of the execution time and the *minsup* thresholds for the Mushroom dataset

6.3 Experiment Three

This experiments was carried out on the Car dataset. This dataset contains 1728 instances and 7 attributes. Table 9 shows the execution time and the number of discovered frequent itemsets of the FP-Growth and the proposed algorithm of the various *minsup* thresholds such as 10%, 20% , 30%, and 50%.

Table 9: Comparison results for the Car dataset with various *minsup* thresholds.

No.	*Minsup*	Execution time per milliseconds (ms)		# Discovered Frequent itemsets	
		FP-Growth	Proposed algorithm	FP-Growth	Proposed algorithm
1	10%	900	500	11	8
2	20%	600	400	10	6
3	30%	300	200	9	3
4	50%	200	100	6	2

Table 9 indicates that the execution time and the number of discovered frequent itemsets of the proposed algorithm are less than FP-Growth algorithm. The comparison results between the execution time and the *minsup* thresholds of the Car dataset for both algorithms are shown in Figure 5.

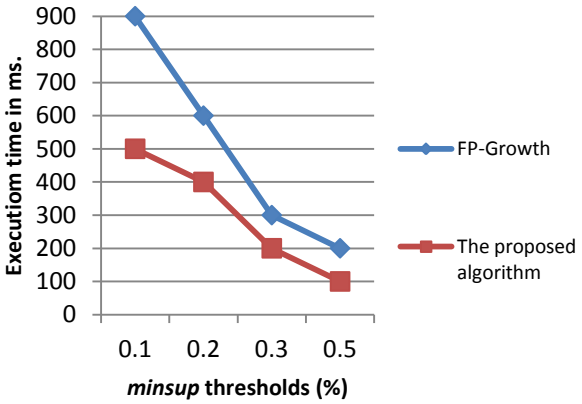


Figure 5: Comparing the results of the execution time and the *minsup* thresholds for the Car dataset

This figure indicates as the *minsup* threshold goes up, the execution time and the number of generated frequent itemsets in both algorithms decrease dramatically.

7. CONCLUSION AND FUTURE WORK

Efficient algorithms of mining frequent itemsets are crucial for mining association rules. In this paper, an improved FP-Growth algorithm for efficient mining of the frequent itemsets is proposed. The proposed algorithm uses OFILs to construct the OFIM. Consequently, the proposed algorithm uses OFIM to construct a highly compact FP-tree, and then saves the costly dataset scans in the subsequent mining processes which reduces the time of FP-tree construction. It applies a pattern growth method which avoids costly candidate generation. OFIM saves traversal time for all items and the next level of the FP-tree can be initialized directly. The proposed algorithm deletes infrequent items appropriately, thereby the subsequent processes are completing their respective tasks more efficiently without unnecessarily wasting their efforts in processing the irrelevant data. The performance gain achieved by the proposed algorithm is due in most part to the highly compact structure of FP-tree, which stores only the frequent items in a frequency-descending order using OFIM. The experimental results show that the proposed algorithm is superior to the original FP-Growth algorithm in mining time and the number of discovered frequent itemsets.

Future research can improve the proposed algorithm to incrementally update an FP-tree, such as adding daily new transactions into a database containing records accumulated for months.

8. REFERENCES

[1] Han, J., Pei, J., and Kamber, M. 2011. Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco, California, USA.

[2] Shridhar, M., and Parmar, M. 2017. Survey on association rule mining and its approaches. International Journal of Computer Sciences and Engineering (IJCSE), 5(3), pp.129-135.

[3] Agrawal, R., and Srikant, R. 1994. Fast algorithms for mining association rules. In Proceeding of 20th International Conference on Very Large Databases (VLDB), pp. 487-499.

[4] Han, J., Pei, J., and Yin, Y. 2000. Mining frequent patterns without candidate generation. ACM. pp, 1-12.

[5] Wei, F., and Xiang, L. 2015. Improved frequent pattern mining algorithm based on FP-Tree. In Proceedings of The Fourth International Conference on Information Science and Cloud Computing (ISCC2015), pp.18-19.

[6] Krupali, R., Garg, D., and Kotecha, K. 2017. An improved approach of FP-Growth tree for frequent itemset mining using partition projection and parallel projection techniques. International Recent and Innovation Trends in Computing and Communication, 5(5), pp. 929-934.

[7] Khanali, H., and Vaziri, B. (2017). A survey on improved algorithms for mining association rules. International Journal of Computer Applications(IJCA), 165(9), pp. 6-11.

[8] Gruca, A. 2014. Improvement of FP-Growth algorithm for mining description-oriented rules. In Man-Machine Interactions, Part of Advances in Intelligent Systems and Computing, (AISC), Springer, vol. 242, pp. 183-192.

[9] Sohrabi, M. K., and Marzooni, H. H. 2016. Association rule mining using new FP-Linked list algorithm. Journal of Advances in Computer Research (JACR), 7(1), pp. 23-34.

[10] Dange, A. S., and Patil, S. J. 2016. A combined approach of frequent pattern growth and decision tree for infrequent weighted itemset mining. International Research Journal of Engineering and Technology (IRJET), 3(7), pp. 2070- 2075.

[11] Sagar, B. P., and Kale, S. 2017. Efficient algorithms to find frequent itemsets using data mining. International Research Journal of Engineering and Technology (IRJET), 4(6), pp. 2645- 2648.

[12] Hao, J., and Xu, H. 2017. An improved algorithm for frequent itemsets mining. In 5th International Conference on Advanced Cloud and Big Data (CBD), IEEE Computer Society , pp. 314-317.

[13] Devi, R. S., and Shanthi, D. 2016. A new hybrid frequent Pattern-Apriori (FP-AP) algorithm for high utility item set mining. Middle East Journal of Scientific Research (MEJSR), 24(3), pp. 986-991.

[14] Princy. S, Ankita, H., Babita, P., and Shiv, K. 2017. A survey on FP (Growth) tree using association rule mining. International Research Journal of Engineering and Technology(IRJET), vol. 4, Issue 7, pp. 1637-1640.

[15] Jiten, G., Ashish, P., Swapnit, M., and Christi, L. 2017. Compressed frequent pattern tree. International Journal of Engineering Sciences and Research Technology (IJESRT), 6(4), pp. 652-657.

[16] Saxena, P. and Jain, R. 2016. An improved FP-Tree algorithm with relationship technique for refined result of association rule mining. International Journal of Scientific Research in Science, Engineering and Technology(IJSRSET), vol. 2, pp. 525-529.

[17] Usman, A., Zhang, P., and Theel, O. 2017. An efficient and updatable item-to-item frequency matrix for frequent itemset generation. ICC'17, Cambridge, United Kingdom, ACM, pp. 978 -983.

[18] Blake, C. L., and Merz., M. J, UCI Repository of Machine Learning Databases [http://www. ics. uci. edu/~ mlearn/ MLRepository. html]. Irvine, CA: University of Californial, Department of Information and Computer Science.