Terbit *online* pada laman web jurnal :
http://ejournal.amikompurwokerto.ac.id/index.php/telematika/

Telematika

Terakreditasi Sinta "3" KEMENRISTEKDIKTI, No. 21/E/KPT/2018

UNIVERSITAS AMIKOM PURWOKERTO

Penerapan *Naïve Bayes* pada Pendeteksian *Malware* dengan Diskritisasi Variabel

Inda Anggraini[1], Yesi Novaria Kunang[2], dan Firdaus[3]

[1,2,] Magister Teknik Informatika, Universitas Bina Darma
[3] Magister Teknik Sipil, Universitas Bina Darma
Email : indaanggraini@gmail.com[1], yesinovariakunang@binadarma.ac.id[2], firdaus.dr@binadarma.ac.id[3]

INFO ARTIKEL

Sejarah Artikel:
Menerima 14 Agustus 2019
Revisi 7 Oktober 2019
Diterima 20 Februari 2020
Online 28 Februari 2020

Keywords:
Malware Detection
Naïve Bayes
Discretization
Data Mining

ABSTRACT

Malicious software (malware) is rogue software specifically designed to carry out malicious or destructive software activities on computers such as viruses, Trojans, and others that are spread through the internet network. The number of activities that spread malware that occurs through the internet network makes many users uneasy one form of the attack is to insert malicious or malicious files into the computer. For example, such as the web shell scripting script that is inserted into the internet service provider computer. This study aims to analyze malware attacks using the Naïve Bayes Classifier Algorithm with the discretization of 3-interval and 5-interval Min-Max variables for continuous attributes. Discretization (discretion) attribute is a technique for changing a function or continuous value into a discrete form. This technique is done as an adjustment to the possibility of the emergence of continuous values in a very small dataset feature. Discretization of variables is done in a dataset of type continuous so that the probability value indicates the possibility of the same value coming out of a class. Using the Naive Bayes algorithm is expected to help facilitate users in finding the right method for detecting attacks from malware. The experimental results show that the application of Naïve Bayes in the classification of data that has not gone through the discretion stage produces an accuracy of 69.72% with the prediction of malware 63.53 % while the data that has passed the discretization stage can provide accuracy of up to 79.97 % with 81.29 % malware prediction. The use of the Naïve Bayes by the binning method in this study has an increased detection ability compared to the classification process without using the binning process (discretization). The discretion process can make the Naïve Bayes algorithm more accurate in detecting malware.

Kata Kunci:
Pendeteksian *Malware*
Naïve Bayes
Diskritisasi
Data Mining

Korespondensi:
Telepon: +62 81379517789
Email:
indaanggraini@gmail.com

ABSTRAK

Malicious software (malware) adalah software jahat yang dirancang khusus untuk melakukan aktifitas berbahaya atau perusak perangkat lunak pada komputer seperti virus, Trojan, dan lain-lain yang disebar melalui jaringan internet. Banyaknya aktifitas penyebaran malware yang terjadi melalui jaringan internet membuat banyak pengguna menjadi resah salah satu bentuk dari serangan tersebut yaitu dengan melakukan penyisipan file-file berbahaya atau malicious ke komputer. Contohnya seperti penyisispan skrip web shell yang di sisipkan ke komputer penyedia layanan internet. Penelitian ini bertujuan untuk melakukan analisa terhadap serangan malware dengan menggunakan Algoritme *Naïve Bayes* Clasiffier dengan diskritisasi variabel Min-Max diskritisasi 3-interval dan 5-interval untuk atribut kontinu. Discretization (pendiskritan) atribut merupakan teknik untuk merubah sebuah fungsi atau nilai kontinu kedalam bentuk diskrit. Teknik ini dilakukan sebagai penyesuaian terhadap kemungkinan kemunculan nilai kontinu dalam fitur dataset yang sangat kecil. Pendiskritisasian variabel dilakukan pada dataset yang bertipe kontinu, sehingga nilai probabilitas menunjukan kemungkinan nilai yang sama keluar pada suatu kelas. Dengan menggunakan Algoritme naive bayes ini diharapkan dapat membantu mempermudah pengguna dalam menemukan metode yang tepat untuk

mendeteksi serangan dari malware. Hasil percobaan menunjukan bahwa penerapan *Naïve Bayes* pada klasifikasi data yang belum melalui tahap pendiskritan menghasilkan tingkat akurasi sebesar 69.72 % dengan prediksi malware 63.53 % sedangkan pada data yang telah melewati tahap diskritisasi mampu memberikan akurasi hingga 79.97 % dengan prediksi malware 81.29 %. Penggunaan metode *Naïve Bayes* dalam penelitian ini memiliki kemampuan deteksi yang meningkat dibandingkan dengan proses klasifikasi tanpa menggunakan proses binning (diskritisasi). Proses pendiskritan dapat menjadikan Algoritme *Naïve Bayes* menjadi lebih akurat di dalam mendeteksi malware.

PENDAHULUAN

Perkembangan teknologi yang semakin pesat terutama teknologi komputer khususnya di bidang jaringan selain memberikan kemudahan, juga memberikan masalah di sisi keamanan dari komputer yang terintegrasi. Permasalahan keamanan komputer yang paling banyak dijumpai adalah penyebaran *malware* (*malicious software*) melalui jaringan *internet* yang menyebabkan berbagai macam kerugian (Setiawan dkk., 2017).

Malware sendiri merupakan perangkat lunak yang secara khusus dirancang untuk melakukan aktifitas berbahaya yang bisa merusak perangkat lunak lain. Contoh *malware* seperti Virus, *Trojan, Spyware* dan *Exploit* yang dibuat khusus agar tersembunyi sehingga mereka bisa tetap berada di dalam sistem komputer pada periode waktu tertentu tanpa sepengetahuan pemilik sistem (Cahyanto dkk., 2017). Beberapa *Malware* diciptakan dengan tujuan memata-matai seseorang, melakukan aktifitas merugikan seperti pencurian data dan informasi pribadi, membobol keamanan program dan sistem operasi serta banyak lagi. Untuk membobol suatu perangkat lunak atau sistem operasi dilakukan dengan menggunakan *script* yang diselipkan secara tersembunyi oleh penyerang (Sandag dkk., 2018).

Dengan banyaknya aktifitas penyebaran *malware* yang terjadi melalui jaringan *internet* membuat banyak pengguna menjadi resah. Untuk itu perlu melakukan pendeteksian terhadap serangan *malware* khususnya di jaringan agar pengguna bisa mengetahui apakah data yang berasal dari *internet* aman dari penyisipan malware atau tidak (Akbi & Rosyadi, 2018). Beberapa peneliti menggunakan pendekatan pembelajaran mesin seperti *k-nearest neighbor* (kNN) (Sandag dkk., 2018), Support Vector Machine (SVM) (Herlambang & Basuki, 2019) dan juga metoda klastering (Akbi & Rosyadi, 2018). Penelitian-penelitian tersebut sebagian besar masih belum mencapai nilai akurasi yang maksimal (Herlambang & Basuki, 2019).

Penelitian lain dilakukan oleh (Wirawan & Eksistyanto, 2015) menggunakan pendekatan *Naïve Bayes* pada sistem pendeteksi serangan atau *Intrusion Detection System* dengan menggunakan teknik diskritisasi Variabel. Hasil penelitian ini memperlihatkan penggunaan teknik binning mampu meningkatkan pendeteksian secara signifikan jika dibandingkan dengan proses klasifikasi tanpa menggunakan teknik binning. Dengan Teknik *binning* (pendiskritisasian) menjadikan probabilitas dari Algoritme *Naïve Bayes* bisa lebih dihandalkan dalam penentuan kelas.

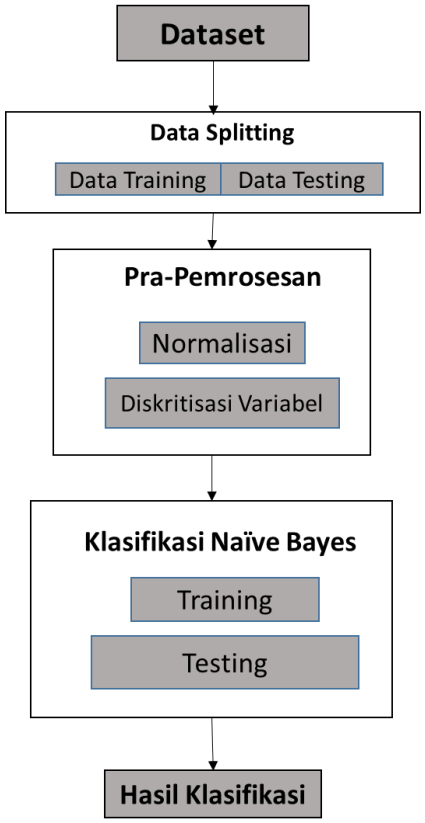
Algoritme *Naïve Bayes* merupakan Algoritme data *mining* yang relatif sederhana yang memiliki kelebihan tingkat akurasi yang tinggi dan dapat menangani data dalam jumlah besar (Huaturuk et al., 2018). Untuk itu dalam penelitian ini akan menerapkan metode *Naïve Bayes Classifier* dalam mendeteksi *malware* dengan teknik diskritisasi variabel. Pendiskritan ini dilakukan sebagai penyesuaian terhadap kemungkinan kemunculan nilai kontinu di dalam fitur dataset yang akan mempengaruhi hasil proses

klasifikasi. Untuk mengatasi hal tersebut dilakukan pendekatan teknik diskritisasi dengan menggunakan mean/standar deviasi.

METODE PENELITIAN

1. Desain Penelitian

Dalam penelitian ini peneliti menggunakan metode Algoritme *Naïve Bayes Clasiffier* dalam mendeteksi *malware.* Alur tahapan penelitian untuk pendeteksian *malware* bisa dilihat pada Gambar 1. yang diproses menggunakan tool *RapidMiner*.



Gambar 1. Desain Penelitian

Adapun prosesnya sebagai berikut: (1) Tahapan dimulai dengan membaca dataset berupa data dengan format data file csv.; (2) Dataset dilakukan proses *splitting* (pembagian) menjadi data Training dan testing.; (3) Sebelum dilakukan proses klasifikasi/ pendeteksian dilakukan pra pemrosesan data. Proses ini sangat penting dan akan sangat berpengaruh pada hasil pendeteksian dan lamanya waktu pemrosesan. Pada tahap pra-pemrosesan, ada dua tahapan yang dilakukan yaitu normalisasi data dan diskritisasi variabel. Normalisasi data dilakukan dengan tujuan mengurangi adanya kesalahan pada proses pembacaan data. Proses pendiskritan dalam *dataset* dilakukan untuk penyesuaian terhadap kemungkinan munculnya nilai kontinu dalam karakteristik *dataset* yang kecil sehingga akan membawa pengaruh dalam proses klasifikasi dengan metode *Naïve Bayes*.; (4) Tahap terakhir dilakukan proses training dengan Algoritme *Naïve Bayes*. Model dilatih untuk mengenali data *malware* dan *benign* menggunakan data Training. Kemudian model pendeteksi *malware* akan dites dengan data testing untuk mengidentifikasi *malware*.

2. Dataset

Penelitian ini menggunakan dataset dengan tipe file *CSV (Comma Separated Values)* yang berextensi file excel. *CSV (Comma Separated Values)* merupakan suatu format data dalam basis data dimana setiap record dipisahkan dengan tanda koma (,) atau titik koma (;). Data yang digunakan adalah data sekunder dari dataset *malware* yang diambil dari *website* kaggle milik saravana (Saravana, 2018). Jumlah dataset *malware* yang digunakan peneliti yaitu 100.000 data dengan 34 attribut seperti pada Tabel 1.

Tabel 1. Tipe Data Attribute Dataset *Malware*

No	*Attribut*	*Tipe Data*
1	*Millisecond*	*Numeric*
2	*Classification*	*String*
3	*State*	*Numeric*
4	*Usage_counter*	*Numeric*
5	*Prio*	*Numeric*
6	*Static_prio*	*Numeric*
7	*Normal_Prio*	*Numeric*
8	*Policy*	*Numeric*
9	*Vm_pgoff*	*Numeric*
10	*Vm_truncate_count*	*Numeric*
11	*Task_size*	*Numeric*
12	*Cached_hole_size*	*Numeric*
13	*Free_area_cache*	*Numeric*
14	*Mm_user*	*Numeric*
15	*Map_count*	*Numeric*
16	*Hiwater_rss*	*Numeric*
17	*Total_vm*	*Numeric*
18	*Shared_vm*	*Numeric*
19	*Exec_vm*	*Numeric*
20	*Reserved_vm*	*Numeric*
21	*Nr_ptes*	*Numeric*
22	*End_data*	*Numeric*
23	*Last_interval*	*Numeric*
24	*Nvcsw*	*Numeric*
25	*Nivcsw*	*Numeric*
26	*Min_flt*	*Numeric*
27	*Maj_flt*	*Numeric*
28	*Fs_excl_counter*	*Numeric*
29	*Lock*	*Numeric*
30	*Utime*	*Numeric*
31	*Stime*	*Numeric*
32	*Gtime*	*Numeric*
33	*Cgtime*	*Numeric*
34	*Signal_nvcsw*	*Numeric*

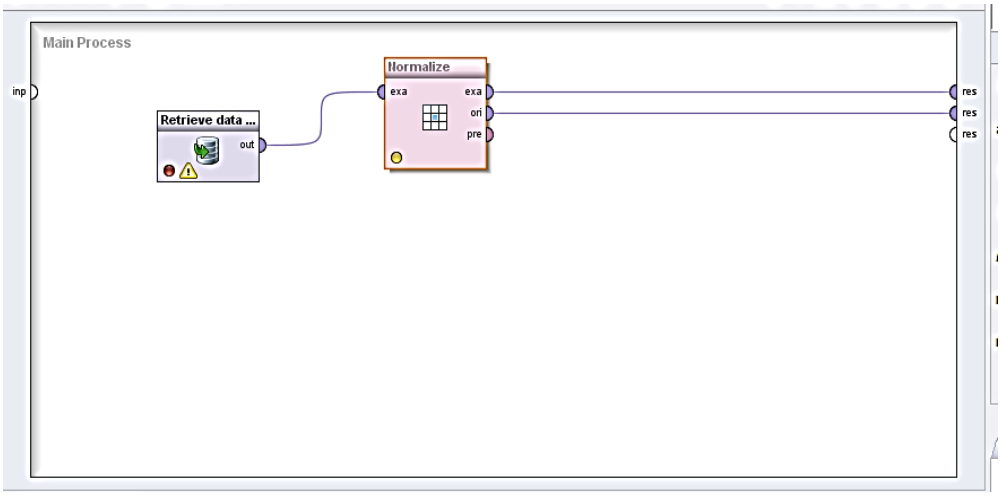
Tabel 2. Contoh isi Dataset *Malware*

Milli second	*Classi Fication*	*State*	*Usage _counter*	*prio*	*static_prio*	*Normal _prio*	*policy*	*vm_pgoff*	*vm_ truncate_count*
0	*Malware*	0	0	3069378560	14274	0	0	0	13173
1	*Malware*	0	0	3069378560	14274	0	0	0	13173
2	*Malware*	0	0	3069378560	14274	0	0	0	13173
3	*Malware*	0	0	3069378560	14274	0	0	0	13173
4	*Malware*	0	0	3069378560	14274	0	0	0	13173
5	*Malware*	0	0	3069378560	14274	0	0	0	13173
6	*Malware*	0	0	3069378560	14274	0	0	0	13173
7	*Benign*	319488	0	3069378560	23404	0	0	0	14856
8	*Benign*	319488	0	3069378560	23404	0	0	0	14856
9	*Benign*	319488	0	3069378560	23404	0	0	0	14856
10	*Benign*	319488	0	3069378560	23404	0	0	0	14856

Contoh isi dari tabel dataset bisa dilihat pada Tabel 2. yang memperlihatkan 10 baris pertama dan 10 kolom pertama dari dataset malware yang digunakan.

3. Normalisasi Data

Tahap normalisasi merupakan tahapan pra-pemrosesan yang dilakukan sebagai penyesuaian terhadap kemunculan nilai kontinu dalam fitur dataset yang akan mempengaruhi hasil proses klasifikasi dengan *Naïve Bayes*. Proses dilakukan dengan membaca dataset *malware* yang telah diinputkan sebelumnya ke dalam *tool* RapidMiner, kemudian mulai dilakukan tahap normalisasi pada Gambar 2. Proses normalisasi disini dimaksudkan untuk merubah jenis skala pengukuran yang dari data numerik. Proses normalisasi dilakukan dengan fungsi *Min-Max Normalize* yang akan mentransformasi data dengan rentang nilai (0,0) dan (0,1) seperti pada Tabel 3.



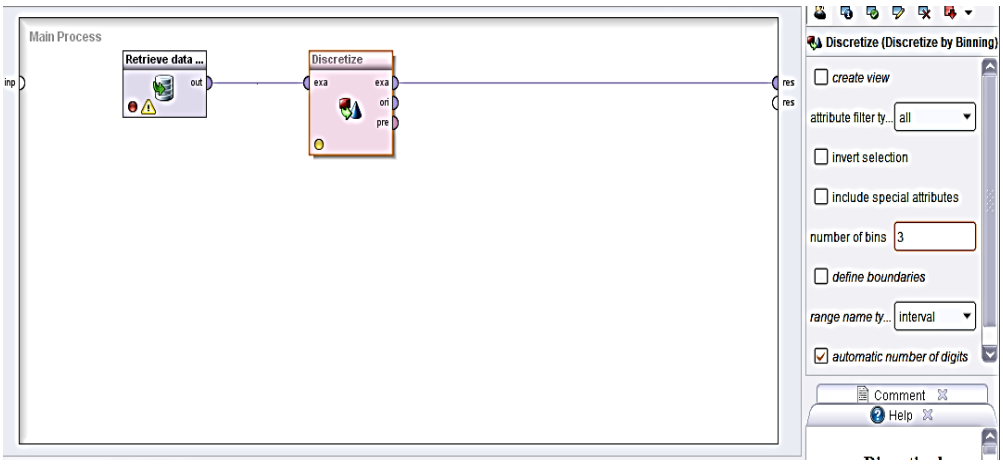
Gambar 2. Proses Normalisasi Data

Tabel 3. Hasil Normalisasi Data

Classification	*Milli Second*	*State*	*Usage_counter*	*prio*	*static_prio*	*Normal_prio*	*policy*	*vm_pgoff*	*vm_truncate_count*
malware	0	0	0	0.183	0.016	0	0	0	0.199
malware	0.001	0	0	0.183	0.016	0	0	0	0.199
malware	0.002	0	0	0.183	0.016	0	0	0	0.199
malware	0.003	0	0	0.183	0.016	0	0	0	0.199
malware	0.004	0	0	0.183	0.016	0	0	0	0.199
benign	0	0	0	0.206	0.138	0	0	0	0.289
benign	0.001	0	0	0.206	0.138	0	0	0	0.289
benign	0.002	0	0	0.206	0.138	0	0	0	0.289
benign	0.001	0	0	0.206	0.138	0	0	0	0.289
benign	0.002	0	0	0.206	0.138	0	0	0	0.289

4. Diskritisasi Data

Tahap pra-pemrosesan berikutnya adalah diskritisasi data. Proses diskritisasi ini dilakukan sebagai penyesuaian terhadap kemungkinan munculnya nilai kontinu dalam fitur dataset yang dapat mempengaruhi hasil proses pengklasifikasian dengan menggunakan metode *Naïve Bayes*. Dataset yang telah melalui tahap normalisasi dengan menggunakan metode Min-Max, dilakukan proses diskritisasi dengan teknik *binning*. Dalam penelitian ini dievaluasi proses *binning* kedalam 3 bagian atau interval yaitu (-1,0,1) dan 5 interval (-2, -1, 0, 1, 2). Proses diskritisasi dari data hasil normalisasi pada *tool* RapidMiner bisa dilihat pada Gambar 3.



Gambar 3. *Flow Diagram* Proses Diskritisasi Data

Setelah proses diskritisasi dijalankan maka akan menghasilkan data yang telah diproses ke dalam pendiskritan. Adapun variabel yang didiskritisasi yaitu variabel *millisecond, state, prio, static prio, vm_truncate_count, free_area_cache, mm_user, map_count, total_vm, shared_vm, exec_vm, reserved_vm, end_data, last interval, nvcsw, min_flt, maj_flt, fs_excl_counter, utime, stime, gtime.* Contoh hasil diskritisasi bisa dilihat pada Tabel 4. yang memperlihatkan hasil diskritisasi data 3 interval dan 5 interval. Diskritisasi data 3 interval akan mengelompokan data dalam 3 kelompok, sedangkan diskritisasi data 5 interval akan mengelompokkan data dalam 5 kelompok.

Tabel 4. Hasil Diskritisasi Data

Label	Diskritisasi 3-interval	Diskritisasi 5-interval
Millisecond	(-∞-0.3), (0.3-0.7), (0.7-∞)	(-∞-0.2), (0.2-0.4), (0.4-0.6), (0.6-0.8), (0.8-∞)
Prio	(-∞-0.3), (0.3-0.7), (0.7-∞)	(-∞-0.2), (0.2-0.4), (0.4-0.6), (0.6-0.8), (0.8-∞)
Static_Prio	(-∞-0.3), (0.3-0.7), (0.7-∞)	(-∞-0.2), (0.2-0.4), (0.4-0.6), (0.6-0.8), (0.8-∞)
Last_Interval	(-∞-0.3), (0.3-0.7), (0.7-∞)	(-∞-0.2), (0.2-0.4), (0.4-0.6), (0.6-0.8), (0.8-∞)
Map_Count	(-∞-0.3), (0.3-0.7), (0.7-∞)	(-∞-0.2), (0.2-0.4), (0.4-0.6), (0.6-0.8), (0.8-∞)
End_Data	(-∞-0.3), (0.3-0.7), (0.7-∞)	(-∞-0.2), (0.2-0.4), (0.4-0.6), (0.6-0.8), (0.8-∞)
STime	(-∞-0.2), (0.2-0.5), (0.5-∞)	(-∞-0.2), (0.2-0.3), (0.3-0.4), (0.4-0.6), (0.6-∞)

5. Metode Yang Digunakan

Metode penelitian yang digunakan dalam penelitian ini adalah Algoritme *Naïve Bayes*. Algoritme *Naïve Bayes* merupakan sebuah metoda klasifikasi menggunakan metode probabilitas dan statistik. *Naïve Bayes* merupakan sebuah pengklasifikasian probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan (Saleh, 2015). Algoritme mengunakan teorema Bayes mengasumsikan semua atribut independen tidak saling bergantung yang berdampak pada nilai variabel kelas. Algoritme *Naïve Bayes* memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai *Teorema Bayes* (Mustafa dkk., 2018). Persamaan dari metode *Naïve Bayes Classifier* bisa dilihat pada persamaan (1) dan (2).

$$P(H \mid X) = \frac{P(X \mid H).P(H)}{P(X)} \quad (1)$$

$P(H|X)$ yang dicari merupakan probabilitas hipotesis H berdasar kondisi X (posteriori probabilitas). Dimana $P(X|H)$ merupakan probabilitas X berdasarkan kondisi pada hipotesis H; X

merupakan data dengan *class* yang belum diketahui; H merupakan data hipotesis suatu *class* tertentu; $P(H)$ merupakan probabilitas hipotesis H (prior probabilitas) dan $P(X)$ merupakan Probabilitas X.

Proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis tersebut. Karena itu, metode *Naive bayes* persamaan (1) disesuaikan menjadi persamaan (2).

$$P(C \mid F_1 \ldots F_n) = \frac{P(C).P(F_1 \ldots F_n \mid C)}{P(F_1 \ldots F_n)} \quad (2)$$

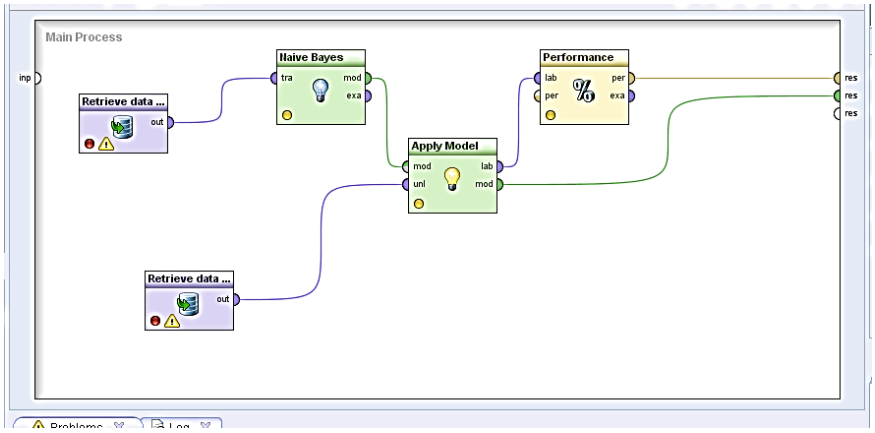
Di mana Variabel C merepresentasikan kelas, sementara variabel $F_1 \ldots F_n$ merepresentasikan karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi.

HASIL DAN PEMBAHASAN

Pada penelitian ini dicoba 3 skenario yaitu proses (1) Pendetksian dengan normalisasi tanpa diskritisasi, (2) Pendeteksian dengan diskritisasi 3 interval, dan yang ke (3) Pendeteksian dengan diskritisasi 5 interval. Dataset dilakukan pembagian dengan 90% data menjadi data training dan 10% data digunakan untuk pengujian (testing).

1. Normalisasi Tanpa Diskritisasi

Pada proses normalisasi tanpa diskritisasi ini data yang telah melalui proses normalisasi pada gambar 2 akan menjadi input untuk *Naïve Bayes Classifier*. Pada Gambar 4. terlihat alur proses dari pendeteksian *malware*. Tahap pra-pemrosesan akan menghasilkan *retrieve data* yang sudah melalui proses normalisasi baik untuk data training dan data testing. Dari data training akan dilatih dengan menggunakan Algoritme *Naïve Bayes* untuk menghasilkan model. Selanjutnya model dilatih dengan data testing.



Gambar 4. *Flow Diagram* Klasifikasi Tanpa *Binning*

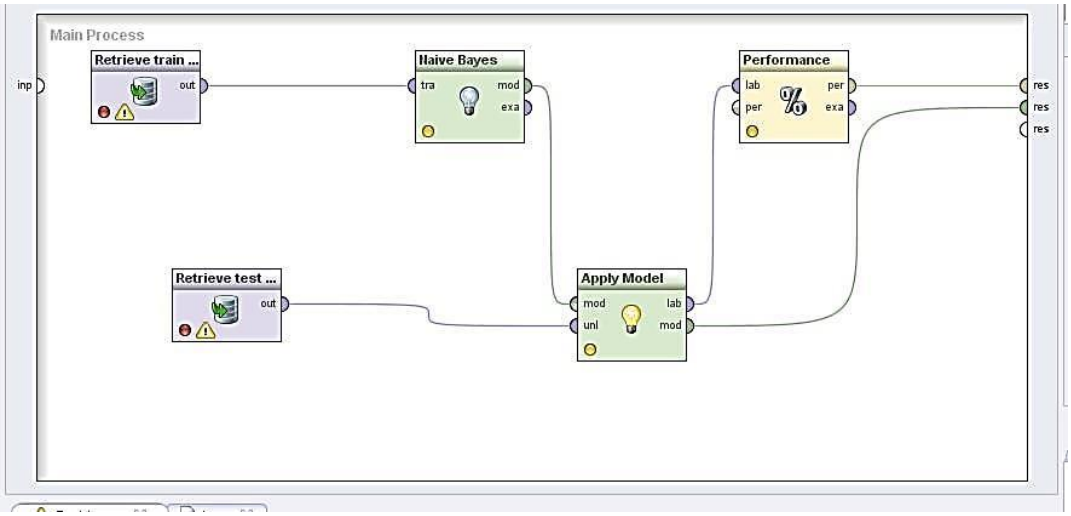
Hasil pengujian model yang pertama ini bisa dilihat pada Confusion Matrix pada Tabel 5. Dengan hanya menggunakan proses normalisasi tanpa diskritisasi pada proses pra-pemrosesan, Algoritme *Naïve Bayes Classifier* menghasilkan tingkat keakurasian sebesar 69,72%. Kemampuan pendeteksian dengan *dataset* ini masih belum optimal karena adanya atribut dengan nilai kontinu yang memiliki probabilitas kemunculan sangat kecil dalam data sehingga data itu tidak dapat diklasifikasikan dengan benar.

Tabel 5. Hasil *Confusion Matrix* Normalisasi tanpa Diskritisasi

Accuracy : 69.72 %			
	True malware	*True benign*	*Class precision*
Pred.malware	4707	2702	63,53 %
Pred.benign	326	2265	87,42 %
Class recall	93,52 %	45,60 %	

2. Diskritisasi dengan 3-interval

Pada pengujian model ke-2 ini dataset yang telah melalui tahap normalisasi dengan menggunakan metode Min-Max (Gambar 2), kemudian dilakukan proses diskritisasi dengan teknik *binning* kedalam 3 bagian atau interval (-1,0,1) (gambar 3). Proses klasifikasi akan membaca dataset training dan *testing* yang sudah melalui proses diskritisasi. Alur proses untuk klasifikasi dengan menggunakan teknik *binning* bisa dilihat pada Gambar 5. Jika dilihat pada tahapan hampir sama seperti proses pendeteksian tanpa teknik *binning*. Perbedaan proses *binning* ada pada data *retrieve* untuk data training dan testing yang sudah dikonversi menjadi data diskritisasi 3 variabel seperti pada Tabel 4.



Gambar 5. *Flow Diagram* Klasifikasi dengan Teknik Diskritisasi

Hasil *Confusion Matrix* pendeteksian *malware* dengan dataset yang telah dilakukan proses diskritisasi 3 interval bisa dilihat pada Tabel 6. Tingat keakurasian pendeteksian untuk data testing meningkat sebesar 78,16%. Peningkatan keakurasian pendeteksian dikarenakan kemunculan nilai kontinu dari dataset telah dihilangkan dengan teknik *binning*. Secara detail nilai presisi dan sensitivitas (*recall*) dari class *malware* dan *benign* bisa dilihat pada Tabel 6.

Tabel 6. Hasil *Confusion Matrik* untuk Diskritisasi 3-Interval

Accuracy : 78,16 %			
	True malware	*True benign*	*Class precision*
Pred.malware	4565	1716	72,68 %
Pred.benign	468	3251	87,42 %
Class recall	90,70 %	65,45 %	

3. Diskritisasi dengan 5-interval

Setelah proses normalisasi dan diskritisasi variabel 3 interval, model yang ketiga dataset yang telah dinormalisasi dilakukan teknik diskritisasi dengan *binning* kedalam bentuk 5 interval yaitu (-2,-1,0,1,2) seperti pada Tabel 4. Alur proses model pendeteksian hampir sama seperti pada teknik diskritisasi 3 variabel (Gambar 5). Hasil klasifikasi data *testing* bisa dilihat pada Tabel 7 memperlihatkan *Confusion Matrix* untuk model ke tiga.

Tabel 7. Hasil *Confusion Matrik* untuk Diskritisasi 5-interval

Accuracy : 79.97 %			
	True malware	*True benign*	*Class precision*
Pred.malware	3936	906	81,29 %
Pred.benign	1097	4061	78,73 %
Class recall	78,20 %	81,76 %	

Setelah proses diskritisasi 5 interval dengan teaknik *binning* dilakukan maka didapatkan hasil akurasi pendeteksian meningkat sebesar sebesar 79,97%. Demikian juga untuk nilai *Recall* (sensitivitas) dan presisi terutama untuk *class malware* hasilnya meningkat.

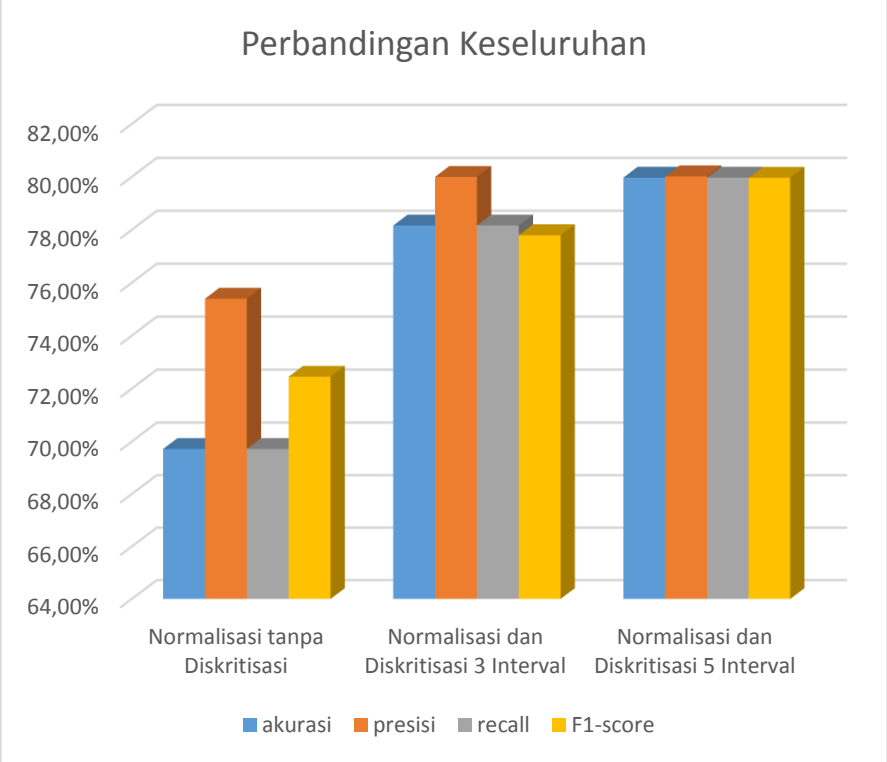
4. Perbandingan Hasil

Secara keseluruhan untuk nilai presisi, *recall* dan *F1-Score* dari ketiga model yang diuji bisa dilihat pada Tabel 8 dan Gambar 6. Pada tabel 8 terlihat bahwa tingkat persentase presisi keseluruhan yang dihasilkan dari dataset tanpa proses *binning* (pendiskritan) hanya sebesar 75,40% dan meningkat setelah didiskritisasi 5 interval menjadi sebesar 80.02%. Demikian juga untuk nilai Recall yang tanpa diskritisasi hanya sebesar 69,72% meningkat menjadi 78,16% pada hasil diskritisasi 3 interval dan sebesar 79,97% pada hasil diskritisasi 5 interval. Terutama untuk pengenalan *class malware* nilai presisi meningkat hingga 81,29%.

Tabel 8. Perbandingan hasil persentase dengan *Naïve Bayes Classifier*

Class	**Tanpa Diskrit**			**Diskrit 3-interval**			**Diskrit 5-interval**		
	Pre	***Recall***	***F1 Score***	***Pre***	***Recall***	***F1 Score***	***Pre***	***Recall***	***F1 Score***
Malware	63,53%	93,52%	75,66%	72,68%	90,70%	80,70%	81,29%	78,20%	79,22%
Benign	87,42%	45,60%	59,94%	87,42%	65,45%	74,86%	78,73%	81,78%	80,22%
Overall	75,40%	69,72%	72,45%	80,00%	78,16%	77,80%	80,02%	79,97%	79,97%

Untuk keseluruhan kinerja dari parameter yang dilihat maka pada Gambar 6 bisa dilihat untuk nilai akurasi, presisi keseluruhan, *recall* keseluruhan dan *F1-score* keseluruhan hasil diskritisasi 5 interval memperlihatkan hasil yang paling baik dibandingkan tanpa proses diskritisasi dan diskritisasi 3 interval.



Gambar 6. Hasil Evaluasi Keseluruhan

Hasil tersebut memperlihatkan dengan proses pra-pemrosesan dengan teknik diskritisasi dapat meningkatkan kinerja dari metode *Naïve Bayes Classifier* dibandingkan dengan penelitian sebelumnya yang menghasilkan persentase yang berbeda (Saravana, 2018). Proses diskritisasi ini sangat sesuai digunakan pada dataset *malware* yang atribut-atribut datanya sebagian besar merupakan data numerik, sehingga berpengaruh pada hasil pendeteksian.

KESIMPULAN DAN SARAN

Berdasarkan hasil pendeteksian dengan menguji tiga model dapat disimpulkan bahwa penerapan metode *Naïve Bayes* dengan pra-pemrosesan yang menggunakan teknik diskritisasi bisa meningkatkan hasil keakurasian pendeteksian *malware* dibandingkan dengan proses klasifikasi tanpa menggunakan teknik *binning* (diskritisasi). Tahap pendiskritan ini dapat menjadikan Algoritme *Naïve Bayes* menjadi tampak akurat di dalam mendeteksi *malware*.

Untuk penelitian selanjutnya model perlu diuji dengan menggunakan dataset yang besar dan lebih komplek. Selain itu menarik untuk dievaluasi pengaruh penggunaan teknik diskritisasi untuk pra-pemrosesan untuk Algoritme *machine learning* dan *data mining* lainnya.

DAFTAR PUSTAKA

Akbi, D. R., & Rosyadi, A. R. (2018). Analisis Klasterisasi Malware: Evaluasi Data Training Dalam Proses Klasifikasi Malware. *Jurnal ELTIKOM*, *2*(2), 58–66. https://doi.org/10.31961/eltikom.v2i2.88

Amalia, N., Shaufiah, S., & Sa'adah, S. (2015). Penerapan teknik data mining untuk klasifikasi ketepatan waktu lulus mahasiswa teknik informatika universitas telkom menggunakan Algoritme *naive bayes* classifier. *EProceedings of Engineering*, *2*(3).

Anam, C., & Santoso, H. B. (2018). Perbandingan kinerja Algoritme c4. 5 dan *naive bayes* untuk klasifikasi penerima beasiswa. *ENERGY*, *8*(1), 13–19.

Asih, T. S. N., Waluya, B., & Supriyono, S. (2018). Perbandingan finite difference method dan finite element method dalam mencari solusi persamaan diferensial parsial. *PRISMA, Prosiding Seminar Nasional Matematika*, *1*, 885–888.

Cahyanto, T. A., Wahanggara, V., & Ramadana, D. (2018). Analisis dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis dan Malware Analisis Statis. *JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia*, *2*(1), 12.

Haryati, S., Sudarsono, A., & Suryana, E. (2015). Implementasi data mining untuk memprediksi masa studi mahasiswa menggunakan Algoritme c4. 5 (studi kasus: Universitas dehasen bengkulu). *Jurnal Media Infotama*, *11*(2).

Herlambang, S., & Basuki, S. (2019). Deteksi Malware Android Berdasarkan System Call Menggunakan Algortima Support Vector Machine. *Prosiding SENTRA (Seminar Teknologi dan Rekayasa*, *4*, 157–165.

Huaturuk, N. R. S., Rahmadani, R. D., & Ak, D. J. (2018). Komparasi Akurasi *Naïve Bayes* dan Support Vector Machine (SVM) untuk Rekomendasi Produk in Fashion Dress. *Conference on Electrical Engineering, Telematics, Industrial technology, and Creative Media (CENTIVE)*, 168–173.

Mustafa, M. S., Ramadhan, M. R., & Thenata, A. P. (2018). Implementasi data mining untuk evaluasi kinerja akademik mahasiswa menggunakan Algoritme *naive bayes* classifier. *Creative Information Technology Journal*, *4*(2), 151–162.

Nasari, F., & Darma, S. (2015). Penerapan K-Means Clustering pada Data Penerimaan Mahasiswa Baru (Studi Kasus: Universitas Potensi Utama). *Seminar Nasional Teknologi Informasi dan Multimedia*, *3*, 73–78.

Novrianda, R., Kunang, Y. N., & Shaksono, P. H. (2014). Analisis Forensik Malware pada Platform Android. *KonferensiNasional Ilmu Komputer (KONIK)*, 377–385.

Nugroho, P. A. (2016). Penanganan dan Pendeteksian Penyebaran Malware Menggunakan X Ray PC Pada Sistem Operasi Windows XP (Studi Kasus Worm "MR. COOLFACE"). *Jurnal Inovasi Informatika*, *1*(2), 32–41.

Saleh, A. (2015). Implementasi metode klasifikasi *naive bayes* dalam memprediksi besarnya penggunaan listrik rumah tangga. *Creative Information Technology Journal*, *2*(3), 207–217.

Sandag, G. A., Leopold, J., & Ong, V. F. (2018). Klasifikasi *Malicious* Websites Menggunakan Algoritme K-NN Berdasarkan Application Layers dan Network Characteristics. *CogITo Smart Journal*, *4*(1), 37. https://doi.org/10.31154/cogito.v4i1.100.37-45

Saravana, N. (2018, April 12). *Malware Detection*. https://www.kaggle.com/nsaravana/malware-detection

Setiawan, F. G. N. D., Ijtihadie, R. M., & Studiawan, H. (2017). Pendeteksian Malware pada Lingkungan Aplikasi Web dengan Kategorisasi Dokumen. *Jurnal Teknik ITS*, *6*(1), 71–74. https://doi.org/10.12962/j23373539.v6i1.22163

Wirawan, I. N. T., & Eksistyanto, I. (2015). Penerapan *Naive bayes* pada Intrusion Detection System dengan Diskritisasi Variabel. *JUTI: Jurnal Ilmiah Teknologi Informasi*, *13*(2), 182. https://doi.org/10.12962/j24068535.v13i2.a487