**PROGRAM STUDI TEKNIK INFORMATIKA**

**ANALISIS DAN PENGEMBANGAN BASIS DATA INVENTORI BARANG**

**PADA RIA KENCANA UNGU (RKU) KOMPUTER**

**MUHAMMAD MAXSUM**

**09142242**

**Skripsi ini diajukan sebagai syarat memperoleh gelar Sarjana Komputer**

**di Universitas Bina Darma**



**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BINA DARMA**

**2013**



**ANALISIS DAN PENGEMBANGAN BASIS DATA INVENTORI BARANG**

**PADA RIA KENCANA UNGU (RKU) KOMPUTER**

**MUHAMMAD MAXSUM**

**09142242**

**Skripsi ini diajukan sebagai syarat memperoleh gelar Sarjana Komputer**

**di Universitas Bina Darma**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BINA DARMA**

**PALEMBANG**

**2013**

**LEMBAR PENGESAHAN**

**ANALISIS DAN PENGEMBANGAN BASISDATA INVENTORI BARANG**

**PADA RIA KENCANA UNGU (RKU) KOMPUTER**

**MUHAMMAD MAXSUM**

**09142242**

**Telah diterima sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer Pada Program Studi Teknik Informatika**

**Palembang, Oktober 2013**

**Program Studi Teknik Informatika**

**Fakultas Ilmu Komputer**

**Universitas Bina Darma**

**Pembimbing I, Dekan,**

**(Afriyudi, M.Kom..) (M. Izman H., S.T., M.M., Ph.D.)**

**Pembimbing II,**

**(Widiyanto, M.M.,M.Kom)**

**LEMBAR PERSETUJUAN**

Skripsi berjudul “**Analisis Dan Pengembangan Basisdata Inventori Barang Pada Ria Kencana Ungu (RKU) Komputer”** oleh  **MUHAMMAD MAXSUM (09142242)”,** telah dipertahankan di depan Komisi Penguji pada tanggal **30 Juli**

**2013**.

# Komisi Penguji

1. Ketua : **Afriyudi, M.Kom.** (...........……..………..)

2. Sekretaris : **Widiyanto, M.M.,M.Kom.** **(**............……..………..)

3 Anggota : **Fatoni, M.M.,M.Kom.** **(**............……..………..)

4. Anggota : **Firamon Syakti, M.M., M.Kom.** **(**............……..………..)

**Mengetahui, Oktober 2013**

**Program Studi Teknik Informatika**

**Fakultas Ilmu Komputer**

**Universitas Bina Darma**

**Ketua,**

**(Syahril Rizal, S.T., M.M., M.Kom.)**

**PERNYATAAN**

Saya yang bertanda tangan dibawah ini :

 Nama : MUHAMMAD MAXSUM

 NIM : 09142242

Dengan ini menyatakan bahwa:

1. Karya tulis saya (tugas akhir/ sekeripsi) ini adalah asli dan belum pernah diajukan untuk mendapat gelar akademik baik (sarjana) di Universitas Bina Darma atau perguruan tinggi lain;
2. Karya tulis ini murni gagasan, rumusan dan penelitian saya sendiri dengan arahan tim pembimbing;
3. Di dalam karya tulis ini tidak terdapat karya atau pendapat yang telah di tulis atau dipublikasikan nama pengarang dan memesukan kedalam daftar rujukan;
4. Saya bersedia tugas akhir sekeripsi, yang saya hasilkan dicek keaslianya menggunakan plagiarism checker sera di unggah ke interne, sehingga dapat di akses publik secara daring;
5. Surat pernyataan ini saya tulis dengan sungguh-sungguh dan apbila terbukti melakukan penyimpangan atau ketidak benaran dalam pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan peratulan dan undang-undangan yang berlaku.

Demikian surat pernyataan ini saya buat agar dapat dipergunakan sebagaimana mestinya.

Palembang, Desember 2013

Yang membuat pernyataan,

 **MUHAMMAD MAXSUM**

 **09142242**

**Motto dan Persembahan**

### MOTTO

* **Maju Terus Pantang Mundur**
* **Manjadda Wajadda (siapa yang bersungguh-sungguh maka dia akan berhasil)**

 **Kupersembahkan Kepada :**

* Orang Tua yang selalu mendoakanku disetiap sholat malamnya.
* Seseorang yang tercinta yang selalu memberi dorongan dan do’a untukku
* Saudaraku yang selalu kusayangi
* Teman-teman seperjuangan.
* Almamaterku

**ABSTRAK**

Perkembangan teknologi informasi menjadi salah satu pemicu terjadinya perubahan terhadap kebutuhan suatu sistem informasi yang efisien dan efektif. Hal ini tidak lepas dari basis data yang merupakan kumpulan tabel-tabel yang berisi data-data yang akan diolah dan ditampilkan melalui sebuah sistem informasi. RKU Komputer merupakan sebuah perusahaan yang bergerak di bidang penjualan barang-barang computer beserta aksesorisnnya. Adapun judul yang dipilih yaitu **“Analisis Dan Pengembangan Basisdata Inventori Barang Pada Ria Kencana Ungu (Rku) Komputer”.** Tujuan dari penelitian ini membuat rancangan basis datainventori barang agar dapat dimanfaatkan untuk pembuatan sistem atau aplikasi inventori barang. Metode yang dipakai dalam penelitian ini adalah *Database System Development Lifecycle* menurut pendekatan *Connolly* dan *Begg*. Hasil dari penelitian ini dapat mempermudah dalam pengembangan aplikasi sistem inventori barang karena sudah di lengkapi dengan *Trigger* yang memungkinkan beberapa proses pengolahan data dapat dilakukan di *database* dan penggunaan *View* dapat meringankan proses *Query* yang memerlukan relasi lebih dari satu tabel.

Kata kunci : Basis Data, Inventori, Connolly and Begg

**KATA PENGANTAR**

****

Puji syukur kehadirat Allah SWT karena berkat rahmat dan karunia-Nya jualah sehingga penulis dapat menyelesaikan skripsi ini guna memenuhi salah satu syarat yang telah ditetapkan dalam menempuh pendidikan di Universitas Bina Darma. Palembang.

Dalam penulisan ini, penulis telah berusaha semaksimal mungkin untuk memberikan dan menyajikan yang terbaik, namun penulis menyadari bahwa dalam pembuatan skripsi ini masih jauh dari sempurna, hal ini dikarenakan terbatasnya pengetahuan. Oleh karena itu dalam rangka melengkapi kesempurnaan dari skripsi ini penulis mengharapkan adanya saran dan kritik yang diberikan bersifat membangun.

Pada kesempatan yang baik ini, tak lupa penulis menghaturkan terima kasih kepada semua pihak yang telah memberikan bimbingan, pengarahan, nasehat dan pemikiran dalam penulisan skripsi ini, terutama kepada :

1. Prof. Ir. H. Bochari Rahman, M.Sc. selaku Rektor Universitas Bina Darma Palembang.
2. M. Izman, H., ST., MM., P.hd, selaku Dekan Fakultas Ilmu Komputer
3. Syahril Rizal, ST., MM., M.Kom., Ketua Program Studi Teknik Informatika.
4. Afriyudi, M.Kom.**.**, selaku pembimbing I yang telah memberikan bimbingan proposal penulisan skripsi ini.
5. Widiyanto, M.M.,M.Kom., selaku pembimbing II yang telah memberikan bimbingan proposal penulisan skripsi ini.
6. Orang Tua dan Saudara-saudaraku tercinta serta seluruh teman dan sahabat-sahabatku yang selalu memberikan dorongan, masukan dan bantuan baik moril maupun materil yang tak ternilai harganya.

 Akhirnya penulis hanya dapat berharap semoga skripsi ini dapat bermanfaat bagi penulis dan semua pembaca. Semoga Allah SWT melimpahkan semua rahmat dan karunia-Nya kepada kita semua. Amin.

Palembang, Oktober 2013

 MUHAMMAD MAXSUM

**DAFTAR ISI**

Halaman

**HALAMAN JUDUL i**

**LEMBAR PENGESAHAN ii**

**LEMBAR PERSETUJUAN iii**

**PERNYATAAN iv**

**MOTTO DAN PERSEMBAHAN v**

**ABSTRAK vi**

**KATA PENGANTAR vii**

**DAFTAR ISI ix**

**DAFTAR TABEL xi**

**DAFTAR GAMBAR xii**

**BAB I PENDAHULUAN**

 1.1 Latar Belakang 1

 1.2 Perumusan Masalah 2

 1.3 Tujuan Penelitian 3

 1.4 Batasan Masalah 3

 1.5. Manfaat Penelitian 3

 1.6. Metodologi Penelitian 4

 1.6.1 Waktu dan Tempat Penelitian 4

 1.6.2 Alat dan Bahan 4

 1.6.3 Metode Penelitian dan Pengumpulan Data 4

 1.6.4 Metode Pengembangan Basis Data 5

 1.7 Sistematika Penulisan 7

**BAB II TINJAUAN PUSTAKA**

 2.1 Tinjauan Umum 8

 2.1.1. Sejarah CV. Ria Kencana Ungu (RKU) Komputer 8

 2.2. Landasan Teori 9

 2.2.1. Analisis 9

 2.2.2. Pengembangan 9

 2.2.3. Data dan Basis Data 10

 2.2.4. Database Management System (DBMS) 11

 2.2.5. *Entity Relationship Modeling* (*ER Modeling*) 12

 2.2.6. *Inventory* (Persedian Barang) 13

 2.2.7. *Mysql* 13

 2.2.8. *Stored Procedure/ Function* 14

 2.2.9. *Trigger* 14

 2.2.10. *View* 14

2.3. Penelitian Sebelumnya............................................................. 15

**BAB III ANALISA DAN PERANCANGAN**

3.1. Analisa dan Pengumpulan Kebutuhan Basis Data 17

 3.1.1. Analisis Sistem yang Berjalan 17

 3.2. *Database Planning* (Perencanaan Basis Data) 18

 3.2.1. Pernyataan Misi 18

 3.2.2. Pernyataan Tujuan 19

 3.2.3. Permasalahan yang dihadapi 19

 3.3. Desain Basis Data 20

 3.3.1. Perencanaan Basis Data Konseptual 20

 3.3.1.1 Identifikasi Tipe Entitas 20

 3.3.1.2 Pengidentifikasian Relasi 21

 3.3.1.3 Identifikasi dan Asosiasi Atribut Suatu Entitas 23

 3.3.1.4 Identifikasi Domain Atribut 24

 3.3.1.5 Menentukan *Primary* *Key dan Candidat Key* 25

 3.3.2 Perencanaan Basis Data *Logical*................................ 25

 3.3.2.1 Menentukan Model Logika Data 25

 3.3.2.2 Normalisasi 26

 3.3.3 Perencanaan Basis Data Fisikal 29

 3.3.3.1 Struktur Tabel Hasil Normalisasi 29

**BAB IV HASIL DAN PEMBAHSAN**

 4.1. Hasil 32

 4.2. Pembahasan 33

 4.2.1. Pembuatan Basis Data dengan MySQL 33

 4.2.2. Pembahasan *Query* Basis Data 35

 4.2.3. *Trigger* 43

 4.2.4. *View* 47

**BAB V KESIMPULAN DAN SARAN**

5.1 Kesimpulan 50

 5.2 Saran 51

**DAFTAR PUSTAKA** 52

**DAFTAR TABEL**

 **Halaman**

Tabel 2.1. Penelitian Sebelumnya 15

Tabel 3.1. Masalah dan Usulan Pemecahan Masalah 19

Tabel 3.2. Tabel Entitas 21

Tabel 3.3. Tabel Relasi 22

Tabel 3.4. Tabel Asosiasi Atribut 23

Tabel 3.5. Tabel Domain Atribut 24

Tabel 3.6. Tabel *Primary Key* dan *Canditate Key* 25

Tabel 4.1. Tabel *Entitas* Barang 35

Tabel 4.2. Tabel Stok 36

Tabel 4.3. Tabel *Suplier* 38

Tabel 4.4. Tabel Harga beli 39

Tabel 4.5. Tabel Harga Jual 40

Tabel 4.6. Tabel Pembelian 41

Tabel 4.7. Tabel Retur Pembelian 42

Tabel 4.8. Tabel Bagian Gudang 43

**DAFTAR GAMBAR**

 **Halaman**

Gambar 1.1. Connoly dan Begg 6

Gambar 2.1. Simbol-simbol ER-Diagram 12

Gambar 3.1. Alur Proses Dokumen Sistem Yang Berjalan……………….18

Gambar 3.2. Entity Relational Diagram 20

Gambar 3.3. ERD Konseptual 22

Gambar 3.4 Relasi Barang Dengan Pembelian…………………………...25

Gambar 3.5 Relasi Pembelian Dengan Retur Pembelian………………...26

Gambar 3.6 Relasi Bagian Gudang Dengan Pembelian………………….26

Gambar 3.7. ERD Fisical 31

Gambar 4.1. Hasil Perancangan Basis Data Dengan MySQL 33

Gambar 4.2. Tampilan Awal MySQL 34

Gambar 4.3. Tampilan Membuat Tabel 34

Gambar 4.4. *Query* Tabel Barang 35

Gambar 4.5. Tampilan Tabel Barang 36

Gambar 4.6. *Query* Tabel Stok 36

Gambar 4.7. Tampilan Tabel Stok 37

Gambar 4.8. *Query* Tabel *Suplier* 37

Gambar 4.9. Tampilan Tabel *Suplier* 38

Gambar 4.10. *Query* Tabel Harga Beli 38

Gambar 4.11. Tampilan Table Harga Beli 39

Gambar 4.12. *Query*  Tabel Harga Jual 39

Gambar 4.13. Tampilan Tabel Harga Jual 40

Gambar 4.14. *Query* Tabel Pembelian 40

Gambar 4.15. Tampilan Tabel Pembelian 41

Gambar 4.16. *Query* Tabel Retur Pembelian 42

Gambar 4.17. Tampilan Tabel Retur Pembelian 42

Gambar 4.18. *Query* Tabel Bagian Gudang 43

Gambar 4.19. Tampilan Tabel Bagian Gudang 43

Gambar 4.20. *Syntax create trigger Auto\_update\_*penambahan\_stok 44

Gambar 4.21. Tabel Stok Sebelum *Insert* Pembelian 44

Gambar 4.22. Tabel Pembelian Ketika *Insert* 45

Gambar 4.23. Tabel Stok Setelah *Insert* Pembelian 45

Gambar 4.24. *Syntax create trigger Auto\_update\_*pengurangan\_stok 45

Gambar 4.25. Tampilan Stok Sebelum *Insert Retur* Barang 46

Gambar 4.26. Tabel Retur Barang Ketika *Insert* 46

Gambar 4.27. Tabel Stok Setelah *Insert* Retur Barang 46

Gambar 4.28. *Syntax create view* Detail Barang 47

Gambar 4.29. Struktur View Detail Barang 47

Gambar 4.30. *Syntax create view* Retur Pembelian 48

Gambar 4.31. Struktur Tabel view Detail Retur Pembelian 48

Gambar 4.32. *Syntax create view* Detail Pembelian 49

Gambar 4.33. Strutur Tabel *view* Detail Pembelian 49

**BAB I**

**PENDAHULUAN**

1. **Latar Belakang**

Pada saat ini perkembangan teknologi informasi menjadi salah satu pemicu terjadinya perubahan terhadap kebutuhan suatu sistem informasi yang efisien dan efektif. Hal ini tidak lepas dari basis data yang merupakan kumpulan tabel-tabel yang berisi data-data yang akan diolah dan ditampilkan melalui sebuah sistem informasi. Pemrosesan basis data sebagai perangkat andalan sangat diperlukan oleh berbagai institusi dan perusahaan, dalam pengembangan sistem informasi diperlukan basis data sebagai media penyimpanan data, kehadiran basis data dapat meningkatkan daya saing perusahaan. Untuk mengelola sumber informasi yang pertama kali dilakukan adalah merancang suatu aplikasi basis data agar informasi yang ada dapat digunakan secara maksimal mungkin.

Perancangan suatu basis data yang berfungsi menyimpan data dalam proses pengelolaan penatausahaan aset fasilitas kantor merupakan sebuah proses yang wajib dilakukan dalam memulai sebuah sistem, dengan mentransformasikan proses manual menjadi suatu kumpulan tabel-tabel yang terstruktur.

RKU Komputer merupakan sebuah perusahaan yang bergerak di bidang penjualan barang-barang computer beserta aksesorisnya. Semakin berkembangnya RKU Komputer, semakin banyak pula barang-barang yang keluar dan masuk pada bagian gudang. Selama ini pencatatan barang keluar dan masuk masih dilakukan dengan pencatatan pada kertas-kertas kemudian dimasukan kedalam aplikasi komputer. Hal ini mempersulit bagian gudang dalam melakukan pembuatan laporan barang keluar, barang masuk dan stok persedian barang yang ada di gudang. oleh karena itu perlu dibuatkan basis data inventori barang dengan struktrur yang baik sehingga data yang tersimpan lebih aman, terintegrasi dan lebih efisien. Kemudian perancangan basis data yang baik dan dilengkapi dengan *stored procedure/function, trigger, dan view* dapat mempermudah tahap pengembangan aplikasi inventori barang.

Berdasarkan uraian-uraian di atas penulis bermaksud untuk mengangkat permasalahan tersebut sebagai bahan penelitian untuk proposal. Adapun judul yang dipilih yaitu **“Analisis Dan Pengembangan Basis Data Inventori Barang Pada Ria Kencana Ungu (Rku) Komputer”.**

1. **Perumusan Masalah**

Berdasarkan uraian di atas, merumuskan yang ada untuk dijadikan titik tolak pada pembahasan skripsi ini adalah “bagaimana menganalisis dan mengembangkan basisdata inventori barang pada Ria Kencana Ungu (RKU) Komputer?”.

1. **Tujuan Penelitian**

Adapun tujuan penelitian ini adalah sebagai berikut :

1. Membuat rancangan basis datainventori barang agar dapat dimanfaatkan untuk pembuatan sistem atau aplikasi inventori barang.
2. Dapat mengembangkan hasil analisis dan perancangan menjadi data base inventoripada *Mysql*.
3. Penelitian ini dapat menambah wawasan bagi pembaca dan dapat dipergunakan sebagai referensi untuk penelitian-penelitian selanjutnya
4. **Batasan Masalah**

Agar penelitian ini terarah dan tidak meluas, maka penulis memberikan beberapa batasan masalah sebagai berikut :

1. Data yang digunakan hanya meliputi pada proses barang keluar dan masuk pada RKU Komputer Palembang.
2. Basis data yang nantinya akan dipakai untuk merancang data tersebut adalah *MySQL.*
3. **Manfaat Penelitian**

Adapun manfaat penelitian ini adalah sebagai berikut :

1. Memberikan kemudahan bagi pengembang program dalam pemanfaatan basis data inventori.
2. Bagi penulis sendiri dapat mengembangkan ilmu komputer yang telah ditempuh selama penelitian.
3. **Metodologi Penelitian**
	1. **Waktu dan Tempat Penelitian**

Penelitian analisis dan pengembangan basisdata inventori barang pada Ria Kencana Ungu (RKU) komputer akan dilakukan mulai bulan Maret 2012 sampai dengan Agustus 2012.

* 1. **Alat dan Bahan**

Kebutuhan alat dan bahan yang digunakan dalam implementasi pendekatan semantik objek model pada perancangan sistem basis data Inventory barang CV. RKU Komputer yang digunakan meliputi alat serta bahan–bahan penunjang lainnya.

1. *Hadrware* yaitu :

1. *Processor Intel Core 2 Duo*
2. *RAM 1 GB, Hardisk 320 GB,*
3. *Monitor SVGA Color*

2. *Software* Yaitu :

1. *Microsoft Windows 7* atau sesuai dengan kebutuhan.
2. *MySql 5*
	1. **Metode Penelitian dan Pengumpulan Data**

Metode penelitian yang digunakan pada penelitian ini yaitu Metode deskriptif yaitu metode yang menggambarkan atau menguraikan keadaan situasi pada tempat observasi, melakukan penelitian serta kemudian melakukan analisis sehingga dapat diperoleh suatu kesimpulan. Penelitian deskriptif bertujuan untuk menggambarkan secara sistematis fakta dan karakteristik objek atau subjek yang diteliti secara tepat. Menurut Whitney (1960: 160) metode deskriptif adalah pencarian fakta dengan interpretasi yang tepat.

Dalam melakukan penelitian untuk mendapatkan data dan informasi, maka metode yang digunakan dalam proses pengumpulan data dilakukan sebagai berikut :

1. Metode Observasi, dalam hal observasi ini yang akan di observasi adalah mempelajari permasalahan tentang analisis dan pengembangan basisdata inventori barang pada ria kencana ungu (RKU) komputer.
2. Metode Studi Pustaka, metode yang dilakukan adalah dengan cara mancari bahan yang mendukung dalam pendefinisian masalah melalui buku-buku, *internet*, yang erat kaitannya dengan objek permasalahan.
3. Metode Wawancara, metode ini dilakukan dengan cara mewawancarai secara langsung para pegawai serta staf-staf yang ada.
	1. **Metode Pengembangan Basis Data**

 Metode analisis dan perancangan yang dipakai *Database System Development Lifecycle* menurut pendekatan *Connolly* dan *Begg* (2005:284) yang terdiri dari beberapa tahapan, yaitu :

a. Database Planning

b. System Definition

c. Requirements Collection and Analysis

d. Database Design

e. DBMS Selection (optional)

f. Application Design

g. Prototyping (optional)

h. Implementation

i. Data Conversion and Loading

j. Testing

 k. Operational Maintenance

Adapun gambar dari Connolly dan Begg seperti di bawah ini :**9.2**

**Gambar 1.1 Connolly dan Begg**

**(sumber : Connolly and begg, 2005:284)**

1. **Sistematika Penulisan**

### BAB I Pendahuluan

### Pada bab ini menguraikan tentang Latar Belakang, Perumusan Masalah, Tujuan dan Manfaat Penelitian, Batasan Masalah, Metodologi Penelitian, Sistematika Penulisan.

### BAB II Tinjauan Pustaka

### Dalam bab ini akan dijelaskan beberapa landasan teori yang digunakan penulis dalam penyusunan skripsi.

### BAB III Analisis Dan Perancangan Sistem

### Pada bab ini akan membahas tentang, Analisis Sistem, Rancangan Sistem Analisis, Detail Desain, Desain Input, Desain Output, Penerapan dan Pemeliharaan.

### BAB IV Hasil Dan Pembahasan

### Dalam bab ini diuraikan tentang hasil dari penelitian dan pembahasan berbagai masalah yang dihadapi.

### BAB V Kesimpulan Dan Saran

### Bab ini menguraikan tentang kesimpulan dari keseluruhan bab serta mencoba memberikan saran-saran yang mungkin berguna untuk mengatasi masalah yang dihadapi.

**BAB II**

**TINJAUAN PUSTAKA**

**2.1. Tinjauan Umum**

**2.1.1. Sejarah CV. Ria Kencana Ungu (RKU) Komputer**

Tanggal 29 November 2001 merupakan awal berdirinya CV. Ria Kencana Ungu (RKU) Komputer. Ria adalah nama adik perempuan dari pendiri perusahaan, kencana ungu adalah kendaraan yang dikendarai oleh nyi loro yang berwarna ungu. Pendiri merupan tamatan S1 di Universitas Tridinanti pada tahun 2000, pada saat tamat pendiri melamar kerja di mana-mana akhirnya pada bulan juni tahun 2000 pendiri bekerja di batam pada perusahaan Jepang Matsusitha sebagai super visor bagian flame dibagian computer dan laptop. Pendiri di training di Jepang bulan September 2000 sampai januari 2001, ketika kembali ke Indonesia, jabatan pendiripun naik menjadi super intendent tetapi pendiri merasa jenuh bekerja di perusahaan tersebut akhirnya pada bulan juli pendiri memutuskan untuk risign dari perusahaan tersebut disetujui pada bulan oktober. Tanggal 15 Oktober 2001 pendiri resmi keluar dari matsusitha dan kembali ke Palembang tanggal 18 Oktober 2001. Pendiri mulai mencari tempat usaha di Plaju dan mendapatkan tempat di Jln. A. Yani No. 24J. awalnya pendiri hanya menerima servis laptop, computer dan printer serta penjualan ATK. Perusahaan ini semakin maju dan mulai menerima berbagai jenis servis lainnya dan menjual laptop, PC, Printer dan aksesoris computer lainnya.

**2.2. Landasan Teori**

**2.2.1. Analisis**

Analisis yaitu proses untuk menentukan kelemahan dan kelebihan suatu sistem, dan mencara asal dari permasalahan yang terjadi selanjutnya memikirkan alternatif untuk pemecahan masalah serta mencari solusi terbaik untuk pemecahan masalah tersebut, Indrajani (2011:8).

Dari teori di atas dapat diambil kesimpulan bahwa penelitian analisis adalah suatu penyelidikan terhadap suatu sistem untuk menentukan permasalahan dan selanjutnya mencari solusi terbaik untuk pemecahan masalah tersebut.

**2.2.2. Pengembangan**

Menurut Senjaya (2008:1) Pengembangan adalah memperdalam dan memperluas pengetahuan yang telah ada. (Sugiono, 2004) Pengembangan adalah kegiatan ilmu pengetahuan dan teknologi yang bertujuan memanfaatkan kaidah dan teori ilmu pengetahuan yang telah terbukti kebenarannya untuk meningkatkan fungsi, manfaat, dan aplikasi ilmu pengetahuan dan teknologi yang telah ada, atau menghasilkan teknologi baru.

Pengembangan adalah kegiatan tidak lanjut penelitian untuk memanfaatkan hasil-hasil penelitian serta mendapatkan informasi tentang cara-cara menggunakan teori dan proses untuk tujuan-tujuan praktis dan kegunaan.

Penelitian dan Pengembangan adalah upaya kreatif dan sistematik yang dilakukan dalam meningkatkan khasanah ilmu pengetahuan dan teknologi (IPTEK) termasuk yang berkaitan dengan masalah kemanusiaan, sosial dan budaya serta pemanfaatannya untuk berbagai kepentingan.

**2.2.3. Data dan Basis Data**

Menurut Fathansyah (2002:2) Basis data terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/ berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Basis Data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedekian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Sedangkan *Database* atau basis data adalah koleksi dari data-data yang saling berelasi, Kusrini (2007:2), dengan cara sedemikian rupa sehingga mudah dalam disimpan dan dimanipulasi, Nugroho (2004:41).

**2.2.4. Database Management System (DBMS)**

Menurut Connolly dan Begg (2005) DBMS adalah suatu sistem piranti lunak yang memungkinkan *user* dapat mengidentifikasi, membuat, memelihara, dan mengatur akses dari basis data.

Basis data terdistribusi, atau basis data tersebar (*Distributed Database*) merupakan basis data yang secara logika menyatu yang tersebar secara fisik pada komputer-komputer yang terletak di beberapa lokasi sekaligus yang terhubung satu sama lain pada jaringan komunikasi data, Nugroho (2004:449).

Sedangkan Heriyanto (2004:465), menyatakan basis data terdistribusi (DDBMS = *Distibuted Database Management System*) adalah kumpulan data yang digunakan bersama yang saling terhubung secara logik tetapi tersebar secara fisik pada suatu jaringan komputer.

Dalam DDBMS memiliki karakteristik sebagi berikut:

* + - 1. Kumpulan data yang digunakan bersama yang secara logik saling terhubung yang tersebar pada sejumlah komputer yang berbeda.
			2. Komputer-komputer yang saling dihubungkan menggunakan jaringan telekomunikasi.
			3. Data pada masing-masing komputer terkendali dalam satu *Database Manajemen System*.
			4. Setiap basis data dapat menangani aplikasi-aplikasi secara otonom.
			5. Masing-masing DBMS berpartisipasi dalam sedikitnya satu aplikasi.

**2.2.5. Entity Relationship Modelling (ER Modelling)**

Model E-R adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas, atribut, dan hubungan antar entitas, kadir (2009:30). Model ini dinyatakan dalam bentuk diagram, itulah sebabnya model E-R sering juga disebut sebagai diagram E-R. Dan perlu diketahui bahwa model seperti ini tidak mencerminkan bentuk fisik yang nantinya akan disimpan dalam database, melainkan hanya bersifat konseptual. *Entity Relationship* Model adalah suatu teknik dasar terpenting dalam mendesain suatu basis data.

 **Gambar 2.1 Simbol-simbol ER-Diagram**

Komponen-komponen yang ada di dalam *Entity Relationship model* :

a. Entity, ialah sesuatu yang bisa dibedakan dalam dunia nyata dimana informasi yang berkaitan dengannya dikumpulkan.

b. Atribut, adalah karakteristik dari entity atau relationship, yang menyediakan penjelasan detail tentang entity atau relationship tersebut. Nilai Atribut merupakan suatu data aktual atau informasi yang disimpan pada suatu atribut di dalam suatu entity atau relationship.

**2.2.6. Inventory (Persediaan barang)**

Istilah persedian barang (*Inventori*) suatu istilah umum yang menunjukan segala suatu atau sumber daya – sumber daya organisasi yang disimpan dalam antisipasinya terhadap pemenuhan permintaan. Adapun pengertian Inventori barang dagangan adalah untuk menunjukkan adanya barang-barang yang tujuannya untuk dijual kembali melalui kegiatan perusahaan yang normal serta barang yang masih dalam proses produksi (*Work in process*), Nugroho (2004:45)

**2.2.7. MySql**

MySQL merupakan software RDBMS (atau server satabase) yang dapat mengelolah database dengan cepat, dapat menampung jumlah data yang sangat besar, dapat diakses oleh banyak user dan dapat melakukan suatu proses secara sinkron atau berbarengan, Raharjo (2011:21). saat ini MySQL banyak digunakan di berbagai kalangan untuk melakukan penyimpanan dan pengolahan data, mulai dari kalangan akademis sampai ke industry, baik industri kecil, menengah, ataupun besar.

**2.2.8. *Strored Procedure/function***

Prosedur dan fungsi merupakan objek *database*  yang berisi runtunan statemen atau perintah yang dibuat untuk memenuhi kebutuhan-kebutuhan khusus tertentu. Sekali dibuat, prosedur dan fungsi dapat digunakan secara berulang, Raharjo (2011:187).

**2.2.9. *Trigger***

*Trigger* adalah objek di dalam database yang berasosiasi dengan suatu table. *Trigger* akan diaktivasi ketika table tersebut dikenai *event* tertentu. *Event*  yang dimaksud disini adalah suatu kejadian yang menimpa table, bisa berupa penambahan, perubahan, maupun penghapusan data, Raharjo (2011:211).

*Trigger* adalah sebuat objek yang bekerja bersama sebuah table dan digunakan untuk memeriksa sebuah operasi atau kejadian tertentu pada sebuah table. Objek ini akan diaktifkan atau dijalankan oleh *MySQL* apabila terjadi operasi atau kejadian tertentu pada sebuah table, Priyanto (2007:87).

**2.2.10. *View***

*View*  adalah objek di dalam *database* yang berisi kumpulan kolom yang dihasilkan dari perintah SELECT. Dengan kata lain yang lebih sederhana, *view* adalah objek yang menyimpan hasil *query,*baik dari satu table atau lebih, Raharjo (2011:139).

*View* adalah sebuah objek mirip seperti table yang data-datanya diperoleh dari hasil perintah *query* SELECTterhadap sebuah table atau beberapa table. Data-data tersebut tetap berada pada table aslinnya, view tidak menyimpan data-data apapun namun data-data view dihasilkan dari output Namun data view dihasilkan dari hasil output perintah SELECT terhadap tabel aslinya, Priyanto (2007:93).

**2.3. Penelitian Sebelumnya**

**Tabel 2.1** Penelitian Sebelumnya

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Peneliti** | **Judul** | **Metode** | **Hasil** |
| 1. | Swandy, David dan Adi Putra, 2012 | Perancangan Sistem Basis DataOperasional Pada Smp StradaSanto Fransiskus Xaverius 2 | *Analysis dan Perancangan Pendekatan Connolly and Begg*   | Sistem untuk mengolah data-data operasional sekolah, khususnya untuk mengelola data penerimaan siswa baru. |
| 2. | Yanti Efendy, 2012 | Analisis Perancangan Basis Data Klaim Asuransi KerugianPT. Asuransi Bina Dana Arta TBK | *Analysis dan Perancangan Pendekatan Connolly and Begg* | Sistem untuk mengklaim asuransi kerugian penjualan saham. |

1. Dari hasil pengembangan basis data yang terdapat didalam jurnal Perancangan Sistem Basis Data Operasional Pada Smp StradaSanto Fransiskus Xaverius 2, yang ditulis oleh Swandy, David, dan Adi Putra dapat ditarik kesimpulan yaitu penelitian ini sudah menghasilkan basis data operasional pada Smp Strada Santo Fransiskus Xaverius 2 yang digunakan untuk mengolah data penerimaan siswa baru. Perancangan basis data yang telah dilakukan menghasilkan tabel-tabel yang menjadi tempat pengolahan data sekolah, seperti tabel : Siswa, OrangTua, Karyawan, Jabatan, PenerimaanSiswaBaru, PembayaranUangPangkal, PembayaranSPP, Absen, Kelas, MataPelajaran dan lain-lain. Sehingga data yang dicari mudah didapatkan secara cepat dan tepat.
2. Dari hasil pengembangan basis data yang terdapat didalam jurnal Analisis Perancangan Basis Data Klaim Asuransi Kerugian PT. Asuransi Bina Dana Arta TBK, yang ditulis oleh Yanti Efendy dapat ditarik kesimpulan sistem basis data yang baru ini dapat mengefisienkan waktu dimana pengelolaan basis datanya terintegrasi sehingga dalam pemrosesan klaim asuransi kerugian menjadi lebih cepat dan akurat.

**BAB III**

**ANALISIS DAN PERANCANGAN**

1. **Analisis dan Pengumpulan Kebutuhan Basis Data**

Tahapan ini merupakan langkah awal untuk mengumpulkan data-data yang diperlukan dalam proses pembuatan basis data inventori yaitu dengan melakukan analisis terhadap sistem yang sedang berjalan.

* 1. **Analisis Sistem yang Berjalan**

Pada tahapan ini penulis mengumpulkan data dengan melakukan wawancara kepada bagian gudang, pelanggan dan pihak-pihak yang terkait. Selain itu juga penulis juga melakukan studi pustaka dengan mengumpulkan berkas-berkas yang digunakan dalam proses inventori barang, kemudian hasil tersebut kita tuangkan dalam bentuk *flowchart* dokumen. Aliran dokumen Inventori RKU Komputer dapat digambarkan *flowchart* dokumen sebagai berikut:

1. Bagian gudang melakukan pengecekan barang-barang yang ada digudang, jika cukup bagian gudang akan membuat laporan stok barang, apabila tidak cukup bagian gudang akan membuat surat permintaan barang dan nantinya akan dibuatkan surat pembelian barang kepada suplier.
2. Setelah surat pembelian barang sampai, suplier akan melakukan pengecekan barang yang dibeli cukup atau tidak, Jika barang yang dibeli cukup, maka suplier akan mengirim barang-barang tersebut, Jika tidak maka suplier akan melakukan konfirmasi kebagian gudang RKU.
3. Bagian gudang melakukan pengecekan kondisi barang-barang yang dibeli, jika baik bagian gudang akan membuat laporan penambahan barang, apabila barang yang dikirim tidak dalam kodisi baik, maka bagian gudang akan melakukan pembutan surat retur pembelian kepada suplier.

**Gambar 3.1** Alur Proses dokumen sistem yang berjalan

1. ***Database Planning* (Perencanaan Basis Data)**

**3.2.1 Pernyataan Misi**

Perancangan basis data dimaksudkan untuk membuatkan struktur basis data yang mempunyai relasi antar tabel kemudian dilengkapi dengan berbagai fasilitas yang dapat mempermudah tahap pengembangan aplikasi inventori barang.

* + 1. **Pernyataan Tujuan**

Pernyataan tujuan dalam perancangan basis data ini dapat dijabarkan dalam beberapa kebutuhan akan informasi yang ada. Adapun kebutuhan akan proses basis data antara lain untuk mengelola (memasukkan, memperbaruhi dan menghapus) data-data. Selain itu sistem dapat melakukan pencarian data dan pembuatan laporan-laporan yang diperlukan dalam hubungannya dengan kegiatan inventori.

* + 1. **Permasalahan yang Dihadapi**

 Dari analisis sistem yang berjalan maka dapat dilihat permasalahan yang terjadi saat ini. Adapun uraian permasalahan dan rencana usulan pemecahan masalah dapat dilihat dalam tabel 3.2.

**Tabel 3.1** Masalah dan Usulan Pemecahan Masalah

|  |  |
| --- | --- |
| **Masalah** | **Usulan Pemecahan Masalah** |
| Dalam proses inventori barang, pembuatan surat-surat yang dilakukan secara manual menyebabkan data yang berhubungan akan sulit dilacak apbila terjadi perubahan.  | Merancang sistem basis data yang terintegrasi antara tabel satu dan lainnya, kemudian dibuatkan *Stored Procedure /Function, Trigger,* dan *View*. sehingga apabila terjadi perubahan pada tabel tertentu maka akan secara otomatis mempengaruhi tabel yang berhubungan. |
| Terjadinya redudansi pada proses permintaan dan pembelian barang dikarenakan terjadinya penulisan ulang data-data barang pada pencatatan data pembelian. | Merancang sistem basis data untuk mengurangi redudansi data dan duplikasi data saat melakukan manipulasi data karena sistem basis data yang saling terhubung. |
| Kesulitan mendapatkan data yang dibutuhkan. | Dengan adanya basis data, maka proses pencarian data dapat dilakukan dengan sangat cepat sehingga informasi dapat diterima langung saat dibutuhkan. |
| Banyaknya berkas yang dikelola membuat penyimpanan data menjadi kurang efisien karena membutuhkan tempat yang cukup besar untuk menyimpan data-data tersebut. | Dengan adanya basis data, maka proses penyimpanan dapat dilakukan dengan efisien dan untuk proses *backup*  data lebih mudah karena bisa disimpan di dalam *flashdisk, CD, DVD* dan media penyimpanan lainnya. |

1. **Desain Basis Data**
	* 1. **Perancangan Basis Data Konseptual**

Perancangan basis data secara konseptual yaitu proses membangun suatu model informasi yang digunakan dalam suatu perusahaan, bebas dari semua pertimbangan fisik. Pada rancangan sistem akan dijelaskan gambaran secara umum sistem basis data yang akan diimplementasikan dengan model desain seperti berikut.

**Gambar 3.2 *Entity Relational Diagram***

* + - 1. **Identifikasi Tipe Entitas**

Pada tahap ini penulis menentukan dan mengidentifikasi tipe entiti yang akan digunakan. Berikut adalah tabel berserta penjelasan dari hasil identifikasi yang penulis lakukan :

**Tabel 3.2** Tipe Entitas

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | **Nama Entitas** | **Atribut** | **Deskripsi** | **Tipe Data dan Ukuran** | **Multi Value** | **Null** |
| 1 | Barang | Kd\_barang | Nomor induk Barang | Varchar (15) | No | No |
| Nama\_barang | Nama Barang | Varchar (50) | No | No |
| Harga\_beli | Harga beli | Int (10) | No | No |
| Harga\_jual | Harga jual | Int (10) | No | No |
| Stok | Keseluruhan stok | Int (3) | No | No |
| Nama\_suplier | Nama suplier | Varchar (50) | No  | No |
| 2 | Pembelian | No\_pembelian | Nomor pembelian | Varchar (15) | No | No |
| Tanggal\_masuk | Tanggal\_barang masuk | Date() | No | No |
| No\_nota | Nomor nota pembelian | Varchar (200) | No | No |
| Kd\_barang | Kode barang | Varchar (15) | No | No |
| Jumlah | Jumlah barang | Int (5) | No | No |
| Harga\_satuan | Harga satuan | Int (10) | No | No |
| Total\_biaya | Total harga beli | Int (10) | No | No |
| Kd\_karyawan | Kode karyawan gudang | Varchar (15) | No | No |
| 3 | Retur pembelian | No\_retur | Nomor Retur pembelian  | Varchar (5) | No | No |
| Tgl\_retur | Tanggal barang diretur | Date() | No | No |
| Jumlah\_retur | Jumlah barang yang di retur | Int (5) | No | No |
| Kondisi | Keterangan kondisi barang | Varchar (2) | No | No |
| No\_pembelian | Nomor pembelian | Varchar (15) | No | No |
| 4 | Bagian gudang | Kd\_karyawan | Kode karyawan | Varchar (15) | No | No |
| Nama\_karyawan | Nama karyawan | Vaechar (50) | No | No |
| Alamat\_karyawan | Alamat karyawan | Varchar (100) | No | No |
| No\_hp | Nomor hp | Varchar (15) | No | No |

* + - 1. **Pengidentifikasian Relasi**

Tahap ini bertujuan untuk mengidentifikasi relasi penting yang terdapat diantara entity yang telah di identifikasi. Tipe relasi dapat di dentifikasikan melalui tahapan membuat ER diagram, menentukan multiplicity dari tipe relational. Gambar 3.1. dan tabel 3.3. merupakan hasil identifikasi relasi yang penulis lakukan.

**Tabel 3.3** Relasi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Nama Entitas** | **Multiplicity** | **Relationship** | **Nama Entitas** | **Multiplicity** |
| 1 | Barang | 1..\* | Mempunyai | pembelian | 1..\* |
| 2 | Pembelian | 1..1 | Mempunyai | Retur pembelian | 1..1 |
|  |  | \*..1 | dilakukan | Bagian gudang | \*..1 |
| 3 | Retur pembelian | 1..1 | Mempunyai | Pembelian | 1..1 |
| 4 | Bagian gudang | 1..\* | Melakukan | Pembelian | 1..\* |

**Gambar 3.3 ERD Konseptual**

* + - 1. **Identifikasi dan Asosiasi Atribut Suatu Entitas**

Setelah melakukan identifikasi relasi, selanjutnya penulis melakukan identifikasi atribut dari tipe identitas, tujuan dilakukannya identifikasi ini adalah untuk menghubungkan atribut dengan eintity atau relationship yang tepat. Berikut hasil identifikasi yang penulis lakukan :

**Tabel 3.4** Asosiasi Atribut

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | **Nama Entitas** | **Atribut** | **Deskripsi** | **Tipe Data dan Ukuran** | **Multi Value** | **Null** |
| 1 | Barang | Kd\_barang | Nomor induk Barang | Varchar (15) | No | No |
| Nama\_barang | Nama Barang | Varchar (50) | No | No |
| Harga\_beli | Harga beli | Int (10) | No | No |
| Harga\_jual | Harga jual | Int (10) | No | No |
| Stok | Keseluruhan stok | Int (3) | No | No |
| Nama\_suplier | Nama suplier | Varchar (50) | No  | No |
| 2 | Pembelian | No\_pembelian | Nomor pembelian | Varchar (15) | No | No |
| Tanggal\_masuk | Tanggal\_barang masuk | Date() | No | No |
| No\_nota | Nomor nota pembelian | Varchar (200) | No | No |
| Kd\_barang | Kode barang | Varchar (15) | No | No |
| Jumlah | Jumlah barang | Int (5) | No | No |
| Harga\_satuan | Harga satuan | Int (10) | No | No |
| Total\_biaya | Total harga beli | Int (10) | No | No |
| Kd\_karyawan | Kode karyawan gudang | Varchar (15) | No | No |
| 3 | Retur pembelian | No\_retur | Nomor Retur pembelian  | Varchar (5) | No | No |
| Tgl\_retur | Tanggal barang diretur | Date() | No | No |
| Jumlah\_retur | Jumlah barang yang di retur | Int (5) | No | No |
| Kondisi | Keterangan kondisi barang | Varchar (2) | No | No |
| No\_pembelian | Nomor pembelian | Varchar (15) | No | No |
| 4 | Bagian gudang | Kd\_karyawan | Kode karyawan | Varchar (15) | No | No |
| Nama\_karyawan | Nama karyawan | Vaechar (50) | No | No |
| Alamat\_karyawan | Alamat karyawan | Varchar (100) | No | No |
| No\_hp | Nomor hp | Varchar (15) | No | No |

* + - 1. **Identifikasi domain atribut**

Tujuan dari langkah ini adalah untuk menentukan domain atribut pada model data konseptual. Berikut ini adalah domain atribut dari masing-masing atribut yang telah diidentifikasikan sebelumnya.

**Tabel 3.5** Domain Atribut

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Entitas** | **Atribut** | **Tipe Data dan Ukuran** | **Domain atribut** |
| 1 | Barang | Kd\_barang | Varchar (15) | *Range value* A-Z, az,0-9 |
| Nama\_barang | Varchar (50) | *Range value* A-Z, az,0-9 |
| Harga\_beli | Int (10) | Range value1-999999 |
| Harga\_jual | Int (10) | Range value1-999999 |
| Stok | Int (3) | Range value1-999999 |
| Nama\_suplier | Varchar (50) | *Range value* A-Z, az,0-9 |
| 2 | Pembelian | No\_pembelian | Varchar (15) | *Range value* A-Z, az,0-9 |
| Tanggal\_masuk | Date() | *Range value* A-Z, az,0-9 |
| No\_nota | Varchar (200) | *Range value* A-Z, az,0-9 |
| Kd\_barang | Varchar (15) | Range value A-Z, az, |
| Jumlah | Int (5) | Range value A-Z, az, |
| Harga\_satuan | Int (10) | Range value1-999999 |
| Total\_biaya | Int (10) | Range value1-999999 |
| Kd\_karyawan | Varchar (15) | *Range value* A-Z, az,0-9 |
| 3 | Retur pembelian | No\_retur | Varchar (5) | Range value A-Z, az, |
| Tgl\_retur | Date() | Format yyyy/mm/dd |
| Jumlah\_retur | Int (5) | Range value1-999999 |
| Kondisi | Varchar (2) | *Range value* A-Z, az,0-9 |
| No\_pembelian | Varchar (15) | *Range value* A-Z, az,0-9 |
| 4 | Bagian gudang | Kd\_karyawan | Varchar (15) | Range value A-Z, az, |
| Nama\_karyawan | Vaechar (50) | Range value A-Z, az, |
| Alamat\_karyawan | Varchar (100) | Range value A-Z, az, |

* + - 1. **Menentukan *Primary key* dan *candidate key***

Pada tahapan ini penulis menentukan *field* manakah dari tiap-tiap tabel yang akan dijadikan *primary key* dan *candidate key*. Berikut adalah *primary key* dan *candidate key* dari tiap-tiap tabel entitas :

**Tabel 3.6** *Primary key* dan *candidate key*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Entitas | *Candidate key* | *Primary key* |
| 1 | Barang | Kd\_barang | Kd\_barang |
| 2 | Pembelian | No\_pembelian | No\_pembelian |
| 3 | Retur pembelian | No\_retur | No\_retur |
| 4 | Bagian gudang | Kd\_karyawan | Kd\_karyawan |

* + 1. **Perancangan basis data logical**

Perancangan basis data logikal menjelaskan tentang proses untuk mengahasilkan gambaran dari implementasi basis data pada tempat penyimpanan, menjelaskan dasar relasi, organisasi file dan indeks.

* + - 1. **Menentukan model logika data**

Tahap ini bertujuan untuk membuat relasi untuk model data logikal lokal untuk menggambarkan entiti, relasi, dan atribut-atribut yang diidentifikasi.

1. **Relasi Barang dengan pembelian**

**Mempunyai**

**Barang**(kd\_barang,nama\_barang,harga\_beli,harga\_jual,stok,nama\_suplier)

**Primary key** : kd\_barang

**pembelian**(no\_pembelian,tgl\_masuk,no\_nota,kd\_barang,jumlah,harga\_satuan,total\_biaya,kd\_karyawan)

**Primary key** : no\_pembelian

**References Barang (kd\_barang)**

1. **Relasi Pembelian dengan Retur pembelian**

**pembelian**(no\_pembelian,tgl\_masuk,no\_nota,kd\_barang,jumlah,harga\_satuan,total\_biaya,kd\_karyawan)

**Primary key** : no\_pembelian

**Retur\_pembelian**(no\_retur,tgl\_retur,kondisi,jumlah\_retur,no\_pembelian)

**Primary key** : no\_retur

**References pembelian(no\_pembelian)**

**Mempunyai**

1. **Relasi Bagian gudang dengan pembelian**

**Melakukan**

**Bagian\_gudang**(kd\_karyawan,nama\_karyawan,alamat,no\_hp)

**Primary key** : kd\_karyawan

**pembelian**(no\_pembelian,tgl\_masuk,no\_nota,kd\_barang,jumlah,harga\_satuan,total\_biaya,kd\_karyawan)

**Primary key** : no\_pembelian

**References bagian\_gudang(kd\_karyawan)**

* + - 1. **Normalisasi**

Untuk mendapatkan kriteria-kriteria normalisasi, semua tabel di rekontruksi menjadi satu tabel (*Universal Table*). Lalu diterapkan kriteria-kriteria normalisasi hingga mendapatkan sejumlah tabel yang sudah normal. Dibawah ini merupakan tabel yang berada pada bentuk normal pertama :

1. Unormal,{kd\_barang,nama\_barang,harga\_beli,harga\_jual,stok,nama\_suplier,no\_pembelian,tgl\_masuk,no\_nota,kd\_barang,jumlah,harga\_satuan,total\_biaya,kd\_karyawan, no\_retur,tgl\_retur,kondisi,jumlah\_retur,no\_pembelian, kd\_karyawan,nama\_karyawan,alamat,no\_hp}
2. Bentuk Normal Pertama, Langkah berikutnya adalah dengan cara memisahkan atribut-atribut yang nilainya sama akan ditulis hanya satu kali. {kd\_barang,nama\_barang,harga\_beli,harga\_jual,stok,nama\_suplier,no\_pembelian,tgl\_masuk,no\_nota,kd\_barang,jumlah,harga\_satuan,total\_biaya,kd\_karyawan,no\_retur,tgl\_retur,kondisi,jumlah\_retur,no\_pembelian,kd\_karyawan,nama\_karyawan,alamat,no\_hp}
3. Bentuk Normal Kedua, Langkah selanjutnya adalah dengan cara menentukan ketergantungan fungsional.

**Tabel Barang**

{ \*kd\_barang,nama\_barang, kd\_suplier }

**Tabel Stok**

{\*kd\_stok,stok,kd\_barang}

**Tabel Suplier**

{\*kd\_suplier,nama\_suplier,no\_hp)

**Tabel harga beli**

{\*kd\_hrg\_beli,harga\_beli,kd\_barang}

**Tabel harga jual**

{\*kd\_hrg\_jual,harga\_jual,kd\_barang}

**Tabel Pembelian**

{\*no\_pembelian,tgl\_masuk,no\_nota,kd\_barang,jumlah,harga\_satuan,total\_biaya,kd\_karyawan}

**Tabel Retur pembelian**

{\*no\_retur,tgl\_retur,kondisi,jumlah\_retur,no\_pembelian}

**Tabel bagian gudang**

{\* kd\_karyawan,nama\_karyawan,alamat,no\_hp }

1. Bentuk Normal Ketiga, berdasarkan bentuk normal Kedua (2NF) di atas masih ditemukan informasi tabel yang kurang akurat, oleh karena itu dibutuhkan suatu relasi antar tabel yang masing-masing atribut utama bergantung fungsional penuh pada masing-masing kunci dimana kunci tersebut bukan bagiannya.

**Tabel Barang**

{ \*kd\_barang,nama\_barang, \*\*kd\_suplier }

**Tabel Stok**

{\*kd\_stok,stok,\*\*kd\_barang}

**Tabel Suplier**

{\*kd\_suplier,nama\_suplier,no\_hp)

**Tabel harga beli**

{\*kd\_hrg\_beli,harga\_beli,\*\*kd\_barang}

**Tabel harga jual**

{\*kd\_hrg\_jual,harga\_jual,\*\*kd\_barang}

**Tabel Pembelian**

{\*no\_pembelian,tgl\_masuk,no\_nota,kd\_barang,jumlah,harga\_satuan,total\_biaya,\*\*kd\_karyawan}

**Tabel Retur pembelian**

{\*no\_retur,tgl\_retur,kondisi,jumlah\_retur,no\_pembelian}

**Tabel bagian gudang**

{\* kd\_karyawan,nama\_karyawan,alamat,no\_hp }

* + 1. **Perancangan basis data fisikal**

Perancangan basis data secara fisik merupakan tahapan untuk mengimplementasikan hasil perancangan basis data secara logis menjadi tersimpan secara fisik pada media penyimpanan eksternal sesuai dengan DBMS yang digunakan. Dapat disimpulkan bahwa proses perancangan fisik merupakan transformasi dari perancangan logis terhadap jenis DBMS yang digunakan sehingga dapat disimpan secara fisik pada media penyimpanan.

* + - 1. **Struktur tabel hasil normalisasi**
1. Nama tabel : Barang

PK : Kd\_barang

FK : kd\_suplier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama *field* | Tipe | Panjang | Keterangan |
| 1 | kd\_barang | Varchar  | (15) | Kode barang |
| 2 | nama\_barang | Varchar  | (50) | Nama Barang |
| 3 | kd\_suplier | Varchar  | (15) | Kode suplier |

1. Nama tabel : Stok

PK : kd\_stok

FK : kd\_barang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama *field* | Tipe | Panjang | Keterangan |
| 1 | kd\_stok | Varchar  | (15) | Kode stok |
| 2 | stok | int | (10) | Jumlah stok barang |
| 3 | kd\_barang | Varchar  | (15) | Kode barang |

1. Nama tabel : Suplier

 PK : kd\_suplier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama *field* | Tipe | Panjang | Keterangan |
| 1 | kd\_suplier | Varchar  | (15) | Kode suplier |
| 2 | nama\_suplier | Varchar  | (50) | Nama suplier |
| 3 | no\_hp | Varchar  | (15) | Nomor hp suplier |

1. Nama tabel : harga beli

PK : kd\_hrg\_beli

FK : kd\_barang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama *field* | Tipe | Panjang | Keterangan |
| 1 | kd\_hrg\_beli  | Varchar  | (15) | Kode harga beli |
| 2 | harga\_beli | Int  | (10) | Harga beli |
| 3 | kd\_barang | Varchar  | (15) | Kode barang |

1. Nama tabel : harga jual

PK : kd\_hrg\_jual

FK : kd\_barang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama *field* | Tipe | Panjang | Keterangan |
| 1 | kd\_hrg\_jual | Varchar  | (15) | Kode harga jual |
| 2 | harga\_jual | Int  | (10) | Harga jual |
| 3 | kd\_barang | Varchar  | (15) | Kode barang |

1. Nama tabel : Pembelian

PK : no\_pembelian

FK : kd\_barang

FK : kd\_karyawan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama *field* | Tipe | Panjang | Keterangan |
| 1 | no\_pembelian | int | (3) | Nomor pembelian |
| 2 | tgl\_masuk | Date | () | Tanggal peminjaman bagian gudang |
| 3 | no\_nota | Varchar  | (5) | Nomor nota |
| 4 | kd\_barang | Varchar  | (15) | Kode barang |
| 5 | jumlah | int  | (3) | Jumlah barang |
| 6 | harga\_satuan | Int | (10) | Harga satuan |
| 7 | total\_biaya | Int | (10) | Total biaya keseluruhan |
| 8 | kd\_karyawan | Kd\_barng | (15) | Kode barang |

1. Nama tabel : retur pembelian

PK : no\_retur

FK : no\_pembelian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama *field* | Tipe | Panjang | Keterangan |
| 1 | no\_retur | Int | (3) | Nomor retur |
| 2 | tgl\_retur | Date | () | Tanggal retur pengembalian barang |
| 3 | kondisi | Varchar  | (100) | Kondisi barang |
| 4 | jumlah\_retur | Int | (3) | Jumlah barang yang dikembalikan |
| 5 | no\_pembelian | Int | (3) | Nomor pembelian |

1. Nama tabel : bagian gudang

PK : kd\_karyawan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama *field* | Tipe | Panjang | Keterangan |
| 1 | kd\_karyawan | Varchar  | (5) | Kode karyawan bagian gudang |
| 2 | nama\_karyawan | Varchar  | (50) | Nama karyawan |
| 3 | alamat  | Varchar  | (100) | alamat |
| 4 | no\_hp | Varhar | (16) | Nomor hp |

**Gambar 3.4 ERD FISICAL**

**BAB IV**

**HASIL DAN PEMBAHASAN**

**4.1. Hasil**

Setelah melewati proses fase-fase perancangan konseptual, logical dan fisikal yang telah diuraikan di bab sebelumnya, maka diperoleh suatu racangan basis data yang siap untuk dikembangkan menjadi aplikasi baik berupa *desktop* ataupun *website,* disesuaikan dengan kebutuhan CV. RKU Komputer. Hasil dari perancangan basis data ini terdiri dari tujuh tabel yang masing-masing tabel meniliki beberapa atribut yang mewakili beberapa item data.

Adapun tabel-tabel dalam basis data tersebut adalah sebagai berikut :

1. Tabel pasien : bagian\_gudang.myd, bagian\_gudang.frm dan bagian\_gudang.myi
2. Tabel barang : barang.myd, barang.frm dan barang.myi
3. Tabel Harga\_beli : harga\_beli.myd, harga\_beli.frm, harga\_beli.myi
4. Tabel Harga\_jual : harga\_jual.myd, harga\_jual.frm, harga\_jual.myi
5. Tabel Pembelian : pembelian.myd, pembelian.frn, pembelian.myi
6. Tabel Retur\_pembelian : retur\_pembelian.myd, retur\_pembelian.frn, retur\_pembelian.myi
7. Tabel Stok : stok.myd, stok.frm, stok.myi
8. Tabel Suplier : suplier.myd, suplier.frm, suplier.myi

 Hasil perancangan basis data disimpan kedalam basis data MySQL dengan nama DB\_RKU.

**Gambar 4.1.** Hasil Perancangan Basis Data dengan MySQL

**4.2. Pembahasan**

**4.2.1. Pembuatan Basis Data dengan MySQL**

Dalam pembuatan basis data kita akan melakukan langkah-langkah sebagai berikut. Langkah pertama menjalankan aplikasi MySQL melalui *broswer* dengan mengetik *localhost/phpmyadmin* pada *address*.

**Gambar 4.2.** Tampilan awal MySQL

 Selanjutnya untuk membuat basis data pada kolom Ciptakan *database* baru dimasukkan nama basis data “DB\_RKU” selanjutnya pilih Ciptakan. Langkah selanjutnya membuat tabel dengan memasukkan melalui *interface* PhpMyAdmin atau langsung melalui perintah *Query*.

**Gambar 4.3**. Tampilan membuat table

**4.2.2. Pembahasan *Query* Basis Data**

 Pada bagian ini akan di jelaskan pembuatan tabel dalam bentuk *query*. Adapun hasil rancangan *Database* dalam sistem basis data ini adalah sebagai berikut :

1. **Tabel Barang,** merupakan tabel yang berfungsi untuk menyimpan data barang puskesmas. Dimana dalam tabel barang terdiri dari 3 (tiga) *field* dengan kode barang sebagai *primary key.*

--

-- Table structure for table `barang`

--

CREATE TABLE `barang` (

 `kd\_barang` varchar(15) NOT NULL,

 `nama\_barang` varchar(50) NOT NULL,

 `kd\_suplier` varchar(15) NOT NULL,

 PRIMARY KEY (`kd\_barang`)

) TYPE=MyISAM;

**Gambar 4.4.** *Query* Tabel Barang

Berikut adalah struktrur dari tabel barang yang akan diciptakan setelah *query* pada gambar 4.4 dijalankan.

**Tabel 4.1** Tabel Entitas Barang

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Atribut** | **Deskripsi** | **Tipe Data dan Panjang** | **Null** | **Multivalue** |
| 1 | kd\_barang | Kode barang | Varchar (15) | No | No |
| 2 | nama\_barang | Nama Barang | Varchar (50) | No | No |
| 3 | kd\_suplier | Kode suplier | Varchar (15) | No | No |

 Setelah dijalankan perintah *query* maka didapat sebuah tabel barang seperti pada tampilan gambar 4.5 dibawah ini.

**Gambar 4.5**. Tampilan Tabel Barang

**2. Tabel Stok,** merupakan tabel yang berfungsi untuk menyimpan data stok. Dimana dalam tabel stok terdiri dari 3 (tiga) *field* dengan kd\_daftar sebagai *primary key.*

--

-- Table structure for table `stok`

--

CREATE TABLE `stok` (

 `kd\_stok` varchar(15) NOT NULL,

 `stok` int(11) NOT NULL,

 `kd\_barang` varchar(15) NOT NULL,

 PRIMARY KEY (`kd\_stok`)

) TYPE=MyISAM;

**Gambar 4.6.** *Query* Tabel Stok

Berikut adalah struktrur dari tabel stok yang akan diciptakan setelah *query* pada gambar 4.6 dijalankan.

**Tabel 4.2** Tabel Stok

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Atribut** | **Deskripsi** | **Tipe Data dan Panjang** | **Null** | **Multivalue** |
| 1 | Kd\_stok | Kode stok | Varchar (15) | No | No |
| 2 | Stok | Stok barang | Int(11) | No | No |
| 3 | Kd\_barang | Kode barang | Varchar (15) | No | No |

Setelah dijalankan perintah query maka didapat sebuah tabel stok seperti pada tampilan gambar 4.7 dibawah ini.

**Gambar 4.7**. Tampilan Tabel Stok

**3. Tabel Suplier,** merupakan tabel yang berfungsi untuk menyimpan data suplier. Dimana dalam tabel suplier terdiri dari 3 (tiga) *field* dengan kd\_suplier sebagai *primary key.*

--

-- Table structure for table `suplier`

--

CREATE TABLE `suplier` (

 `kd\_suplier` varchar(15) NOT NULL,

 `nama\_suplier` varchar(50) NOT NULL,

 `no\_hp` varchar(15) NOT NULL,

 PRIMARY KEY (`kd\_suplier`)

) TYPE=MyISAM;

**Gambar 4.8.** *Query* Tabel Suplier

Berikut adalah struktrur dari tabel suplier yang akan diciptakan setelah *query* pada gambar 4.8 dijalankan.

**Tabel 4.3** Tabelsuplier

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Atribut** | **Deskripsi** | **Tipe Data dan Panjang** | **Null** | **Multivalue** |
| 1 | kd\_suplier | Kode suplier | Varchar (15) | No | No |
| 2 | Nama\_suplier | Nama suplier | Varchar (50) | No | No |
| 3 | No\_hp | Nomor hp | Varchar (15) | No | No |

 Setelah dijalankan perintah query maka didapat sebuah tabel suplier seperti pada tampilan gambar 4.9 dibawah ini.

**Gambar 4.9**. Tampilan Tabel Suplier

**4. Tabel Harga beli,** merupakan tabel yang berfungsi untuk menyimpan data harga beli. Dimana dalam tabel harga beli ini terdiri dari 3 (tiga) *field* dengan kd\_hrg\_beli sebagai *primary key.*

--

-- Table structure for table `harga\_beli`

--

CREATE TABLE `harga\_beli` (

 `kd\_hrg\_beli` int(11) NOT NULL,

 `harga\_beli` int(11) NOT NULL,

 `kd\_barang` varchar(15) NOT NULL,

 PRIMARY KEY (`kd\_hrg\_beli`)

) TYPE=MyISAM;

**Gambar 4.10** *Query* Tabel Harga beli

Berikut adalah struktrur dari tabel harga beli yang akan diciptakan setelah *query* pada gambar 4.10 dijalankan.

**Tabel 4.4** Tabel Harga beli

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Atribut** | **Deskripsi** | **Tipe Data dan Panjang** | **Null** | **Multivalue** |
| 1 | Kd\_hrg\_beli | Kode harga beli | Int(11) | No | No |
| 2 | Harga\_beli | Harga beli | Int(11) | No | No |
| 3 | Kd\_barang | Kode barang | Varchar (15) | No | No |

 Setelah dijalankan perintah query maka didapat sebuah tabel harga beli seperti pada tampilan gambar 4.11 di bawah ini.

**Gambar 4.11.** Tampilan Tabel Harga beli

**5. Tabel Harga jual,** merupakan tabel yang berfungsi untuk menyimpan data harga jual. Dimana dalam tabel harga jual terdiri dari 3 (tiga) *field* dengan id\_harga jual sebagai *primary key.*

--

-- Table structure for table `harga\_jual`

--

CREATE TABLE `harga\_jual` (

 `kd\_hrg\_jual` int(11) NOT NULL,

 `harga\_jual` int(11) NOT NULL,

 `kd\_barang` varchar(15) NOT NULL,

 PRIMARY KEY (`kd\_hrg\_jual`)

) TYPE=MyISAM;

**Gambar 4.12** *Query* Tabel Harga jual

Berikut adalah struktrur dari tabel harga jual yang akan diciptakan setelah *query* pada gambar 4.12 dijalankan.

**Tabel 4.5** Tabel Harga jual

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Atribut** | **Deskripsi** | **Tipe Data dan Panjang** | **Null** | **Multivalue** |
| 1 | Id\_hrg\_jual | Id harga jual | Int(3) | No | No |
| 2 | Harga\_jual | Harga jual | Int(3) | No | No |
| 3 | Kd\_barang | Kode barang | Varchar (15) | No | No |

 Setelah dijalankan perintah query maka didapat sebuah tabel harga jual seperti pada tampilan gambar 4.13 dibawah ini.

 **Gambar 4.13.** Tampilan Tabel Harga jual

**6. Tabel Pembelian,** merupakan tabel yang berfungsi untuk menyimpan data pembelian. Dimana dalam tabel pembelian terdiri dari 8 (delapan) *field* dengan kd\_pembelian sebagai *primary key.*

--

-- Table structure for table `pembelian`

--

CREATE TABLE `pembelian` (

 `no\_pembelian` int(11) NOT NULL auto\_increment,

 `tgl\_masuk` date NOT NULL,

 `no\_nota` varchar(5) NOT NULL,

 `kd\_barang` varchar(15) NOT NULL,

 `jumlah` int(11) NOT NULL,

 `kd\_hrg\_beli` int(11) NOT NULL,

 `total\_biaya` int(11) NOT NULL,

 `kd\_karyawan` varchar(15) NOT NULL,

 PRIMARY KEY (`no\_pembelian`)

) TYPE=MyISAM AUTO\_INCREMENT=1 ;

**Gambar 4.14** *Query* Tabel Pembelian

Berikut adalah struktrur dari tabel pembelian yang akan diciptakan setelah *query* pada gambar 4.14 dijalankan.

**Tabel 4.6** Tabel pembelian

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Atribut** | **Deskripsi** | **Tipe Data dan Panjang** | **Null** | **Multivalue** |
| 1 | no\_pembelian | Nomor pembelian | Int(11) | No | No |
| 2 | tgl\_masuk | Tanggal masuk | Date() | No | No |
| 3 | no\_nota | Nomor nota | Varchar (5) | No | No |
| 4 | kd\_barang | Kode barang | Varchar(15) | No | No |
| 5 | jumlah | Jumlah | Int(11) | No | No |
| 6 | Kd\_hrg\_beli | Kode harga beli | Int(11) | No | No |
| 7 | total\_biaya | Total biaya | Int(11) | No | No |
| 8 | kd\_karyawan | Kode karyawan | Varchar(15) | No | No |

Setelah dijalankan perintah query maka didapat sebuah tabel pembelian seperti pada tampilan gambar 4.15 dibawah ini.

**Gambar 4.15.** Tampilan Tabel Pembelian

**7. Tabel Retur Pembelian,** merupakan tabel yang berfungsi untuk menyimpan data retur pembelian. Dimana dalam tabel pembelian terdiri dari 4 (empat) *field* dengan id\_retur pembelian sebagai *primary key.*

--

-- Table structure for table `retur\_pembelian`

--

CREATE TABLE `retur\_pembelian` (

 `no\_retur` int(11) NOT NULL auto\_increment,

 `tgl\_retur` date NOT NULL,

 `kondisi` varchar(100) NOT NULL,

 `jumlah\_retur` int(11) NOT NULL,

 `no\_pembelian` int(11) NOT NULL,

 PRIMARY KEY (`no\_retur`)

) TYPE=MyISAM AUTO\_INCREMENT=1 ;

**Gambar 4.16** *Query* Tabel Retur pembelian

Berikut adalah struktrur dari tabel retur pembelian yang akan diciptakan setelah *query* pada gambar 4.16 dijalankan.

**Tabel 4.7** Tabel Retur pembelian

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Atribut** | **Deskripsi** | **Tipe Data dan Panjang** | **Null** | **Multivalue** |
| 1 | no\_retur | Nomor retur | Int(11) | No | No |
| 2 | tgl\_retur | Tanggal retur | Date() | No | No |
| 3 | kondisi | Kondisii | Varchar(100) | No | No |
| 4 | jumlah\_retur | Jumlah retur | Int(11) | No | No |
|  | no\_pembelian | Nomor pembelian | Int(11) | No | No |

 Setelah dijalankan perintah query maka didapat sebuah tabel retur pembelian seperti pada tampilan gambar 4.17 dibawah ini.

**Gambar 4.17.** Tampilan Tabel Retur pembelian

**8. Tabel Bagian gudang,** merupakan tabel yang berfungsi untuk menyimpan data bagian gudang. Dimana dalam tabel pembelian terdiri dari 4 (empat) *field* dengan id\_bagian gudang sebagai *primary key.*

--

-- Table structure for table `bagian\_gudang`

--

CREATE TABLE `bagian\_gudang` (

 `kd\_karyawan` varchar(5) NOT NULL,

 `nama\_karyawan` varchar(50) NOT NULL,

 `alamat` varchar(100) NOT NULL,

 `no\_hp` varchar(16) NOT NULL,

 PRIMARY KEY (`kd\_karyawan`)

) TYPE=MyISAM;

**Gambar 4.18** *Query* Tabel Bagian gudang

Berikut adalah struktrur dari tabel bagian gudang yang akan diciptakan setelah *query* pada gambar 4.18 dijalankan.

**Tabel 4.8** Tabel Bagian gudang

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Atribut** | **Deskripsi** | **Tipe Data dan Panjang** | **Null** | **Multivalue** |
| 1 | Kd\_karyawan | Kode karyawan | Varchar(5) | No | No |
| 2 | Nama\_karyawan | Nama karyawan | Varchar(50) | No | No |
| 3 | Alamat | Varchar | Varchar(100) | No | No |
| 4 | No\_hp\_karyawan | Nomor hp | Varchar(16) | No | No |

 Setelah dijalankan perintah query maka didapat sebuah tabel bagian gudang seperti pada tampilan gambar 4.19 dibawah ini.

**Gambar 4.19.** Tampilan Tabel bagian gudang

**4.2.3. *Trigger***

*Trigger* pada rancangan basis data ini berfungsi untuk melakukan perubahan table apabila tabel yang berkaitan terjadi *event* tertentu. Adapun *Trigger* yang dibuat untuk mengatasi perubahan tabel untuk *event-event* tertentu sebagai berikut :

1. **Trigger Auto\_update\_penambahan\_stok**

*Trigger* ini berfungsi mengupdate otomatis stok barang ketika terjadi pembelian barang Adapun *SQL* *syntax create Trigger* *Auto\_update\_penambahan\_stok* sebagai berikut :

DELIMITER $$

CREATE TRIGGER `auto\_update\_penambahan\_stok` AFTER INSERT ON `pembelian`

 FOR EACH ROW BEGIN

 UPDATE stok SET stok=stok+new.jumlah WHERE kd\_barang=new.kd\_barang;

 END;

$$ DELIMITER ;

**Gambar 4.20.** *Syntax create trigger* Auto\_update\_penambahan\_stok

Setelah membuat *Trigger* Auto\_update\_penambahan\_stok, skenarionya jika terjadi pembelian (*insert* pada tabel pembelian) maka secara otomatis tabel stok akan bertambah (*update* pada tabel stok).

Simulasi *Trigger Auto\_update\_penambahan\_stok* untuk proses pembelian:

1. Pada Gambar 4.21 terdapat kd\_barang A0001 dengan stok 15 dan nanti akan dilakukan penambahan otomatis sebanyak jumlah barang yang dibeli (*insert* pada tabel pembelian).



**Gambar 4.21.** Tabel stok sebelum *insert* pembelian

1. Terjadinya *insert* pada tabel pembelian yang dilakukan oleh bagian gudang dengan pembelian barang dengan kode A0001 sebanyak 7(tujuh) seperti gambar dibawah ini.



**Gambar 4.22.** Tabel pembelian ketika *insert*

1. Pada Gambar 4.23 terjadi penambahan sebanyak 7(tujuh) pada tabel pembelian dengan kode barang A0001 menjadi 22.

**Gambar 4.23.** Tabel stok setelah *insert* pembelian

1. **Trigger Auto\_update\_pengurangan\_stok**

*Trigger* ini berfungsi mengupdate otomatis stok barang ketika terjadi retur barang Adapun *SQL* *syntax create Trigger* *Auto\_update\_pengurangan\_stok* sebagai berikut :

DELIMITER $$

CREATE TRIGGER `db\_rku`.`auto\_update\_pengurangan\_stok`AFTER INSERT

 ON `db\_rku`.`retur\_pembelian`

 FOR EACH ROW BEGIN

 UPDATE stok SET stok=stok-new.jumlah\_retur WHERE kd\_barang=new.kd\_barang;

 END$$

DELIMITER ;

**Gambar 4.24.** *Syntax create trigger* Auto\_update\_pengurangan\_stok

Setelah membuat *Trigger* Auto\_update\_pengurangan\_stok, skenarionya jika terjadi retur barang (*insert* pada tabel retur barang) maka secara otomatis tabel stok akan berkurang (*update* pada tabel stok).

Simulasi *Trigger Auto\_update\_pengurangan\_stok* untuk proses retur barang:

1. Pada Gambar 4.25 terdapat kd\_barang A0001 dengan stok 22 dan nanti akan dilakukan pengurangan otomatis sebanyak jumlah barang yang retur (*insert* pada tabel retur barang).

**Gambar 4.25.** Tabel stok sebelum *insert* retur barang

1. Terjadinya *insert* pada tabel retur barang yang dilakukan oleh bagian gudang, retur barang dengan kode A0001 sebanyak 2(dua) seperti gambar dibawah ini.

**Gambar 4.26.** Tabel retur barang ketika *insert*

1. Pada Gambar 4.27 terjadi pengurangan sebanyak 2(dua) pada tabel stok dengan kode barang A0001 menjadi 20.

**Gambar 4.27.** Tabel stok setelah *insert* retur barang

**4.2.4. *View***

*View* pada rancangan basis data ini berfungsi untuk menyingkat perintah SELECT yang menggunakan INNER JOIN dengan banyak tabel. Adapun *View* yang dibuat untuk menyingkat perintah SELECT sebagai berikut :

1. **View Barang**

View Detail Barang merupakan gabungan dari tabel barang, harga\_beli, harga\_jual dan stok, yang digunakan untuk menampilkan detail barang. Berikut adalah *SQL* *syntax create view*\_detail\_barang :

CREATE VIEW `db\_rku`.`View\_detail\_barang`

 AS

(SELECT barang.kd\_barang,nama\_barang,harga\_beli,harga\_jual,stok FROM barang

INNER JOIN harga\_beli

ON barang.kd\_barang=harga\_beli.kd\_barang

INNER JOIN harga\_jual

ON barang.kd\_barang=harga\_jual.kd\_barang

INNER JOIN stok

ON barang.kd\_barang=stok.kd\_barang);

**Gambar 4.28.** *Syntax create view*\_detail\_barang

Setelah dijalankan perintah query maka didapat sebuah tabel *view*\_detail\_barang seperti pada tampilan gambar 4.27 dibawah ini:



**Gambar 4.29.** Struktur *view\_*detail\_barang

1. **View Detail Retur Pembelian**

View Retur pembelian merupakan gabungan dari tabel retur pembelian, pembelian dan barang, yang digunakan untuk menampilkan detail retur pembelian. Berikut adalah *SQL* *syntax create view* retur pembelian :

CREATE VIEW `db\_rku`.`view\_detail\_retur\_pembelian`

 AS

(SELECT no\_retur,tgl\_retur,nama\_barang,jumlah\_retur,kondisi,retur\_pembelian.no\_pembelian FROM retur\_pembelian

INNER JOIN pembelian

ON retur\_pembelian.no\_pembelian=pembelian.no\_pemelian

INNER JOIN barang

ON pembelian.kd\_barang=barang.kd\_barang);

**Gambar 4.30.** *Syntax create view* retur pembelian

Setelah dijalankan perintah query maka didapat sebuah tabel view retur pembelian seperti pada tampilan gambar 4.29 dibawah ini.

**Gambar 4.31.** Struktrur tabel *view\_detail\_*retur\_pembelian

1. **View Detail Pembelian**

View Pembelian merupakan gabungan dari tabel pembelian, barang, bagian\_gudang dan harga\_beli, yang digunakan untuk menampilkan detail pembelian. Berikut adalah *SQL* *syntax create view* detail pembelian :

CREATE VIEW `db\_rku`.`view\_detail\_pembelian`

 AS

(SELECT no\_pembelian,tgl\_masuk,pembelian.kd\_barang,nama\_barang,harga\_beli,jumlah,total\_biaya,nama\_karyawan FROM pembelian

INNER JOIN barang

ON pembelian.kd\_barang=barang.kd\_barag

INNER JOIN bagian\_gudang

ON pembelian.kd\_karyawan=bagian\_gudang.kd\_karyawan

INNER JOIN harga\_beli

ON pembelian.kd\_barang=harga\_beli.kd\_barang);

**Gambar 4.32.** *Syntax create view* detail pembelian

Setelah dijalankan perintah query maka didapat sebuah tabel *view* detail pembelian seperti pada tampilan gambar 4.31 dibawah ini.

**Gambar 4.33.** Struktur tabel *view* detail pembelian

 **BAB IV**

HASIL DAN PEMBAHASAN

* 1. Kesimpulan

Adapun kesimpulan yang dapat diambil dari dalam penelitian ini ialah :

1. Hasil dari perancangan basis data ini menghasilkan suatu rancangan basis data inventori, dengan nama basis data DBRku yang terdiri dari 8 struktur tabel yaitu barang, harga\_jual, harga beli, stok, suplier, bagian\_gudang, pembelian, retur\_pembelian.
2. Hasil penelitian adalah basis data inventori RKU Komputer yang sudah terintegrasi antara tabel satu dan lainnya, sehingga redudansi data dapat dikurangi.
3. Basis data ini diharapkan dapat mempermudah dalam pengembangan aplikasi sistem inventori barang karena sudah di lengkapi dengan *Trigger* yang memungkinkan beberapa proses pengolahan data dapat dilakukan di *database* dan penggunaan *View* dapat meringankan proses *Query* yang memerlukan relasi lebih dari satu tabel.
	1. Saran

Untuk mendukung keberhasilan dalam pemanfataan aplikasi ini ini, penulis memberikan saran sebagai berikut :

1. Diperlukan staf ahli khusus basis data untuk menjadi administrator, sehingga dapat mengimplementasikan basis data inventori ini dengan maksimal.
2. Perlu diadakannya pelatihan tentang basis data kepada seluruh karyawan untuk menambah kemampuan karyawan dalam mengelola basis data.
3. Penelitian ini disarankan untuk dilakukan pengembangkan menjadi aplikasi dekstop ataupun web, sesuai kebutuhan.

**DAFTAR PUSTAKA**

Connolly, T. M. Dan Begg, C. E. 2005. *Database Systems,* edisi ke-3. USA: Addison Wesley.

Efendi, Yanti. (2013), “*Analisis Perancangan Basis Data Klaim Asuransi Kerugian PT. Asuransi Bina Dana Arta TBK****”,*** diakses: <http://news.palcomtech.com/wp-content/uploads/2013/03/YANTI_TE02032012.pdf> ,pada tanggal: 20 mei 2013.

Fathansyah, (2002), *Sistem Basis Data*, Bandung : Penerbit Informatika.

Heriyanto, B. (2004), *Sistem Manajemen Basis Data,* Bandung : Penerbit Informatika.

Indrajani, (2011), Perencanaan *Basis Data dalam All in 1*, Jakarta : elex Media Komputindo.

Kadir, Abdul, (2009), *Dasar Perancangan Dan Implementasi Database Relasional,* Yogyakarta : Penerbit Andi.

Nugroho, A. (2011), *Perancangan dan Implementasi Sistem Basis Data,* Yogyakarta : Penerbit Andi

Priyanto, Rahmat. (2007), *Menguasai MySQL 5*, Jakarta : Elex Media Komputindo.

Raharjo, Budi. (2011), *Belajar otodidak membuat database menggunakan MySQL,* Bandung : Informatika.

## Senjaya, Rudi. (2008), “*Persfektif Penerapan Penelitian Dan Pengembangan ( Litbang) Dalam Penyelenggaraan Pemerintahan Daerah*”,diakses: <http://bapedakabtasik.wordpress.com/2008/05/05/persfektif-penerapan-penelitian-dan-pengembangan-litbang-dalam-penyelenggaraan-pemerintahan-daerah/> , pada tanggal : 20 mei 2013.

Swandy, David, Adiputra, Mario. (2012), “*Perancangan Sistem Basis Data Operasional Pada Smp Strada Santo Fransiskus Xaverius 2”,* diakses: <http://library.binus.ac.id/eColls/eThesis/Lain-lain/2012-1-00616-SI%20Ringkasan.pdf>, pada tanggal: 20 mei 2013.