

Bab II

LANDASAN TEORI

1.1 Teori yang Berhubungan dengan Penelitian

Teori yang berhubungan dengan topik yang diangkat meliputi persediaan (*inventory*), fungsi persediaan (*inventory*), sistem *inventory* produk, pemasok, dan gudang.

1.1.1 Persediaan (*Inventory*)

Inventory (sediaan) yang merupakan semua jenis barang yang dimiliki perusahaan dan digunakan untuk mendukung proses bisnisnya (Martono, 2015:210).

1.1.2 Fungsi dari *Inventory*

Inventory dapat memberikan beberapa fungsi, yang akan menambah fleksibilitas operasi produksi suatu perusahaan. Sejumlah fungsi yang diberikan *inventory* di antaranya adalah:

- a. Untuk dapat memenuhi antisipasi permintaan pelanggan, dimana *inventory* merupakan upaya antisipasi stok, karena diharapkan dapat menjaga terdapatnya kepuasan yang diharapkan pelanggan.
- b. Untuk memisahkan berbagai parts atau komponen dari operasi produksi, sehingga dapat dihindari hambatan dari adanya fluktuasi, karena telah adanya *inventory* ekstra guna memisahkan proses operasi produksi dengan pemasok.
- c. Untuk memisahkan operasi perusahaan dari fluktuasi permintaan, dan memberikan suatu stok barang yang akan memungkinkan dilakukannya pendekatan oleh pelanggan. *Inventory* itu merupakan jenis upaya membangun ritel.
- d. *Inventory* berfungsi untuk memperlancar keperluan operasi produksi, dimana *inventory* dapat membangun kepercayaan dalam menghadapi terjadinya pola musiman, sehingga *inventory* ini disebut sebagai *inventory* musiman.
- e. Untuk memisahkan operasi produksi dengan dengan kejadian atau event, dimana *inventory* digunakan sebagai penyangga di antara keberhasilan operasi produksi. Dengan demikian, kontinuitas operasi produksi dapat terjaga, dan dapat dihindari terdapatnya kejadian kerusakan peralatan, yang menyebabkan operasi produksi terhenti secara temporer.

- f. Untuk melindungi kekurangan stok yang dihadapi perusahaan, karena terlambatnya kedatangan *delivery* dan adanya peningkatan permintaan, sehingga kemungkinan terdapatnya resiko kekurangan pasokan.
- g. Untuk memanfaatkan keuntungan dari siklus pesanan, dengan cara meminimalisasi pembelian, dan biaya persediaan, yang dilakukan dengan membeli dalam jumlah yang melebihi jumlah kebutuhan segera.

1.1.3 Sistem *Inventory*

Sistem *inventory* adalah sekumpulan kebijakan dan pengendalian, yang memonitor tingkat *inventory*, dan menentukan tingkat mana yang harus dijaga, bila stok harus diisi kembali dan berapa banyak yang harus dipesan (Assauri, 2016:225).

Sistem *inventory* akan memberikan kemungkinan struktur organisasi dan kebijakan operasi produksi, untuk menjaga dan mengawasi barang-barang untuk distok. Dengan sistem *inventory* ini, diharapkan manajemen dapat bertanggung jawab terhadap pemesanan dan penerimaan barang yang dipesan. Hal ini dapat dilakukan dengan mengawasi jalannya jalur dari apa yang dipesan, serta berapa banyak barang yang dipesan dan dari siapa vendornya (Assauri, 2016:229).

1.1.4 Produk

Produk adalah sperangkat atribut baik berwujud, termasuk di dalamnya masalah warna, harga, nama baik pabrik, nama baik tokoh yang menjual (pengecer), dan pelayanan pengecer, yang diterima oleh pembeli guna memuaskan keinginannya. Produk adalah segala sesuatu yang dapat ditawarkan di pasar, untuk memuaskan kebutuhan dan keinginan konsumen. Produk terdiri atas barang, jasa, pengalaman, *events*, orang, tempat, kepemilikan, organisasi, informasi dan ide.

Jadi, produk itu bukan hanya berbentuk suatu yang berwujud saja, tetapi juga sesuatu yang tidak berwujud. Semua diperlukan bagi pemuasan kebutuhan dan keinginan dari konsumen. (Manap, 2016:225).

1.1.5 Supplier

Pemasok adalah suatu organisasi atau lembaga yang menyediakan atau memasok produk atau pelayanan kepada konsumen (Siregar, 2005:289).

1.1.6 Gudang

Gudang adalah tempat penyimpanan sementara dan pengambilan *inventory* untuk mendukung kegiatan operasi bagi proses operasi berikutnya, ke lokasi distribusi, atau kepada konsumen akhir. Gudang berfungsi untuk :

- a. Menyimpan barang sementara waktu sambil menunggu giliran untuk diproses.
- b. Memantau pergerakan dan status barang.
- c. Menyediakan media komunikasi dengan konsumen mengenai barang.
- d. Titik penyeimbang aliran *inventory* dan barang.

Jika *inventory* berlebihan, diletakkan di gudang. Setelah itu, bagian gudang harus memastikan bahwa *inventory*/barang disimpan baik, terjaga kondisinya, dan tercatat statusnya sehingga tidak ada modal perusahaan yang hilang akibat kesalahan pengawasan di gudang (Martono, 2015:343-344).

1.2 Teori yang Berhubungan dengan Sistem Secara Umum

Teori yang berhubungan dengan sistem meliputi data, sistem, informasi, dan sistem informasi.

1.2.1 Data

Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (event) adalah sesuatu yang terjadi pada saat yang tertentu. Data merupakan bentuk yang masih mentah yang belum dapat berceritera banyak, sehingga perlu diolah lebih lanjut (Jogiyanto HM, 2005).

1.2.2 Sistem

Sistem adalah jaringan dari pada elemen-elemen yang saling berhubungan, membentuk satu kesatuan untuk melaksanakan suatu tujuan pokok dari sistem tersebut (Jogiyanto HM, 2005).

1.2.3 Informasi

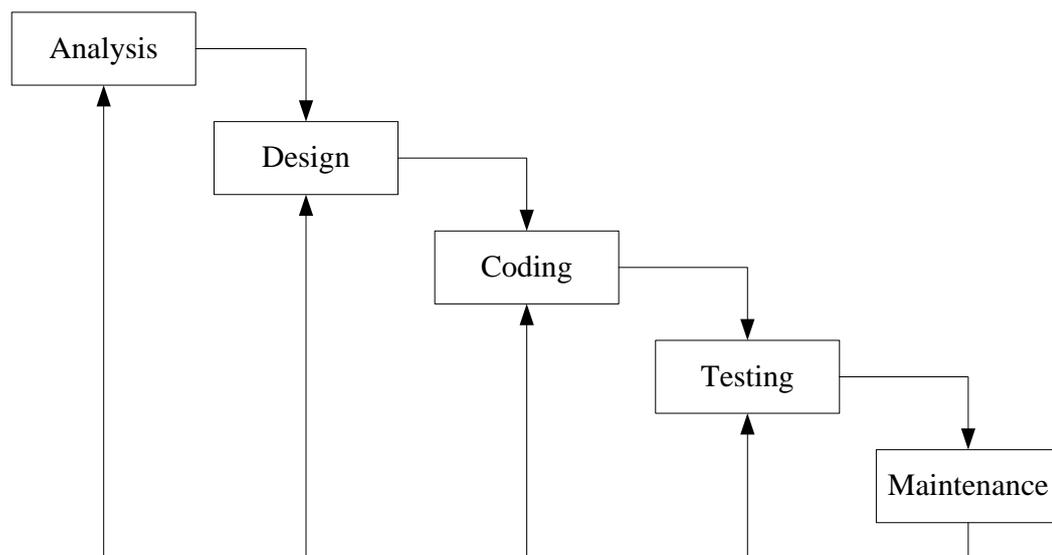
Informasi adalah data yang diolah menjadi satu bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Jogiyanto HM, 2005).

1.2.4 Sistem Informasi

Sistem informasi adalah suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto HM, 2005).

2.6 Metode Pengembangan Sistem

Menurut (Saputra dan Agutin, 2012), perancangan sebuah aplikasi atau sistem memerlukan metode-metode dalam pembangunan atau pengembangan sistem. Perancangan sistem informasi inventory dirancang dengan menggunakan Model *Waterfall*. Metode ini membutuhkan pendekatan yang sistematis dan sekuensial. Tahapan dalam metode ini diantaranya sebagai berikut :



Gambar 2.1 Metode Waterfall

1. Analysis

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur. Sistem analis akan menggali informasi sebanyak-banyaknya dari user sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut. Tahapan ini akan menghasilkan dokumen user

requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen ini lah yang akan menjadi acuan sistem analis untuk menerjemahkan ke dalam bahasa pemrogram.

2. Design

Tahapan dimana dilakukan penuangan pikiran dan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem seperti diagram alir data (data flow diagram), diagram hubungan entitas (entity relationship diagram) serta struktur dan bahasan data.

3. Coding (Pengkodean)

Pengkodean merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menerjemahkan transaksi yang diminta oleh user. Tahapan ini lah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki

4. Testing

Tahapan akhir dimana sistem yang baru diuji kemampuan dan keefektifannya sehingga didapatkan kekurangan dan kelemahan sistem yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi menjadi lebih baik dan sempurna.

5. Maintenance

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

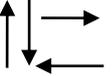
1.3 Teori yang Berkaitan dengan Alat Bantu Pemodelan

Adapun alat dan teknik yang digunakan pada penelitian ini yaitu bagan alir dokumen (*document flowchart*) yang digunakan untuk menggambarkan arus data sistem yang berjalan maupun sistem yang diusulkan, normalisasi sebagai salah satu teknik yang digunakan untuk mendesain *database*, kamus data sebagai alat dokumentasi untuk mendefinisikan struktur data dari *file-file database* dan bagan alir program (*program flowchart*) seperti logika program yang digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika.

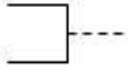
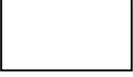
2.3.1 Flowchart

Bagan alir dokumen (*document flowchart*) atau disebut juga bagan alir formulir (*form flowchart*) adalah bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Berikut ini merupakan simbol-simbol yang terdapat pada bagan alir dokumen :

Tabel 2.1 Simbol *System Flowchart*

No.	Simbol	Keterangan	No.	Simbol	Keterangan
1.	Dokumen 	Menunjukkan dokumen <i>input</i> dan <i>output</i> baik proses manual, mekanil atau <i>computer</i>	11.	Hubungan Komunikasi 	Menunjukkan proses transmisi data melalui channel komunikasi
2.	Kegiatan Manual 	Menunjukkan pekerjaan manual	12.	Keyboard 	Menunjukkan <i>input/output</i> menggunakan on-line keyboard
3.	Garis Alir 	Menunjukkan arus proses	13.	Terminasi 	Menunjukkan awal dan akhir suatu proses
4.	Penghubung 	Menunjukkan penghubung ke halaman yang sama atau ke halaman lain	14.	Pita Kertas berlubang 	Menunjukkan <i>input/output</i> menggunakan pita kertas berlubang

Tabel 2.2 Lanjutan Simbol System *Flowchart*

5.	Penjelasan 	Penjelasan dari suatu proses	15.	Kartu Plong 	Menunjukkan <i>input/output</i> menggunakan kartu plong
6.	Simpanan Offline 	File non-komputer yang diarsip urut huruf (<i>alphabetical</i>)	16.	Simpanan Offline 	File non-komputer yang diarsip urut tanggal (<i>cronological</i>)
7.	Diskette 	Menunjukkan <i>input/output</i> menggunakan diskette	17.	Drum magnetik 	Menunjukkan <i>input/output</i> menggunakan Drum magnetik
8.	Harddisk 	Menunjukkan <i>input/output</i> menggunakan harddisk	18.	Pita magnetik 	Menunjukkan <i>input/output</i> menggunakan pita magnetik
9.	Proses 	Menunjukkan proses dari operasi program komputer	19.	Pngurutan offline 	Menunjukkan proses pengurutan data diluar proses komputer
10.	Display 	Menunjukkan output yang tampil di komputer			

(Sumber : Jogiyanto, 2005:796-799)

2.3.2 DFD (*Data Flow Diagram*)

Ada banyak cara untuk mempresentasikan proses model salah satunya menggunakan DFD (*Data Flow Diagram*). Ada 2 (dua) jenis DFD yaitu DFD logis dan DFD fisik. DFD logis menggambarkan proses tanpa menyoroti bagaimana mereka akan melakukan, sedangkan DFD fisik menggambarkan proses model (Fatta, 2007:105).

Data Flow Diagram (DFD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*) (Rosa dan Shalahuddin, 2014:70), sedangkan menurut Bahra (2013:64) *Data Flow Diagram* (DFD) merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. Maka dapat disimpulkan *Data Flow Diagram* (DFD) adalah representasi

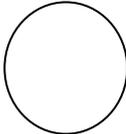
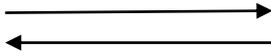
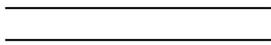
grafik atau pemodelan proses yang menggambarkan sistem ke modul yang lebih kecil.

Didalam DFD terdiri dari 3 Diagram yaitu (Bahra, 2013:64) :

1. Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh *input* ke sistem atau *output* dari sistem.
2. Diagram Nol/Zero (*Overview Diagram*) Diagram nol adalah diagram yang menggambarkan proses dari *data flow diagram*. Diagram nol memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, menunjukkan tentang fungsi-fungsi atau proses yang ada, aliran data, dan eksternal *entity*.
3. Diagram Rinci (*Level Diagram*)

Diagram rinci adalah diagram yang menguraikan proses apa yang ada dalam diagram *zero* atau diagram level. Berikut Tabel 2.3 simbol-simbol DFD

Tabel 2.3 Simbol DFD

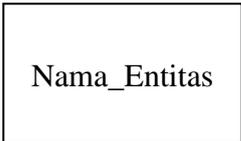
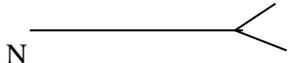
Keterangan	Simbol De Macro dan Yourdan
Proses	
<i>Data flow</i> (Arus Data)	
<i>Data store</i> (Simpanan Data)	
Entitas / Kesatuan Luar / <i>Source</i>	

(Sumber : Fatta, 2007:107)

2.3.3 Entity Relationship Diagram (ERD)

Entity Relation Diagram (ERD) adalah pemodelan awal basis data yang paling banyak digunakan untuk permodelan basis data relasional (Rosa dan Shalahuddin, 2014). ERD dapat menguji model dengan mengabadikan proses yang harus dilakukan, ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antara data. Simbol-simbol yang digunakan dalam pembuatan Entity Relationship Diagram terdiri dari:

Tabel 2.4 Simbol *Entity Relationship Diagram* (ERD)

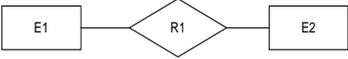
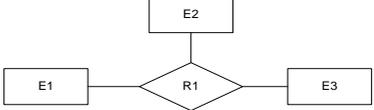
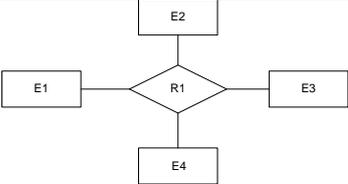
Simbol Chen	Keterangan
 Entitas/ Entity	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan disimpan agar dapat diakses oleh aplikasi komputer; penamaan entitas lebih ke kata benda.
 Atribut	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
 Relasi	Relasi yang menghubungkan antar entitas; diawali kata kerja
 Asosiasi/ Association	Penghubung antar relasi dan entitas

Tahapan-tahapan yang dilakukan ketika akan membuat Entity Relationship Diagram (ERD) yaitu sebagai berikut:

1. Menentukan entitas yang diperlukan
2. Menentukan relasi antar entitas
3. Menentukan cardinality
4. Menentukan atribut-atribut dan key yang diperlukan dari tiap entitas
5. Membuat Entity Relationship Diagram (ERD)

Dalam Entity Relationship Diagram relasi yang bisa terjadi antara dua entitas dapat dilihat dibawah ini :

Tabel 2.5 Hubungan Antar Entitas

No	Simbol	Keterangan
1.	 <p style="text-align: center;">Binary</p>	Satu relasi menghubungkan dua entitas
2	 <p style="text-align: center;">Ternery</p>	Satu relasi menghubungkan tiga buah entitas
3.	 <p style="text-align: center;">N-ary</p>	Satu relasi menghubungkan banyak entitas

2.3.4 Relasi Antar Tabel

1. One to One :

Mempunyai pengertian "Setiap baris data pada tabel pertama dihubungkan hanya ke satu baris data pada tabel ke dua".

2. One to Many:

Mempunyai pengertian "Setiap baris data dari tabel pertama dapat dihubungkan ke satu baris atau lebih data pada tabel ke dua".

3. Many to Many:

Mempunyai pengertian "Satu baris atau lebih data pada tabel pertama bisa dihubungkan ke satu atau lebih baris data pada tabel ke dua".

2.3.5 Spesifikasi File

Menjelaskan spesifikasi tabel-tabel yang diperoleh dari hasil normal pada tahap terakhir. Spesifikasi file dibuat berdasarkan DBMS (Database Management System) yang akan digunakan dalam membangun suatu sistem.

2.3.6 Kamus Data

Kamus data (Data Dictionary) adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (input) dan keluaran (output) dapat dipahami secara umum (memiliki standar penulisan). Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur (M. Shalahudin, 2014).

Format Kamus Data

Nama database :

Nama tabel :

Primary Key :

Foreign Key :

Nama Field	Type	Size	Kondisi	Keterangan

Keterangan: Kondisi berisi (contoh: NULL/NOT NULL)

Gambar 2.2 Format Kamus Data

2.3.7 Sistem Kode

Kode digunakan untuk mengklasifikasikan data, memasukkan data ke dalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya. Kode dapat berupa kumpulan angka, huruf dan karakter khusus. Di dalam merancang suatu struktur kode terdapat tipe dari kode yang dapat dibuat. (Jogiyanto HM, 2005). Maka berdasarkan hal tersebut struktur kode terbagi menjadi :

a. Kode Mnemonik

Kode mnemonik digunakan untuk tujuan supaya mudah diingat. Kode mnemonik dibuat dengan dasar singkatan atau mengambil sebagian karakter dari item yang akan diwakili dengan kode ini. Misalnya untuk kode barang “FS.001” untuk mewakili sparepart Filter Solar dan “001” mewakili jenis mobil yaitu Colt Diesel akan mudah diingat.

b. Kode Urut

Kode urut disebut juga dengan kode seri merupakan kode yang nilainya urut antara satu kode dengan kode berikutnya.

c. Kode Blok

Kode blok mengklasifikasikan item kedalam kelompok blok tertentu yang mencerminkan satu klasifikasi tertentu atas dasar pemakaian maksimum yang diharapkan.

d. Kode Grup

Kode Grup merupakan kode berdasarkan field – field dan tiap – tiap field kode mempunyai arti. Kalau anda mengamati buku – buku teks, maka akan terlihat suatu kode yang disebut dengan ISBN(International Standard Book Number) yang terdiri dari 10 digit terbagi dalam 4 field. ISBN merupakan kode group yang masing – masing field mempunyai arti tertentu.

e. Kode Desimal

Kode desimal mengklasifikasikan kode atas dasar 10 unit angka desimal dimulai dari angka 0 sampai dengan angka 9 atau dari 00 sampai dengan 99 tergantung dari banyaknya kelompok.

2.4 Software Aplikasi

Software aplikasi terdiri atas program yang berdiri sendiri yang mampu mengatasi kebutuhan bisnis tertentu. Aplikasi memfasilitasi operasi bisnis atau pengambilan keputusan manajemen maupun teknik sebagai tambahan dalam aplikasi pemrosesan data konvensional. Software aplikasi digunakan untuk mengatur fungsi bisnis secara real time (Pressman, 2010).

2.5 Basis Data Dan Sistem Manajemen Basis Data (*Database And Database Management System*)

Database adalah sekumpulan data yang saling berelasi (Elmasri, 2000). Database didesain, dibuat, dan diisi dengan data untuk tujuan mendapatkan informasi tertentu. Pendekatan database memiliki beberapa keuntungan seperti keberadaan katalog, independensi program-data, mendukung view (tampilan) untuk banyak pengguna, dan sharing data pada sejumlah transaksi. Selain itu masih ada fleksibilitas, ketersediaan up-to-date informasi untuk semua pengguna, skala ekonomis. Kategori utama pengguna database terbagi menjadi empat kategori, yakni Administrator, Designer, *End user*, *System Analyst dan Application Programmers*. Administrator atau *Data Base Administrator (DBA)* bertanggung jawab pada otoritas akses database, koordinasi dan monitoring penggunaan, dan pemilihan perangkat keras dan lunak yang dibutuhkan. Designer bertanggung jawab pada identifikasi data yang disimpan dalam database dan memilih struktur yang tepat untuk menggambarkan dan menyimpan data. *End User* adalah orang yang kegiatannya membutuhkan akses ke database untuk melakukan query, update, dan membuat laporan. *System Analysts* menentukan kebutuhan *End User*. *Application Programmers* mengimplementasikan program sesuai spesifikasi.

Sistem Manajemen Basis Data (SMBD) adalah program yang digunakan pengguna untuk membuat dan memelihara database. SMBD memfasilitasi untuk mendefinisikan, mengkonstruksi, dan memanipulasi database untuk berbagai aplikasi. Pendefinisian database meliputi spesifikasi tipe data, struktur, dan constraint untuk data yang disimpan dalam database. Pengkonstruksian database adalah proses penyimpanan data itu sendiri pada media penyimpanan.

Pemanipulasian *database* meliputi fungsi memanggil *query database* untuk mendapatkan data yang spesifik, update database, dan meng-generate laporan dari data tersebut (Elmasri, 2000). Keuntungan yang diperoleh menggunakan SMBD meliputi mengontrol redundansi, membatasi akses yang tidak berwenang, menyediakan penyimpanan yang persisten, menghasilkan *interface* (antar muka) banyak pengguna, menjaga integritas constraint, menyediakan *backup* dan *recovery*. Dalam SMBD menyediakan perintah yang digunakan untuk mengelola dan mengorganisasikan data, yakni *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). *Data Definition Language* adalah bahasa untuk mendefinisikan skema atau dan database fisik ke SMBD. (DDL). *Data Manipulation Language* adalah bahasa untuk memanipulasi data yaitu pengambilan informasi yang telah disimpan, penyisipan informasi baru, penghapusan informasi, modifikasi informasi yang disimpan dalam *database*. Selanjutnya, *query* adalah *statement* yang ditulis untuk mengambil informasi. Bagian dari DML yang menangani pengambilan informasi ini disebut bahasa *query*. SQL (dibaca "ess-que-el") singkatan dari *Structured Query Language*. SQL adalah bahasa yang digunakan untuk berkomunikasi dengan database. Menurut ANSI (*American National Standards Institute*), bahasa ini merupakan standard untuk relational database management systems (RDBMS). Secara prinsip, perintah-perintah SQL (biasa disebut dengan pernyataan) dapat dibagi dalam tiga kelompok, yaitu :

1. DDL (*Data Definition Language*) atau bahasa penerjemah data

Adalah perintah-perintah yang berkaitan dengan penciptaan atau penghapusan objek seperti tabel dan indek dalam database. Versi ANSI mencakup *Create Table*, *Create Index*, *Alter Table*, *Drop Table*, *Drop View*, dan *Drop Index*. Beberapa sistem database menambahkan pernyataan DDL seperti *Create Database* dan *Create Schema*.

2. DML (*Data Manipulation Language*) atau bahasa manipulasi data

Mencakup perintah-perintah yang digunakan untuk memanipulasi data. Misalnya untuk menambahkan data (*INSERT*), data (*SELECT*), mengubah data (*UPDATE*), dan menghapus data (*DELETE*).

3. DCL (Data Control Language) atau bahasa pengendali data

Merupakan kelompok perintah yang dipakai untuk melakukan otorisasi terhadap pengaksesan data dan pengalokasian ruang. Misalnya, suatu data bisa diakses si A, tetapi tidak bisa diakses oleh si B. Termasuk dalam kategori DCL yaitu pernyataan-pernyataan *GRANT*, *REVOKE*, *COMMIT*, dan *ROLLBACK* (Kadir, 1999)

2.7 Alat Bantu Perangkat Lunak Pendukung Pemrograman

Adapun alat bantu perangkat lunak pendukung pemrograman yang digunakan dalam perencanaan perancangan sistem diantaranya menggunakan NetBeans IDE (versi 8.02) dan MySQL sebagai basis data (database).

2.7.1 NETBEANS IDE (Versi 8.02)

NetBeans adalah Integrated Development Environment (IDE) berbasis Java dari Sun Microsystems yang berjalan di atas Swing. Swing sebuah teknologi Java untuk pengembangan aplikasi Desktop yang dapat berjalan di berbagai macam platform seperti Windows, Linux, Mac OS X and Solaris. Netbeans merupakan software development yang Open Source, dengankata lain software ini di bawah pengembangan bersama, bebas biaya NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra. Sun Microsystems mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama. Suatu IDE adalah lingkup pemrograman yang diintegrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan pembangun Graphic User Interface (GUI), suatu text atau kode editor, suatu compiler atau interpreter dan suatu debugger.

The NetBeans IDE adalah sebuah lingkungan pengembangan – sebuah kanvas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java - namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE. Netbeans IDE adalah sebuah produk bebas dengan

tanpa batasan bagaimana digunakan. NetBeans IDE mendukung pengembangan semua tipe aplikasi Java (J2SE, web, EJB, dan aplikasi mobile). Fitur lainnya adalah sistem proyek berbasis Ant, kontrol versi, dan refactoring.

2.7.2 MySQL (*My Structured Query Language*)

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multi thread*, dan *multi-user*. MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah Relational *Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Contoh MySQL adalah APACHE, XAMPP, WampServer.

2.8 Perancangan Aplikasi

Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungan-hubungannya (Sommerville, 2003). Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi software (Pressman, 1997). Tahap desain meliputi perancangan data, perancangan fungsional, dan perancangan antarmuka.

1. Perancangan data

Perancangan data mentransformasikan model data yang dihasilkan oleh proses analisis menjadi struktur data yang dibutuhkan pada saat pembuatan program (coding). Selain itu juga akan dilakukan desain terhadap struktur database yang akan dipakai.

2. Perancangan fungsional

Perancangan fungsional mendeskripsikan kebutuhan fungsi-fungsi utama perangkat lunak.

3. Perancangan antarmuka mendefinisikan bagaimana pengguna (user) dan perangkat lunak berkomunikasi dalam menjalankan fungsionalitas perangkat lunak.

2.9 Implementasi Dan Pengujian Unit

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Kemudian pengujian unit melibatkan verifikasi bahwa setiap unit program telah memenuhi spesifikasinya (Sommerville, 2003). Program sebaiknya dirilis setelah dikembangkan, diuji untuk memperbaiki kesalahan yang ditemukan pada pengujian untuk menjamin kualitasnya (Padmini, 2005). Terdapat dua metode pengujian yaitu :

- 1) Metode *white box* yaitu pengujian yang berfokus pada logika internal software (source code program).
- 2) Metode *black box* yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

Pada tahap pengujian, penulis melakukan metode *black box* yaitu menguji fungsionalitas dari perangkat lunak saja tanpa harus mengetahui struktur internal program (*source code*).