

BAB II

LANDASAN TEORI

2.1 Logistik

Proses pemenuhan pesanan pelanggan dan distribusi merupakan salah satu kegiatan pada proses bisnis logistik. Kegiatan logistik dalam suatu perusahaan memiliki peranan yang sangat penting karena merupakan pusat operasional dari perusahaan khususnya perusahaan perdagangan. *Performance* perusahaan harus didukung penuh dari bagian logistik untuk menjamin ketersediaan barang yang tepat jumlah, tepat mutu, tepat biaya, dan tepat waktu dalam rangka meningkatkan profitabilitas perusahaan dan efisiensi biaya operasional.

Menurut Gitosudarmo (1998) kegiatan logistik dapat didefinisikan sebagai suatu perpaduan dari sistem-sistem manajemen distribusi fisik, manajemen material dan transfer persediaan internal. Hal ini menyangkut segala aspek gerakan fisik dari, ke dan diantara lokasi serta fasilitas yang merupakan struktur operasi dari organisasi perusahaan yang bersangkutan.

Secara etimologi, logistik berasal dari bahasa Yunani kuno yaitu *logistikos* yang berarti terdidik atau pandai dalam memperkirakan perhitungan. Istilah logistik sudah banyak dikenal dalam masyarakat, terutama melalui lembaga atau instansi yang mempunyai urusan dengan bidang tersebut.

Pengertian logistik menurut H. Subagya (2008) pada hakekatnya mencakup tiga pengetahuan dasar, yaitu:

1. Luas ruang lingkup (*scope*) yang mencakup segi-segi khusus tertentu administrasi militer.

2. Kedudukannya disamping sejajar dengan ilmu strategi dan ilmu taktik, logistik juga merupakan *the third major branch of the military art* (kegiatan utama ketiga dari seni militer).
3. Arti asalnya, pandai dalam mengadakan atau merumuskan perkiraan-perkiraan.

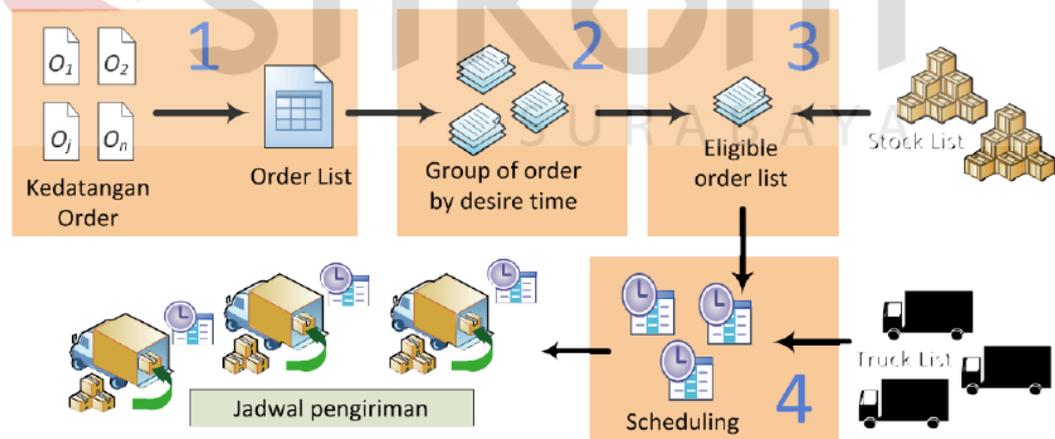
Istilah logistik paling banyak dikenal dikalangan militer. Dalam hal kemiliteran. Logistik merupakan salah satu unsur yang kegiatannya merupakan faktor pendukung terhadap pertempuran dan peperangan, dengan demikian sukses atau tidaknya pertempuran ditentukan pula oleh kemampuan dalam memberikan logistik untuk operasi militer, lebih - lebih lagi kalau operasi cukup besar dan melibatkan ribuan anggota pasukan yang menggunakan peralatan dan persediaan makanan, bensin serta suku bahan bahan bakar, mesin termasuk suku cadang.

Bila diterjemahkan secara bebas mengenai logistik merupakan salah satu kegiatan yang bersangkutan dengan segi-segi:

1. Perencanaan dan pengembangan, pengadaan, penyimpanan, pemindahan, penyaluran, pemeliharaan, pengungsian dan penghapusan alat alat perlengkapan.
2. Pemindahan, pengungsian dan perawatan personil.
3. Pengadaan dan pembuatan, penyelenggaraan, pemeliharaan dan penghapusan fasilitas-fasilitas.
4. Pengusahaan atau pemberian layanan atau bantuan dalam hal ini mencakup perencanaan termasuk pula penentuan kebutuhan-kebutuhan serta penggunaannya.

Proses logistik atau kegiatan pendistribusian barang merupakan sebuah kegiatan dalam rangkaian kegiatan rantai pasok (*supply chain*) yang paling banyak

menghabiskan banyak sumber daya. Oleh karena itu, optimalisasi proses logistik sangat diperlukan demi efektif dan efisiensi pemanfaatan sumber daya. Optimalisasi proses logistik dapat dilakukan secara manual ataupun terkomputerisasi. Akan tetapi, jika area kerja logistik dipandang cukup luas dan rumit maka diperlukanlah sebuah sistem yang mampu mengendalikan proses logistik tersebut. Saat ini, sistem logistik terkomputerisasi merupakan solusi terbaik yang dipandang mampu mengatasi permasalahan logistik. Area kerja sebuah sistem logistik dibuat atau dikembangkan dengan menyesuaikan proses bisnis dimana sistem tersebut akan digunakan. Sistem logistik tersebut dibangun dengan tujuan untuk mengendalikan proses logistik sehingga proses pendistribusian barang dari gudang ke konsumen dapat dilakukan sesuai dengan kesepakatan waktu yang telah ditetapkan. Rangkaian proses logistik pada umumnya dimulai dari proses pengadaan bahan baku, produksi barang, penyimpanan barang, dan diakhiri dengan pendistribusian barang ke konsumen.



Gambar 2.1. Alur Kerja Sistem Logistik.

Pada gambar 2.1 alur kerja sistem logistik di jelaskan bahwa setiap hari sistem menerima *order* yang kemudian *order* tersebut dikumpulkan (*collecting order*) oleh sistem dalam sebuah basis data dan secara otomatis sistem akan

menyeleksi *order* dengan syarat-syarat tertentu (*eligible order*) untuk dijadwalkan. Sistem logistik mempunyai beberapa tahapan dalam proses penyusunan jadwal pengiriman barang yaitu pengumpulan *order*, pengelompokan *order*, pemilihan *eligible order*, dan penjadwalan.

2.2 Penjadwalan

Penjadwalan adalah aspek yang penting dalam pengendalian operasi baik dalam industri manufaktur atau jasa, dalam usaha meningkatkan pasar dan *volume* produksi untuk meningkatkan kepuasan terhadap konsumen, dengan penjadwalan yang efektif dapat meningkatkan keuntungan dalam fungsi operasi di waktu yang akan datang.

Penjadwalan adalah suatu proses pengambilan keputusan yang memainkan peranan penting dalam kebanyakan bidang manufaktur dan pelayanan industri penjadwalan digunakan dalam pengadaan bahan dan produksi dalam bidang transportasi dan distribusi serta dalam proses informasi dan komunikasi. Penjadwalan merupakan alat ukur yang baik bagi perencanaan agregat, untuk jangka pendek dalam rentang periode hari sampai satu bulan, perusahaan harus melakukan penjadwalan produksi untuk memenuhi order atau permintaan konsumen, penjadwalan tersebut untuk melaksanakan rencana agregat dan jadwal induk produksi yang telah dibuat. Order aktual adalah dasar untuk penjadwalan sumber daya produksi (fasilitas, tenaga kerja, dan peralatan), kemudian dilakukan pengurutan kerja pada setiap unit produksi sehingga dicapai optimalitas utilisasi dari kapasitas yang ada atau tujuan lain (Teguh Baroto, 2002).

Penjadwalan yang tidak efektif akan menghasilkan tingkat penggunaan yang rendah dari kapasitas yang ada, fasilitas, tenaga kerja dan peralatan akan

menunggu (*idle*) untuk waktu tertentu karena tidak ada jadwal. Sebagai akibatnya biaya produksi membengkak yang mana dapat menurunkan efektivitas dan daya saing perusahaan, meskipun kapasitas keseluruhan mungkin didesain agar biaya sumber daya minimal, penjadwalan yang tidak tepat dapat menyebabkan menurunnya tingkat pelayanan dan banyak hal lain secara tidak langsung (Teguh Baroto, 2002).

2.3 Analisis Dan Perancangan Sistem

Menurut Kendall (2003), analisa dan perancangan sistem dipergunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi terkomputerisasi. Sedangkan menurut J. Hartono (2005:129), analisis sistem adalah penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Berdasarkan penjelasan diatas, analisa sistem adalah sebuah tahap yang paling penting dalam suatu pemrograman dimana tahap ini untuk mengevaluasi permasalahan yang ada dan kendala-kendala yang dihadapi. Tahap analisis sistem dilakukan setelah tahap perencanaan sistem dan sebelum tahap desain sistem atau perancangan sistem.

Di dalam tahap analisis sistem terdapat langkah-langkah yang harus dilakukan oleh seorang analis sistem, yaitu:

1. *Identify*, merupakan langkah awal yang dilakukan dalam analisis sistem. Mendefinisikan masalah yang menyebabkan sasaran dari sistem tidak tercapai yang kemudian untuk dipecahkan.
2. *Understand*, memahami kerja dari sistem yang ada dengan cara mempelajari secara terinci bagaimana sistem yang ada beroperasi sebelum mencoba untuk menganalisis permasalahan, kelemahan, dan kebutuhan dari pemakai sistem untuk dapat memberikan rekomendasi pemecahannya.
3. *Analyze*, langkah ini dilakukan berdasarkan data yang telah diperoleh dari hasil penelitian yang telah dilakukan.
4. *Report*, laporan ini dibuat sebagai dasar untuk pemecahan masalah dan pencarian solusi dari permasalahan yang ada.

Perancangan adalah proses merancang atau menyusun atau mengembangkan sistem informasi yang lama menjadi sistem informasi yang baru. Dalam tahap ini dipastikan bahwa semua persyaratan untuk menghasilkan sistem informasi dapat dipenuhi. Seluruh sistem yang dirancang harus sesuai dengan kebutuhan pengguna untuk mendapatkan informasi.

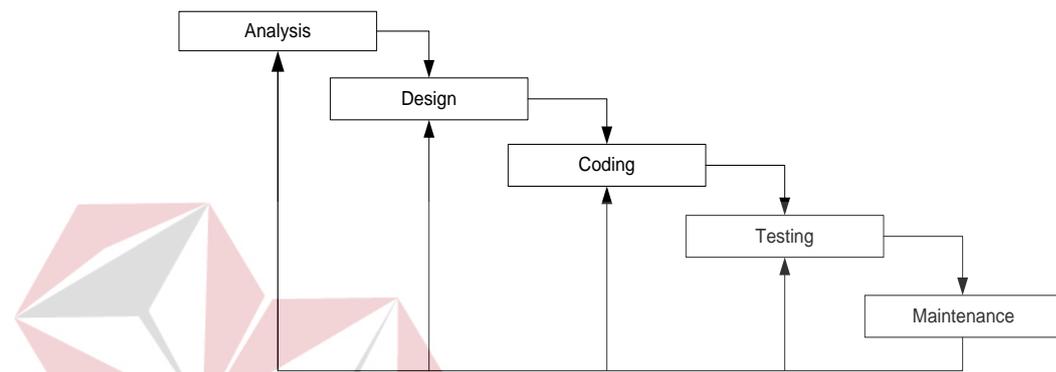
2.4 Rekayasa Perangkat Lunak

2.4.1 Definisi Rekayasa Perangkat Lunak

Menurut J. Hartono (2005), rekayasa perangkat lunak adalah suatu disiplin rekayasa yang berkonsentrasi terhadap seluruh aspek produksi perangkat lunak, mengadopsi pendekatan yang sistematis dan terorganisir terhadap pekerjaannya dan menggunakan *tool* yang sesuai serta teknik yang ditentukan berdasarkan masalah yang akan dipecahkan, kendala pengembangan, dan sumber daya yang tersedia.

2.4.2 Proses Perangkat Lunak

Proses perangkat lunak merupakan aktivitas yang saling terkait (*koheren*) untuk menspesifikasikan, merancang, mengimplementasikan, dan menguji sistem perangkat lunak. Proses perangkat lunak model *waterfall* dapat dilihat pada gambar 2.2 siklus model *waterfall*.



Gambar 2.2. Siklus Model *Waterfall*.

- Analysis* adalah tahap menganalisa hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan atau pengembangan *software*.
- Design* adalah tahap penerjemah dari keperluan-keperluan yang dianalisis ke dalam bentuk yang lebih mudah dimengerti oleh pemakai. Yaitu dengan cara menampilkan ke dalam Diagram Konteks, *Data Flow Diagram* (Diagram Aliran Data), *Entity Relationship Diagram*, Struktur Tabel, dan Struktur Menu.
- Coding* adalah tahap penerjemah data/pemecahan masalah *software* yang telah dirancang ke dalam bahasa pemrograman yang telah ditentukan.
- Testing* adalah tahap pengujian terhadap program yang telah dibuat. Pengujian ini dimulai dengan membuat suatu uji kasus untuk setiap fungsi pada perangkat lunak, kemudian dilanjutkan dengan pengujian terhadap modul-modul dan

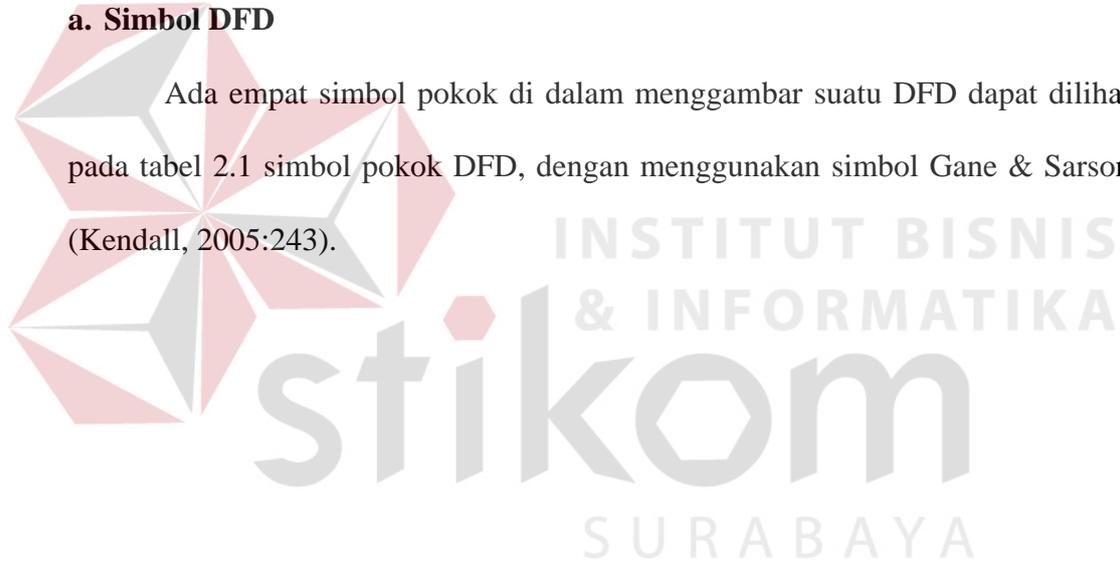
terakhir pada tampilan antar muka untuk memastikan tidak ada kesalahan dan semua berjalan dengan baik dan *input* yang diberikan hasilnya sesuai dengan yang diinginkan.

- e. *Maintenance* adalah perangkat lunak yang telah dibuat dapat mengalami perubahan sesuai permintaan pemakai. Pemeliharaan dapat dilakukan jika ada permintaan tambahan fungsi sesuai dengan keinginan pemakai ataupun adanya pertumbuhan dan perkembangan baik perangkat lunak maupun perangkat keras.

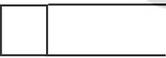
2.5 Data Flow Diagram

a. Simbol DFD

Ada empat simbol pokok di dalam menggambar suatu DFD dapat dilihat pada tabel 2.1 simbol pokok DFD, dengan menggunakan simbol Gane & Sarson (Kendall, 2005:243).



Tabel 2.1. Simbol Pokok DFD

NO	SIMBOL	NAMA	KETERANGAN
1		Entitas Luar	Merupakan entitas diluar sistem yang dapat berupa orang, organisasi / sistem lainnya yang berada di lingkungan luarnya yang memberikan masukan / menerima keluaran dari sistem.
2		Arus Data	Arus data mengalir di antara proses, simpanan data dan entitas luar.
3		Proses	Merupakan kegiatan yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Nama suatu proses biasanya berbentuk suatu kalimat diawali dengan kata kerja.
4		Simpanan Data	Merupakan simpanan data yang dapat berupa suatu file atau basis data di sistem komputer, suatu arsip atau catatan manual, suatu agenda atau buku.

b. Bentuk DFD

Terdapat dua bentuk DFD yaitu *physical data flow diagram* (PDFD) dan *logical data flow diagram* (LDFD) (Kendall, 2005:251). PDFD lebih menekankan pada bagaimana proses dari sistem diterapkan sedang LDFD lebih menekankan pada proses apa yang terdapat di sistem. PDFD lebih tepat digunakan untuk menggambarkan sistem yang ada (sistem yang lama). Penekanan dari PDFD adalah bagaimana proses–proses dari sistem diterapkan (dengan cara apa, oleh siapa dan

di mana) termasuk proses *manual*. LDFD lebih tepat digunakan untuk menggambarkan sistem yang akan diusulkan (sistem yang baru). LDFD tidak menekankan pada bagaimana sistem diterapkan, tetapi penekanannya hanya pada logika dari kebutuhan sistem yaitu proses apa secara logika yang dibutuhkan oleh sistem yang biasanya proses yang digambarkan hanya merupakan proses secara komputer saja.

c. Pembuatan DFD

Untuk memulai membuat DFD dari suatu sistem daftarkan semua komponen yang terlibat (entitas luar, proses, arus data dan simpanan data). Setelah semua teridentifikasi maka dilanjutkan dengan melakukan langkah berikut (Kendall, 2005:245):

a. Pembuatan *Context Diagram*

Context diagram adalah level tertinggi dalam sebuah DFD dan hanya berisi satu proses yang merupakan representasi dari suatu sistem. Proses dimulai dengan penomoran ke - 0 dan tidak berisi simpanan data.

b. Pembuatan diagram level 0

Diagram level 0 merupakan hasil pemecahan dari *context diagram* menjadi bagian yang lebih terinci yang terdiri dari beberapa proses. Sebaiknya jumlah proses pada level ini maksimal 9 proses untuk menghindari diagram yang sulit untuk dimengerti. Setiap proses diberikan penomoran dengan sebuah bentuk *integer*. Simpanan data mulai ditampilkan pada level ini.

c. Pembuatan *Child Diagram*

Setiap proses pada diagram level 0 dipecah lagi agar didapat level yang lebih terinci lagi (*child diagram*). Proses pada level 0 yang dipecah lebih terinci lagi disebut *parent process*. *Child diagram* tidak menghasilkan keluaran atau menerima masukan yang mana *parent process* juga tidak menghasilkan keluaran atau menerima masukan. Semua arus data yang menuju ke atau keluar dari *parent process* harus ditampilkan lagi pada *child diagram*.

d. Pengecekan kesalahan

Pengecekan kesalahan pada diagram digunakan untuk melihat kesalahan yang terdapat pada sebuah DFD. Kesalahan yang umum terjadi dalam pembuatan DFD yaitu:

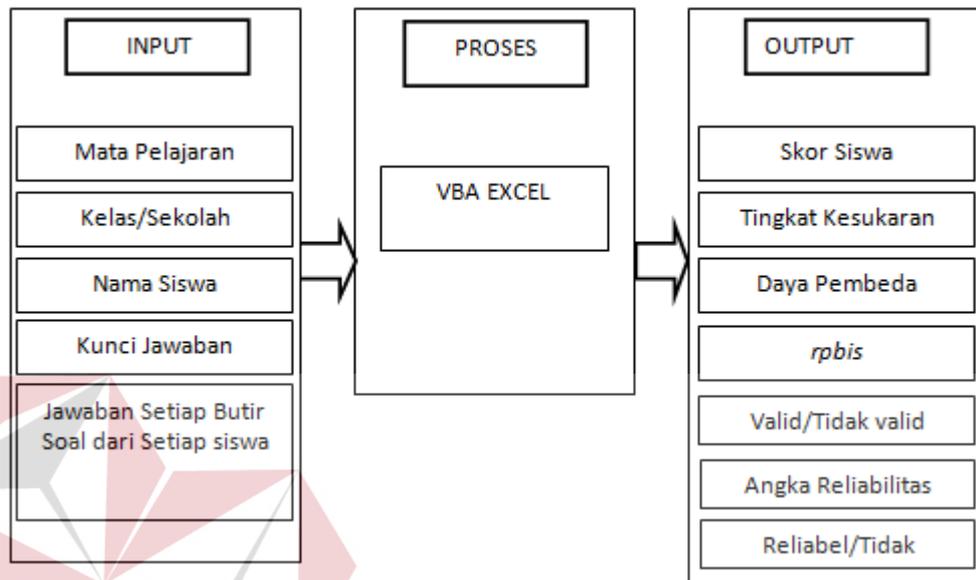
1. Sebuah proses tidak mempunyai masukan atau keluaran.
2. Simpanan data dengan entitas luar dihubungkan secara langsung tanpa melalui suatu proses.
3. Kesalahan dalam penamaan pada proses atau pada arus data.
4. Memasukkan lebih dari sembilan proses dalam sebuah diagram yang akan menyebabkan kebingungan dalam pembacaan.
5. Membuat ketidaksesuaian *decomposition* pada *child diagram*. Setiap *child diagram* harus mempunyai masukan dan keluaran yang sama dengan *parent process*.

2.6 Diagram Blok

Langkah awal yang dilakukan dalam menerjemahkan suatu bidang ilmu ke dalam sistem berbasis aturan adalah melalui diagram blok. Diagram blok

merupakan susunan dari rules yang terdapat di dalam sebuah bidang ilmu (Dologite, 1993). Contoh dari diagram blok dapat dilihat pada tabel 2.2.

Tabel 2.2. Diagram Blok.



2.7 Database

Menurut Marlinda (2004), *database* adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya.

1. Definisi Basis Data

Menurut Marlinda (2004), basis data atau *database* adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya.

Terdapat beberapa aturan yang harus dipatuhi pada file basis data agar dapat memenuhi kriteria sebagai suatu basis data. Beberapa aturan itu berhubungan dengan (Kroenke, 2002):

1. Kerangkapan data, yaitu munculnya data-data yang sama secara berulang-ulang pada *file* basis data.
2. Inkonsistensi data, yaitu munculnya data yang tidak konsisten pada *field* yang sama untuk beberapa *file* dengan kunci yang sama.
3. Data terisolasi, disebabkan oleh pemakaian beberapa *file* basis data.
4. Keamanan data, berhubungan dengan masalah keamanan data dalam sistem basis data.
5. Integrasi data, berhubungan dengan unjuk kerja sistem agar dapat melakukan kendali atau kontrol pada semua bagian sistem sehingga sistem selalu beroperasi dalam pengendalian penuh.

2. Sistem Basis Data (DBMS)

Menurut Marlinda (2004), *Database Management Sistem* (DBMS) merupakan kumpulan file yang saling berkaitan dan program untuk pengelolanya. Basis Data adalah kumpulan datanya, sedang program pengelolanya berdiri sendiri dalam suatu paket program yang komersial untuk membaca data, menghapus data, dan melaporkan data dalam basis data. DBMS (*Database Management System*) kumpulan program yang digunakan *user* untuk *me-management database* (*create, maintenance*). Proses yang terdapat dalam DBMS yaitu:

1. *Defining: database* mendefinisikan tipe data, struktur dan batasan (*constraint*) dari data yang disimpan dalam *database*.

2. *Manipulating: database* mencakup berbagai fungsi dan *query* untuk mendapatkan data yang dicari, termasuk operasi *insert*, *update* dan *delete* serta dalam *generate report data*.
3. *Sharing: database* dapat diatur untuk dapat *sharing multiple user* dan program untuk mengakses *database* secara bersama-sama.

Fungsi yang lebih penting dari DBMS adalah proteksi dan *maintenance database* dalam jangka panjang.

- a. Proteksi: mengandung *system protection* yang menangani kondisi *malfunction (crash)* baik pada *hardware* ataupun *software*, juga mengandung *security protection* yang menangani pengaksesan oleh *user* terlarang.
- b. *Maintenance*: mengandung sistem *maintenance* yang selalu meningkatkan kebutuhan perubahan tiap waktu.

3. Bagian-Bagian Basis Data

Membangun basis data adalah langkah awal dari pembuatan sebuah aplikasi. Keberhasilan dalam membangun basis data akan menyebabkan program lebih mudah dibaca, mudah dikembangkan dan mudah mengikuti perkembangan perangkat lunak. Berikut ini diuraikan mengenai komponen-komponen yang terdapat dalam basis data:

- a. Tabel, merupakan kumpulan dari suatu *field* dan *record* (kolom dan baris).
- b. *Field*, adalah sebutan untuk mewakili suatu *record*.
- c. *Record*, merupakan kumpulan elemen-elemen yang saling berkaitan yang berisi informasi tentang suatu isi data secara lengkap. Satu *record* mewakili

satu data atau informasi tentang seseorang misalnya, nomor daftar, nama pendaftar, alamat, tanggal masuk.

- d. *PrimaryKey*, merupakan kolom (*field*) yang menjadi titik acuan pada sebuah tabel yang bersifat unik, dimana dalam artian data tidak ada satu nilai pun yang sama atau kembar dalam tabel tersebut.
- e. *ForeignKey* (kunci relasi), suatu kolom dalam tabel yang digunakan sebagai “kaitan” atau relasi untuk membentuk satu hubungan yang didapati dari tabel induk, umumnya hubungan yang terbentuk antar tabel adalah satu ke banyak (*one to many*).
- f. *Index*, merupakan struktur basis data secara fisik, yang digunakan untuk mengoptimalkan pemrosesan data dan mempercepat proses pencarian data.

2.8 Testing dan Implementasi Sistem

Menurut Standar ANSI/IEEE1059, *testing* adalah proses menganalisa suatu entitas *software* untuk mendeteksi perbedaan antara kondisi yang ada dengan yang diinginkan (*defects/error/bugs*) dan mengevaluasi fitur dari entitas.

Menurut Hetzel, proses pemantauan kepercayaan akan kinerja program atau sistem sebagaimana yang diharapkan (Hetzel, 1973). Tiap aktivitas yang digunakan untuk dapat melakukan evaluasi suatu atribut atau kemampuan dari program atau sistem dan menentukan apakah telah memenuhi kebutuhan atau hasil yang diharapkan.

2.8.1 White Box Testing

White box testing atau *glass box testing* atau *clear box testing* adalah suatu metode desain *test case* yang menggunakan struktur kendali dari desain prosedural.

Metode desain *test case* ini dapat menjamin:

1. Semua jalur (*path*) yang independen/terpisah dapat dites setidaknya sekali tes.
2. Semua logika keputusan dites dengan jalur yang salah atau jalur yang benar.
3. Semua *loop* dapat dites terhadap batasannya dan ikatan operasionalnya.
4. Semua struktur internal data dapat dites untuk memastikan validasinya.

2.8.2 Black Box Testing

Black box testing atau *behavioral testing* atau *specification-based testing*, *input/output testing* atau *functional testing* dilakukan tanpa sepengetahuan detail struktur internal dari sistem atau komponen yang dites. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan spesifikasi kebutuhan dari *software*.

Menggunakan *black box testing*, perancang *software* dapat menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa keseluruhan kebutuhan fungsional pada suatu program.

Kategori *error* dapat diketahui melalui *black box testing*, antara lain:

1. Fungsi yang hilang atau tidak benar.
2. *Error* dari antar-muka.
3. *Error* dari struktur data atau akses *external database*.
4. *Error* dari kinerja atau tingkah laku.
5. *Error* dari inisialisasi dan terminasi.