

BAB II

LANDASAN TEORI

Pada landasan teori akan dijelaskan dasar-dasar teori yang berhubungan dengan permasalahan yang dibahas pada tugas akhir ini. Hal ini sangat penting karena teori-teori tersebut digunakan sebagai landasan pemikiran dalam tugas akhir ini, adapun teori-teori yang digunakan sebagai berikut:

2.1 Sistem Informasi

Menurut Fathansyah (2007) sistem informasi adalah suatu alat untuk menyajikan informasi sedemikian rupa sehingga bermanfaat bagi penerimanya. Tujuannya adalah untuk menyajikan informasi guna pengambilan keputusan pada perencanaan, pemrakarsaan, pengorganisasian, pengendalian kegiatan operasi suatu perusahaan yang menyajikan sinergi organisasi pada proses. Dengan demikian Sistem informasi berdasarkan konsep (*input, processing, output* – IPO).

Menurut Herlambang dan Tanuwijaya (2005), sistem informasi adalah elemen dari sistem yang terdiri dari tujuan, masukan keluaran, proses, mekanisme pengendali dan umpan lingkungan dan sistem yang lain. Sistem sendiri menurut Herlambang dan Tanuwijaya (2005) dapat didefinisikan melalui dua pendekatan, yaitu pendekatan prosedur yang mempunyai tujuan tertentu, sedangkan pendekatan komponen, sistem merupakan kumpulan dari komponen-komponen yang saling berkaitan untuk mencapai tujuan tertentu.

1. Tujuan

Tujuan merupakan pedoman sistem untuk melaksanakan tugas serta merupakan pemacu untuk mencapai hasil tertentu.

2. Masukan

Masukan (*input*) adalah segala sesuatu yang dimasukkan kedalam karakter-karakter huruf maupun berupa numerik. Data ini akan diproses dengan metode-metode tertentu dan akan menghasilkan *output* yang berupa informasi yang dihasilkan dapat berupa laporan maupun solusi dari proses yang dijalankan.

3. Proses

Kegiatan yang ada dalam proses meliputi, mencatat, mengklasifikasi, menghitung, menganalisis, membuat hipotesa dan perkiraan-perkiraan, menarik kesimpulan, serta membuat keputusan. Hasil proses ini akan diberikan pada bagian berikutnya yaitu *output*.

4. Keluaran

Keluaran (*output*) diterima dari proses yang dihasilkan. Hasil dari proses bisa berupa informasi, laporan, gambar, dan grafik.

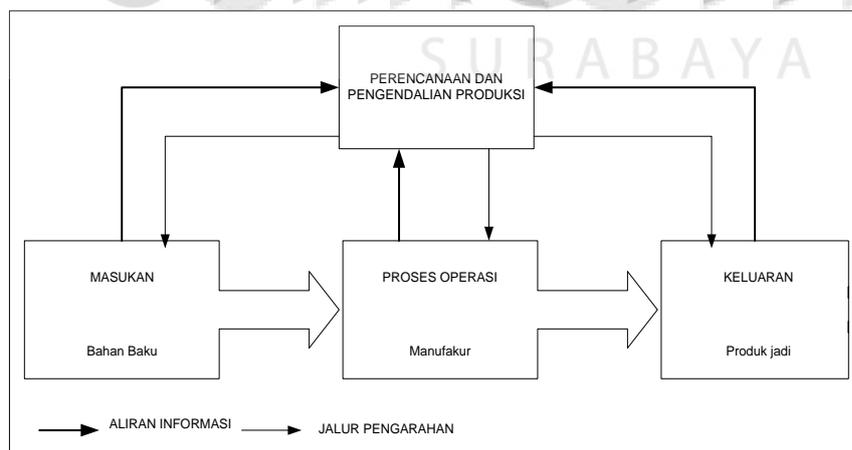
5. Batas

Batas merupakan pemisah antara sistem dengan daerah di luar sistem. Sistem yang berada di luar sistem disebut lingkungan. Ada delapan elemen lingkungan yang mempengaruhi sistem yaitu pemasok, pelanggan, serikat pekerja, masyarakat keuangan, pemegang saham atau pemilik, pesaing, pemerintah, dan masyarakat *global*.

2.2 Proses Manufaktur

Menurut Kusuma (2009: 5) proses manufaktur dapat digambarkan dalam kerangka masukan-keluaran seperti terlihat pada Gambar 1. Masukannya berupa bahan baku, selanjutnya bahan baku dikonversi (dengan bantuan peralatan, waktu, keahlian, uang, manajemen, dan lain sebagainya) menjadi keluaran yang kita sebut sebagai produk akhir. Pengendalian produksi berkepentingan dengan peramalan atau perkiraan keluaran, penentuan *input* yang dibutuhkan, serta perencanaan dan penjadwalan pengolahan bahan baku berdasarkan urutan produksi atau konversi yang dibutuhkan.

Proses konversi dapat amat sederhana namun dapat pula amat rumit. Aliran produk dapat berupa satuan yang kontinyu atau diskrit. Produk jadi dapat terdiri atas beberapa komponen yang didapatkan dari beberapa pemasok. Terdapat banyak hal yang mungkin terjadi selama material mengalir ke seluruh pabrik. Tetapi satu hal yang pasti: harus ada pengendalian terhadap segala proses konversi. Pada tempat inilah pengendalian produksi berperan.



Gambar 2.1 Manufaktur sebagai proses *input-output*

Dalam suatu organisasi, pengendalian produksi berguna untuk meningkatkan produktifitas. Definisi produktifitas adalah rasio nilai barang dan jasa yang dihasilkan dibagi dengan nilai sumber daya yang digunakan dalam produksi. Jika mesin atau orang menganggur karena tidak ada pekerjaan, atau komponen menumpuk di gudang karena tidak tersedia mesin untuk mengolah komponen tersebut, maka hal ini berarti sumber daya yang dimiliki terbuang percuma. Peran pengendalian produksi adalah meminimasi pemborosan dengan mengkoordinasikan ketersediaan tenaga kerja, peralatan dan bahan. Tak terhitung banyaknya kasus yang membuktikan bahwa persediaan dan kapasitas yang terlalu tinggi dapat menyebabkan organisasi kehilangan sejumlah besar uang. Perbaikan produktifitas dapat dilakukan dengan meningkatkan rancangan dan tatacara kerja produksi sehingga menjadi lebih efisien. Produktifitas juga dapat ditingkatkan dengan pengendalian produksi yang lebih baik.

2.3 Penjadwalan

Penjadwalan didefinisikan sebagai proses pengaturan waktu dari suatu kegiatan operasi, secara umum penjadwalan bertujuan untuk meminimalkan waktu proses, waktu tunggu langganan, dan tingkat persediaan, serta penggunaan yang efisien dari fasilitas, tenaga kerja, dan peralatan. Penjadwalan disusun dengan pertimbangan berbagai keterbatasan yang ada. Penjadwalan yang baik akan memberikan dampak positif, yaitu rendahnya biaya operasi dan waktu pengiriman, yang akhirnya dapat meningkatkan kepuasan pelanggan.

Adapun fungsi pokok dari penjadwalan produksi adalah untuk membuat proses produksi dapat berjalan lancar sesuai waktu yang telah direncanakan,

sehingga bekerja dengan kapasitas penuh dengan biaya seminimal mungkin serta kuantitas produk yang diinginkan dapat diproduksi tepat pada waktunya.

Menurut Heizer (2008:203) terdapat dua jenis penjadwalan yaitu penjadwalan *forward scheduling* dan penjadwalan *backward scheduling*.

2.4 Forward Scheduling

Penjadwalan *forward scheduling* dimulai ketika terdapat permintaan pengerjaan untuk suatu produk yang kemudian dibuat sebuah penjadwalan untuk waktu ke depan sampai produk selesai diproduksi. Contoh dari bidang usaha yang menggunakan *forward scheduling* adalah rumah sakit, klinik, restaurant, dan jasa antar makanan.

2.5 Backward Scheduling

Penjadwalan *backward scheduling* dimulai dari penjadwalan kegiatan produksi yang paling akhir berdasarkan waktu batas pengerjaan produk yang kemudian dikurangi dengan waktu yang dibutuhkan untuk pengerjaan produk tersebut sehingga ditemukan waktu untuk memulai pengerjaan produk tersebut.

2.6 Penjadwalan Produksi

Menurut Heizer (2008:546) Penjadwalan produksi merupakan salah satu fungsi dari pengawasan produksi yang mempunyai peranan yang cukup penting karena dapat mempengaruhi keberhasilan pengawasan produksi itu sendiri. Pada beberapa perusahaan, kegagalan atau kesalahan dalam penyusunan penjadwalan produksi tidak hanya dapat megacaukan usaha pengawasan produksinya, tetapi juga dapat mempengaruhi hal-hal lain dalam perusahaan seperti jumlah produk yang dihasilkan.

Unsur-unsur vital dalam penjadwalan adalah sumber-sumber (*resources*) yang dikenal dengan daya mesin, dan tugas-tugas (*tasks*) yang dikenal dengan pekerjaan-pekerjaan (*jobs*), untuk dapat melakukan penjadwalan dengan baik, maka waktu proses kerja setiap mesin serta jenis pekerjaan (*job*) yang akan dijadwalkan perlu diketahui.

Dengan penjadwalan produksi yang baik tentunya mesin-mesin yang digunakan dapat dioperasikan sesuai kapasitas yang dimiliki dan memperkecil kemungkinan timbulnya waktu yang tidak produktif dari mesin-mesin yang digunakan, meskipun belum tentu mesin tersebut dioperasikan sebatas kapasitas maksimum, namun demikian setidaknya-tidaknya dengan suatu penjadwalan produksi yang baik maka hasil produksi relatif akan lebih tinggi.

2.7 *Priority Rules*

Menurut Heizer (2008:612) *priority rules* menyediakan sebuah pedoman untuk membuat sebuah urutan pekerjaan yang harus dikerjakan terlebih dahulu. *Priority rules* berusaha untuk meminimalkan waktu penyelesaian proses, jumlah pekerjaan didalam system, dan memaksimalkan utilisasi fasilitas.

Berikut ini merupakan *priority rules* yang paling populer:

1. FCFS (*first come first serve*) : pekerjaan diproses berdasarkan pekerjaan yang datang lebih awal pada suatu pusat kerja
2. SPT (*shortest processing time*) : pekerjaan diproses berdasarkan waktu proses yang paling singkat yang terlebih dahulu dikerjakan.
3. EDD (*earliest due date*) : pekerjaan berdasarkan *due date* setiap pekerjaan, jadi pekerjaan yang harus selesai paling awal dikerjakan lebih dulu.

4. CR (*critical ratio*) : pekerjaan yang mempunyai rasio paling kecil dari due date terhadap lama waktu proses dikerjakan terlebih dahulu.
5. STR (*slack time to remaining*) : pekerjaan yang memiliki *slack time* yang terkecil dikerjakan terlebih dahulu.
6. LPT: *longest processing time*. Pekerjaan dengan waktu proses paling panjang akan diproses terlebih dahulu

Fokus penelitian ini adalah pada *Priority rules* dengan menggunakan metode *First Come First Serve* (FCFS).

Tabel 2.1 *Rules Priority*

Pekerjaan	Waktu Pekerjaan (pemrosesan) (hari)	Batas Waktu Pekerjaan (hari)
A	6	6
B	2	8
C	8	16
D	3	19
E	9	28

2.8 *First Come First Serve* (FCFS)

Menurut Jay Heizer dan Barry Render (2008:271) pada metode *First Come First Serve* (FCFS) yang pertama datang, yang pertama dilayani. Pekerjaan pertama yang datang di sebuah pusat kerja diproses terlebih dahulu.

Tabel 2.2 Tabel Perhitungan FCFS

Urutan Pekerjaan	Waktu Pekerjaan (pemrosesan)	Aliran Waktu	Batas Waktu Pekerjaan	Keterlambatan
A	6	6	8	0
B	2	8	6	2
C	8	16	18	0
D	3	19	15	4
E	9	28	23	5
Total	28	77		11

Aturan FCFS menghasilkan ukuran efektivitas berikut.

$$\begin{aligned} \text{a. Waktu penyelesaian rata-rata} &= \frac{\text{Jumlah aliran waktu total}}{\text{Jumlah pekerjaan}} \\ &= \frac{77}{5} = 15,4 \text{ hari} \end{aligned}$$

$$\begin{aligned} \text{b. Utilisasi} &= \frac{\text{Jumlah waktu pengerjaan (pemrosesan)}}{\text{Jumlah aliran waktu total}} \\ &= \frac{28}{77} = 36,4\% \end{aligned}$$

$$\text{c. Jumlah pekerjaan rata-rata dalam sistem} = \frac{\text{Jumlah aliran waktu total}}{\text{Jumlah waktu pekerjaan}}$$

$$\begin{aligned} &= \frac{77 \text{ hari}}{28 \text{ hari}} = 2,75 \text{ pekerjaan} \\ \text{d. Keterlambatan pekerjaan rata-rata} &= \frac{\text{Jumlah hari keterlambatan}}{\text{Jumlah pekerjaan}} \\ &= \frac{11}{5} = 2,2 \text{ hari} \end{aligned}$$

2.9 Shortest Processing Time (SPT)

Menurut Jay Heizer dan Barry Render (2008:271) pada metode *Shortest Processing Time* (SPT) waktu pemrosesan terpendek. Pekerjaan yang memiliki waktu pemrosesan terpendek ditangani dan diselesaikan terlebih dahulu.

Tabel 2.3 Tabel Perhitungan SPT

Urutan Pekerjaan	Waktu Pekerjaan (pemrosesan)	Aliran Waktu	Batas Waktu Pekerjaan	Keterlambatan
B	2	2	6	0
D	3	5	15	0
A	6	11	8	3
C	8	19	18	1
E	9	28	23	5
Total	28	65		9

Aturan SPT menghasilkan ukuran efektivitas berikut.

- a. Waktu penyelesaian rata-rata = $\frac{65}{5} = 13$ hari
- b. Utilisasi = $\frac{28}{65} = 43,1\%$
- c. Jumlah pekerjaan rata-rata dalam sistem = $\frac{65}{28} = 2,32$ pekerjaan
- d. Keterlambatan pekerjaan rata-rata = $\frac{9}{5} = 1,8$ hari

2.10 Earliest Due Dates (EDD)

Menurut Jay Heizer dan Barry Render (2008:271) pada metode *Earliest Due Date* (EDD) batas waktu paling awal. Pekerjaan dengan batas waktu yang paling awal dikerjakan terlebih dahulu.

Tabel 2.4 Tabel Perhitungan EDD

Urutan Pekerjaan	Waktu Pekerjaan (pemrosesan)	Aliran Waktu	Batas Waktu Pekerjaan	Keterlambatan
B	2	2	6	0
A	6	8	8	0
D	3	11	15	0
C	8	19	18	1
E	9	28	23	5
Total	28	68		6

Aturan EDD menghasilkan ukuran efektivitas berikut.

- a. Waktu penyelesaian rata-rata = $\frac{68}{5} = 13,6$ hari
- b. Utilisasi = $\frac{28}{68} = 41,2\%$
- c. Jumlah pekerjaan rata-rata dalam sistem = $\frac{68}{28} = 2,43$ pekerjaan
- d. Keterlambatan pekerjaan rata-rata = $\frac{6}{5} = 1,2$ hari

2.11 Longest Processing Time (LPT)

Menurut Jay Heizer dan Barry Render (2008:271) pada metode *Longest Processing Time* (LPT) waktu pemrosesan terpanjang. Pekerjaan yang memiliki waktu pemrosesan lebih panjang, lebih besar biasanya sangat penting dan diutamakan terlebih dahulu.

Tabel 2.5 Tabel Perhitungan LPT

Urutan Pekerjaan	Waktu Pekerjaan (pemrosesan)	Aliran Waktu	Batas Waktu Pekerjaan	Keterlambatan
E	9	9	23	0
C	8	17	18	0
A	6	23	8	15
D	3	26	15	11
B	2	28	6	22
Total	28	103		48

Aturan LPT menghasilkan ukuran efektivitas berikut.

e. Waktu penyelesaian rata-rata = $\frac{103}{5} = 20,6$

e. Utilisasi = $\frac{28}{103} = 27,2\%$

f. Jumlah pekerjaan rata-rata dalam sistem = $\frac{103}{28} = 3,68$ pekerjaan

g. Keterlambatan pekerjaan rata-rata = $\frac{48}{5} = 9,6$ hari

2.12 Siklus Hidup Pengembangan Sistem

Dalam merancang dan membangun sistem informasi, terdapat tahap-tahap terstruktur yang harus dijalani pengembang, yakni siklus hidup pengembangan sistem (SHPS) atau yang juga dikenal dengan *system development life cycle* (SDLC). Siklus hidup pengembangan sistem adalah pendekatan melalui beberapa

tahap untuk menganalisis dan merancang sistem yang dimana sistem tersebut telah dikembangkan dengan sangat baik melalui penggunaan siklus kegiatan penganalisis dan pengguna secara spesifik (Kendall, 2003:11). Tahapan-tahapan tersebut adalah sebagai berikut.

1. Mengidentifikasi masalah, peluang, dan tujuan.

Pada tahap ini, masalah, peluang, dan tujuan-tujuan yang hendak dicapai diidentifikasi. Tahap pertama ini bertujuan untuk melihat apa yang terjadi dalam bisnis, menentukan dengan tepat masalah-masalah yang terjadi, dan peluang yang mungkin dapat tercapai melalui penggunaan sistem informasi.

2. Menentukan syarat-syarat informasi.

Informasi yang dibutuhkan mulai dikumpulkan dalam tahap ini. Data-data yang diperlukan untuk perancangan sistem digali dari berbagai sumber dengan metode pengumpulan data yang berbeda-beda: wawancara, observasi, kuisioner, dan dokumentasi. Untuk setiap metode pengumpulan data, diperlukan instrument yang sesuai.

3. Menganalisis kebutuhan sistem.

Pada tahapan ini sistem, baik yang sudah terkomputerisasi atau belum, dianalisis dan diidentifikasi kebutuhannya. Tujuan dari analisis kebutuhan ini adalah agar sistem yang akan dibuat mampu menjawab permasalahan yang ada.

4. Merancang sistem yang direkomendasikan.

Dalam tahap desain dari siklus hidup pengembangan sistem, informasi-informasi yang terkumpul sebelumnya digunakan untuk mencapai desain sistem informasi yang logis. Tahap ini dilakukan dengan perancangan sistem

mulai *system flow*, diagram aliran data, antarmuka pengguna, basis data, perancangan fungsi-fungsi program dan file-file yang digunakan.

5. Mengembangkan dan mendokumentasikan perangkat lunak.

Pada tahapan ini sistem yang telah dirancang akan dikembangkan. Pengembangan sistem informasi dilakukan dengan menggunakan bahasa pemrograman dan tool pendukungnya sesuai rancangan yang telah dibuat sebelumnya. Dokumentasi yang baik juga perlu dilakukan dengan pembuatan *user manual* atau panduan penggunaan perangkat lunak.

6. Menguji dan mempertahankan sistem.

Pengujian harus dilakukan terlebih dahulu sebelum sistem informasi dapat digunakan. Rangkaian pengujian ini dijalankan bersama-sama dengan data contoh dan data aktual dari sistem yang telah ada.

7. Mengimplementasikan dan mengevaluasi sistem

Setelah sistem informasi yang telah dikembangkan diuji ketepatannya maka sistem informasi tersebut siap untuk diimplementasikan dalam organisasi. Sistem baru ini juga perlu dievaluasi, apakah sistem informasi yang telah dibuat dapat mengatasi permasalahan yang ada, digunakan dan membantu pengguna yang menggunakannya.

2.13 Analisis dan Perancangan Sistem

Menurut Kendall (2003:7), analisis dan perancangan sistem merupakan kegiatan menganalisis input data atau aliran data secara sistematis, memproses atau mentransformasikan data, menyimpan data, dan menghasilkan output informasi dalam konteks bisnis khusus. Analisis dan perancangan sistem digunakan untuk menganalisis, merancang, dan mengimplementasikan

peningkatan-peningkatan fungsi bisnis yang bisa dicapai melalui penggunaan sistem informasi terkomputerisasi.

Analisis dan desain sistem ditujukan untuk menyusun analisis dan perancangan sistem informasi, usaha yang banyak yang memungkinkan hal-hal yang tidak terpikirkan diatasi dengan berbagai cara. Hal ini bisa dianggap sebagai serangkaian proses yang secara sistematis dilakukan untuk meningkatkan bisnis melalui penggunaan sistem informasi terkomputerisasi. Bagian terbesar dari analisis dan perancangan sistem melibatkan kerja sama dengan pengguna sistem informasi.

2.14 Document Flow

Menurut Kendall (2003:125) *Document flow* adalah diagram yang menggambarkan aliran dokumen pada sistem. Dokumen tersebut dihasilkan dari proses yang dilakukan oleh sistem. *Document flow* dapat digambarkan berdasarkan sistem yang telah dibuat sebelumnya ataupun sistem yang akan dibuat sehingga memudahkan pengembangan untuk membuat/mengembangkan sistem.

2.15 System Flow

Menurut Kendall (2003:127) *System Flow* adalah diagram yang menunjukkan arus pekerjaan secara menyeluruh dari sebuah sistem dimana pada bagian ini setiap prosedur dari sistem dijelaskan. Pada *system flow* ini juga bagian dari sistem (subsistem) tergambarkan dengan jelas beserta fungsi-fungsi yang ada di dalamnya. Beberapa macam symbol yang digunakan dalam pembuatan system flow untuk merancang sebuah sistem diantaranya *terminator*, *manual operation*,

document, process, database, manual input, decision, off-line storage, on page reference, dan off-page reference.

2.16 Diagram Aliran Data

Menurut Kendall (2003:263), diagram aliran data (DAD) atau yang juga disebut *data flow diagram* (DFD) menggambarkan pandangan sejauh mungkin mengenai masukan, proses, dan keluaran sistem, yang berhubungan dengan masukan, proses, dan keluaran dari model sistem umum. Adapun beberapa kelebihan pendekatan aliran data menurut Kendall adalah sebagai berikut.

1. Kebebasan dari menjalankan implementasi teknis sistem yang terlalu dini.
2. Pemahaman lebih jauh mengenai keterkaitan satu sama lain dalam sistem dan subsistem.
3. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram aliran data.
4. Menganalisis sistem yang diajukan untuk menentukan apakah data-data dan proses yang diperlukan sudah ditetapkan.

2.17 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah diagram dari sistem yang menggambarkan hubungan antar entitas beserta relasinya yang saling terhubung. Menurut Marlinda (2004:28), relasi adalah hubungan antar entitas yang berfungsi sebagai hubungan yang mewujudkan pemetaan antar entitas. ERD terdiri atas 3 bagian, yaitu:

1. Entitas

Entitas adalah sesuatu yang ada dan didefinisikan dalam sebuah organisasi.

Entitas dalam berbentuk abstrak dan nyata.

2. Relasi

Relasi merupakan hubungan antar entitas yang berfungsi untuk menunjukkan hubungan antar entitas.

3. Atribut

Atribut adalah keterangan yang menempel pada entitas dan relasi. Atribut terdiri atas:

a) *Simple attribute*

Atribut tipe ini adalah atribut yang unik dan tidak dimiliki oleh atribut yang lain.

b) *Composite attribute*

Atribut ini adalah atribut yang memiliki dua nilai.

c) *Single value attribute*

Atribut ini adalah atribut yang hanya memiliki satu nilai

d) *Multi value attribute*

Atribut ini adalah atribut yang memiliki banyak nilai.

e) *Null value attribute*

Atribut ini adalah atribut yang tidak memiliki nilai

Entity Relationship Diagram (ERD) diperlukan untuk menggambarkan dua jenis model:

1. *Conceptual Data Model*

Conceptual Data Model (CDM) adalah jenis model data yang menggambarkan hubungan antar table secara konseptual.

2. *Physical Data Model*

Physical Data Model (PDM) adalah jenis model data yang menggambarkan hubungan antara table secara fisikal.

2.18 Konsep Sistem Basis Data

2.18.1 Sistem Basis Data

Menurut Marlinda (2004:1), sistem basis data adalah suatu sistem menyusun dan mengola *records* menggunakan komputer untuk menyimpan atau merekam serta memelihara dan operasional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi optimal yang diperlukan pemakai untuk proses mengambil keputusan. Pada sebuah sistem basis data terdapat komponen-komponen utama yaitu perangkat keras (*hardware*), sistem operasi (*operating system*), basis data (*database*), sistem pengelola basis data (DBMS), pengguna (*user*), dan aplikasi lain yang sifatnya opsional.

Kelebihan dari sistem basis data yang mendasari penggunaannya adalah sebagai berikut.

1. Mengurangi kerangkapan data (*redundancy*), yaitu data yang sama disimpan dalam berkas data yang berbeda-beda sehingga *update* dilakukan berulang-ulang.
2. Mencegah ketidakkonsistenan.
3. Keamanan data dapat terjaga, yaitu data dapat dilindungi dari pemakai yang tidak berwenang.

4. Integritas dapat dipertahankan.
5. Data dapat dipergunakan bersama-sama.
6. Menyediakan *recovery*.
7. Memudahkan penerapan standarisasi.
8. Data bersifat mandiri (*data independence*).
9. Keterpaduan data terjaga, memelihara keterpaduan data berarti data harus akurat. Hal ini sangat erat hubungannya dengan pengontrolan kerangkapan data dan pemeliharaan keselarasan data.

Adapun kekurangan-kekurangan yang patut dipertimbangkan dalam penggunaannya adalah sebagai berikut.

1. Diperlukan tempat penyimpanan yang besar.
2. Diperlukan tenaga yang terampil dalam mengolah data.
3. Kerusakan sistem basis data dapat mempengaruhi departemen yang terkait.

2.18.2 Sistem Manajemen Basis Data

Menurut Marlinda (2004:6), Sistem Manajemen Basis Data atau *Database Management System* (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.

Bahasa-bahasa yang terdapat dalam DBMS adalah sebagai berikut.

1. *Data Definition Language* (DDL)

Pola skema basis data dispesifikasikan dengan satu set definisi yang diekspresikan dengan satu bahasa khusus yang disebut DDL. Hasil kompilasi

perintah DDL adalah satu set table yang disimpan di dalam *file* khusus yang disebut *data dictionary/directory*.

2. *Data Manipulation Language (DML)*

Bahasa yang digunakan untuk mengakses dan memanipulasi data yang ada di dalam *database*.

Fungsi dasar dari sebuah DBMS adalah sebagai berikut.

1. *Data Definition*

DBMS harus dapat melakukan pendefinisian data.

2. *Data Manipulation*

DBMS harus dapat melakukan perubahan/manipulasi data pada *database*.

3. *Data Security and Integrity*

DBMS dapat memeriksa keamanan dan integritas sesuai yang ditentukan oleh DBA.

4. *Data Recovery and Concurrency*

DBMS harus dapat melakukan penanganan terhadap data-data yang hilang akibat kesalahan sistem, kerusakan harddisk, dan lain sebagainya. DBMS juga harus menjaga *concurrency*, yakni dimana ketika *database* diakses lebih dari 1(satu)*user*.

5. *Data Dictionary*

DBMS harus mempunyai *data dictionary*.

2.19 Interaksi Manusia-Komputer

Rizky (2006:4) mendeskripsikan interaksi manusia-komputer sebagai sebuah disiplin ilmu yang mempelajari desain, evaluasi, implementasi dari sistem komputer interaktif untuk dipakai oleh manusia, beserta studi tentang faktor-

faktor utama dalam lingkungan interaksinya. Interaksi antara manusia dan komputer bertemu pada *user interface* (atau yang sering disebut dengan *interface*) yang meliputi *software* dan *hardware*. Input diterima melalui *hardware* seperti *keyboard* dan *mouse*, kemudian diteruskan ke *software* untuk diolah dan data/informasi hasil olahan tersebut dikeluarkan dalam bentuk tampilan layar ataupun pada hardware yang lain. Interaksi Manusia dan Komputer ini penting dalam pembuatan sistem dalam penelitian ini karena proses pembuatan *website* (yang merupakan basis aplikasi yang akan dibuat) memerlukan desain interaksi manusia dan komputer yang baik sehingga *user* yang menggunakannya dapat mengerti dan dapat dengan mudah mengoperasikannya.

2.20 *Testing Software*

Menurut Romeo (2003:3), *testing* adalah proses mengoperasikan *software* dalam suatu kondisi yang dikendalikan untuk melakukan verifikasi apakah telah berlaku sebagaimana yang telah ditetapkan (menurut spesifikasi), mendeteksi error, dan validasi apakah spesifikasi yang telah ditetapkan sudah memenuhi keinginan atau kebutuhan dari pengguna yang sebenarnya. Verifikasi merupakan pengecekan atau pengetesan entitas-entitas untuk pemenuhan dan konsistensi dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan. Validasi adalah proses melihat kebenaran sistem, apakah proses yang telah ditulis dalam spesifikasi apakah sesuai dengan yang sebenarnya diinginkan dan dibutuhkan oleh pengguna. Sedangkan dengan deteksi *erro*, *testing* berorientasi untuk membuat kesalahan secara intensif, untuk menentukan apakah suatu hal tersebut terjadi bilamana tidak seharusnya terjadi atau suatu hal tersebut tidak terjadi dimana seharusnya mereka ada.

Testing merupakan suatu aktivitas menyeluruh dan terus menerus sepanjang proses pengembangan. Hal ini dilakukan agar informasi yang dibutuhkan dapat dikumpulkan untuk melakukan evaluasi efektivitas kerja. Tiap aktivitas yang digunakan dengan obyektif untuk membantu proses evaluasi suatu *software* dapat disebut sebagai suatu aktivitas *testing*.

Test case merupakan suatu tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi, maupun hasil yang telah ditentukan sebelumnya. Ada dua jenis *test case* yang dapat digunakan, yakni *black box testing* untuk melakukan *testing* kesesuaian suatu komponen terhadap spesifikasi dan *white box testing* yang digunakan untuk melakukan *testing* kesesuaian komponen terhadap desain.

White box testing adalah suatu metode desain *test case* yang menggunakan struktur kendali dari desain procedural. Metode desain *test case* ini dapat menjamin semua jalur (*path*) yang independen dapat dites setidaknya sekali tes, semua logika keputusan dapat dites dengan jalur yang salah dan atau jalur yang benar, semua *loop* dapat dites terhadap batasannya dan ikatan operasionalnya, serta semua struktur internal data dapat dites untuk memastikan validitasnya. Sedangkan *black box testing* dilakukan tanpa pengetahuan detil struktur internal dari sistem atau komponen yang dites. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan spesifikasi kebutuhan dari *software*. Kategori *error* yang akan diketahui melalui *black box testing* antara lain: fungsi yang hilang atau tidak benar, *error* dari antarmuka, *error* dari struktur data atau akses eksternal database, *error* dari kinerja atau tingkah laku, *error* dari inisialisasi dan terminasi.

2.21 *Black box Testing*

Black box testing, dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang ditest, juga disebut sebagai *behavioral testing*, *specification-based testing*, *input / output testing* atau *functional testing*. *Black box testing* berfokus pada kebutuhan fungsional pada software, berdasarkan pada spesifikasi kebutuhan dari *software*. Kategori *error* yang akan diketahui melalui *black box testing* adalah sebagai berikut:

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

Test di desain untuk menjawab pertanyaan sebagai berikut:

1. Bagaimana validasi fungsi yang akan ditest ?
2. Bagaimana tingkah laku kinerja dari sistem yang akan ditest ?
3. Kategori masukan apa saja yang bagus digunakan untuk test case ?
4. Apakah sebagian sistem sensitif terhadap suatu nilai masukan tertentu ?
5. Bagaimana batasan suatu kategori masukan ditetapkan ?
6. Sistem mempunyai toleransi jenjang dan volume data apa saja ?
7. Apa saja akibat dari kombinasi data tertentu yang akan terjadi pada operasi dari sistem ?