

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Sistem Informasi

Sistem adalah serangkaian subsistem yang saling terikat dan tergantung satu sama lain, bekerja bersama-sama untuk mencapai tujuan dan sasaran yang sudah ditetapkan sebelumnya. Semua sistem memiliki *input*, proses, *output* dan umpan balik. [1]

Informasi adalah data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Fungsi informasi adalah memberikan suatu dasar kemungkinan untuk menanggapi seleksi kepada pengambil keputusan dengan mengurangi kenakeragaman dan ketidakpastian sehingga dapat diambil suatu keputusan yang baik, memberikan standar-standar, aturan ukuran, dan aturan-aturan keputusan untuk penentu dan penyebaran tanda-tanda kesalahan dan umpan balik guna mencapai tujuan kontrol. [2]

Sistem Informasi adalah suatu sistem di dalam suatu 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. [3]

Komponen-komponen sistem informasi meliputi [4] :

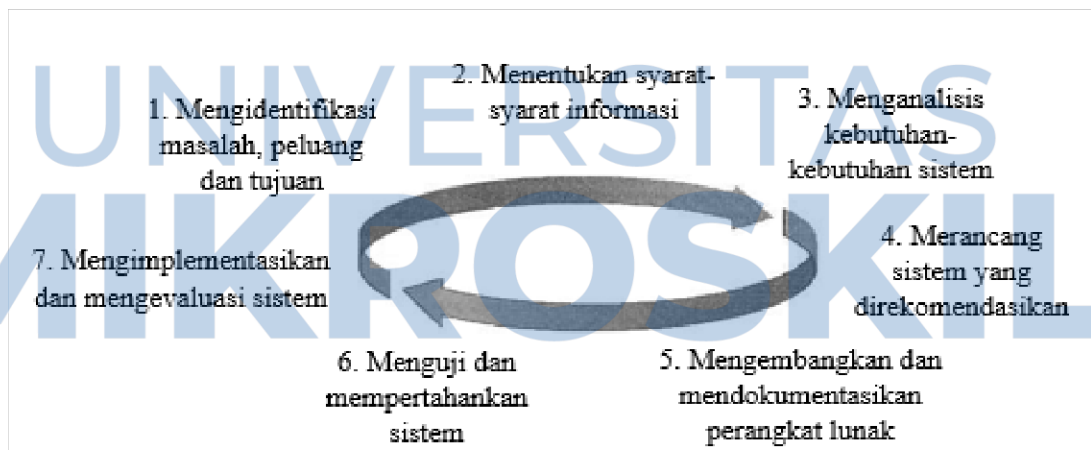
1. Sumber data manusia (*Brainware*), yaitu :
 - a. Para pakar, yang meliputi sistem analisis, pembuat *software*, dan operator sistem.
 - b. Pemakai akhir, yang meliputi orang-orang lainnya yang menggunakan sistem informasi.
2. Sumber daya perangkat keras (*hardware*), yaitu :
 - a. Mesin, yang meliputi komputer, monitor, video, *diskdrive* magnetis, *printer* dan lainnya.
 - b. Media, yang meliputi *floppy disk*, *magnetic tape*, *disk optical*, kartu *plastic* formulir kertas, dan lainnya.
3. Sumber daya perangkat lunak (*software*), yaitu program yang meliputi prosedur *entry data*, prosedur memperbaiki kesalahan, prosedur pendistribusian cek gaji, dan lainnya.

4. Sumber daya data, yaitu deskripsi produk, catatan pelanggan, *file* kepegawaian, *database* persediaan dan lainnya.
5. Sumber daya jaringan, yaitu media komunikasi, pemrosesan komunikasi, serta *software* untuk mengakses dan mengendalikan jaringan lainnya.

2.2 Siklus Hidup Pengembangan Sistem

Siklus hidup pengembangan sistem atau *System Development Life Cycle* (SDLC) adalah pendekatan yang dilakukan melalui beberapa tahap untuk menganalisis dan merancang sistem dimana sistem tersebut telah dikembangkan dengan sangat baik melalui penggunaan siklus kegiatan penganalisis dan pemakai secara spesifik. [1]

Penganalisis tidak sepakat dengan berapa banyak tahap yang ada di dalam Siklus hidup pengembangan sistem, namun mereka umumnya memuji pendekatan terorganisir mereka. SDLC dibagi atas tujuh tahap. Meskipun masing-masing tahap ditampilkan secara terpisah, namun tidak pernah tercapai sebagai satu langkah terpisah. Melainkan, beberapa aktivitas muncul secara simultan, dan aktivitas tersebut dilakukan secara berulang-ulang. Lebih berguna lagi memikirkan bahwa SDLC bisa dicapai dalam tahap-tahap (dengan aktivitas berulang yang saling tumpang tindih satu sama lainnya dan menuju ke tujuan terakhir) dan tidak dalam langkah-langkah terpisah. [1]



Gambar 2. 1 Siklus Hidup Pengembangan Sistem

Dari gambar diatas, Siklus hidup pengembangan sistem dapat diuraikan dan dijelaskan sebagai berikut [1] :

a. Mengidentifikasi Masalah, Peluang dan Tujuan

Penganalisis mengidentifikasi masalah, peluang dan tujuan-tujuan yang hendak dicapai. Tahap ini sangat penting bagi keberhasilan proyek, karena tidak seorangpun yang ingin membuang-buang waktu kalau tujuan masalah yang keliru. Penganalisis harus menemukan

apa yang sedang dilakukan dalam bisnis lalu melihat beberapa aspek dalam aplikasi-aplikasi sistem informasi untuk membantu bisnis supaya mencapai tujuan-tujuannya dengan menyebut *problem* atau peluang-peluang tertentu. Aktivitas dalam tahap ini meliputi wawancara terhadap manajemen pemakai, menyimpulkan pengetahuan yang diperoleh, mengestimasi cakupan proyek, dan mendokumentasikan hasil-hasilnya.

b. Menentukan Syarat-Syarat Informasi

Diantara perangkat-perangkat yang dipergunakan untuk menetapkan syarat-syarat informasi di dalam bisnis diantaranya ialah menentukan sampel dan memeriksa data mentah, wawancara, mengamati perilaku pembuat keputusan dan lingkungan kantor dan *prototyping*. Tahap ini membentuk gambaran mengenai organisasi dan tujuan-tujuan yang dimiliki seorang penganalisa. Penganalisa akan bisa memahami fungsi-fungsi bisnis dan melengkapi informasi tentang masyarakat, tujuan, data, dan prosedur yang terlibat.

c. Menganalisis Kebutuhan Sistem

Perangkat dan teknik-teknik tertentu akan membantu penganalisis menentukan kebutuhan. Perangkat tersebut ialah penggunaan *Data Flow Diagram* untuk menyusun daftar input, proses, dan output fungsi bisnis dalam bentuk grafik terstruktur. Penganalisis juga menganalisis keputusan terstruktur yang dibuat yaitu keputusan dimana kondisi, kondisi alternatif, tindakan serta aturan tindakan ditetapkan. Selain itu, penganalisis menyiapkan suatu proposal sistem yang berisikan ringkasan apa yang ditemukan, analisis biaya atau keuntungan alternatif yang tersedia, serta rekomendasi apa yang harus dilakukan.

d. Merancang Sistem yang Direkomendasikan

Penganalisis merancang prosedur *data-entry* sedemikian rupa sehingga data yang dimasukkan kedalam sistem informasi benar-benar akurat. Selain itu, penganalisis menggunakan teknik-teknik bentuk dan perancangan layar tertentu untuk menjamin keefektifan input sistem informasi. Tahap ini juga mencakup perancangan *file-file* atau basis data yang bisa menyimpan data-data yang diperlukan oleh pembuat keputusan. Penganalisis harus merancang prosedur-prosedur *back up* dan kontrol untuk melindungi sistem dan data serta untuk membuat paket-paket spesifikasi program bagi pemrogram.

e. Mengembangkan dan Mendokumentasikan Perangkat Lunak

Selama tahap ini, penganalisis bekerjasama dengan pemakai untuk mengembangkan dokumentasi perangkat lunak yang efektif, mencakup melakukan prosedur secara manual, bantuan *online* dan *website* yang dikirimkan bersama-sama dengan perangkat lunak baru. Kegiatan dokumentasi menunjukkan kepada pemakai tentang cara penggunaan perangkat lunak dan apa yang harus dilakukan bila perangkat lunak mengalami masalah.

f. Menguji dan Mempertahankan Sistem

Sebagian pengujian dilakukan oleh pemrogram sendiri, dan lainnya dilakukan oleh penganalisis sistem. Rangkaian pengujian ini pertama-tama dijalankan bersama-sama dengan data contoh serta dengan data aktual dari sistem yang telah ada. Sebagian besar prosedur sistematis yang dijalankan penganalisis selama siklus hidup pengembangan sistem membantu memastikan bahwa pemeliharaan bisa dijaga sampai tingkat minimum.

g. Mengimplementasikan dan Mengevaluasi Sistem

Tahap ini melibatkan pelatihan bagi pemakai untuk mengendalikan sistem. Sebagian pelatihan tersebut dilakukan oleh vendor, namun kesalahan pelatihan merencanakan konversi dari sistem lama ke sistem baru. Proses ini mencakup pengubahan *file-file* dari *format* lama ke *format* baru atau membangun suatu basis data, menginstal peralatan, dan membawa sistem baru untuk diproduksi. Ketika penganalisis menyelesaikan suatu tahap pengembangan sistem akan berlanjut ke tahap berikutnya, penemuan suatu masalah bisa memaksa penganalisis kembali ke tahap sebelumnya dan memodifikasi pekerjaannya di tahap tersebut.

2.3 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah salah satu *tool* utama yang digunakan untuk memodelkan proses, di mana diagram ini menggambarkan proses usulan dan/atau yang sudah ada di dalam sebuah sistem bersamaan dengan input, output dan data di dalamnya. Model DFD menunjukkan aliran data antar proses beserta tempat penyimpanan data tersebut. Dengan kata lain, model proses ini juga berlaku sebagai cetak biru dari proses bisnis untuk diimplementasikan dan dapat berlaku sebagai perangkat lunak yang dapat dibeli atau dibangun [5].

Berikut ini adalah aturan dasar dalam menggambarkan DFD yang perlu diikuti [1]:

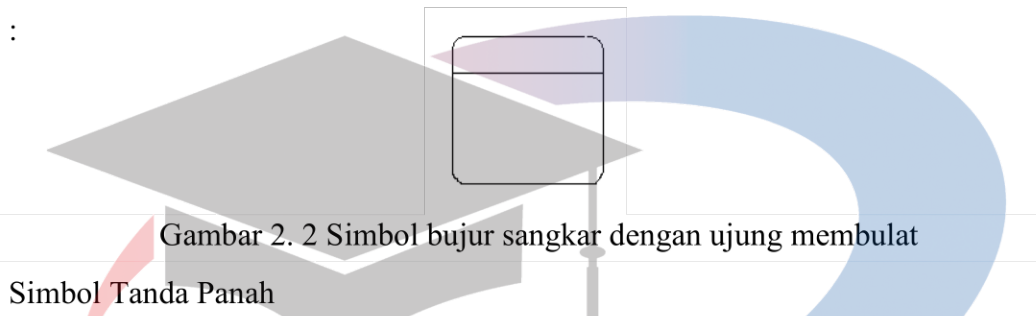
1. Sebuah DFD harus mempunyai paling sedikit satu proses, dan tidak boleh ada satu objek yang berdiri sendiri atau objek yang terhubung pada dirinya sendiri.
2. Sebuah proses harus menerima paling sedikit satu aliran data menuju proses dan meneruskan paling sedikit satu aliran data keluar dari proses.
3. Sebuah *data store* harus terhubung ke paling sedikit satu proses.
4. Entitas eksternal tidak boleh terhubung satu sama lain. Meskipun entitas eksternal berkomunikasi secara bebas, komunikasi tersebut bukan bagian dari sistem yang dirancang menggunakan DFD.

Ada 4 simbol dasar yang digunakan untuk memetakan gerakan DFD [1] :

1. Simbol bujur sangkar dengan sudut membulat

Simbol bujur sangkar dengan sudut membulat digunakan untuk menunjukkan adanya proses transformasi. Proses-proses tersebut selalu menunjukkan suatu perubahan data, jadi aliran data yang meninggalkan suatu proses di beri label yang berbeda dari aliran data yang masuk. Proses-proses yang menunjukkan hal itu di dalam sistem harus diberi nama yang jelas untuk memudahkan memahami proses apa yang sedang dilakukan.

Bentuk simbol bujur sangkar dengan ujung membulat dapat di lihat pada gambar 2.2 berikut

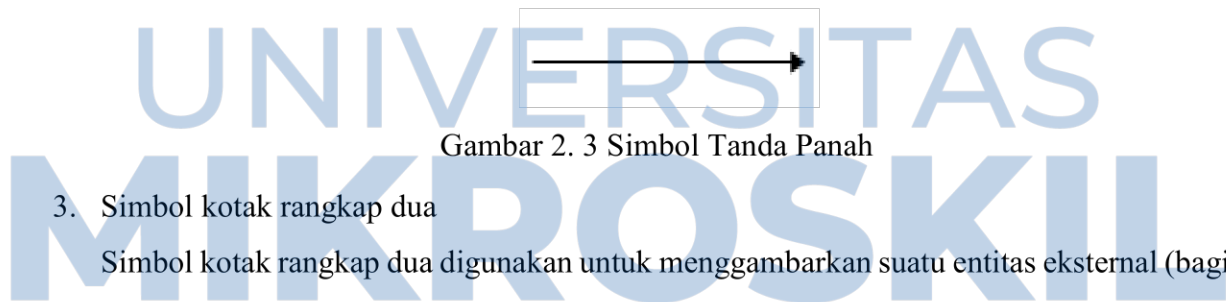


Gambar 2. 2 Simbol bujur sangkar dengan ujung membulat

2. Simbol Tanda Panah

Tanda panah menunjukkan perpindahan data dari suatu titik ke titik yang lain, dengan kepala tanda panah, mengarah ke tujuan data. Aliran data yang muncul secara simultan biasa digambarkan hanya menggunakan tanda paralel, karena sebuah panah menunjukkan seseorang, tempat, atau sesuatu, maka harus digambarkan dalam kata benda.

Bentuk simbol tanda panah dapat di lihat pada gambar 2.3 berikut.



Gambar 2. 3 Simbol Tanda Panah

3. Simbol kotak rangkap dua

Simbol kotak rangkap dua digunakan untuk menggambarkan suatu entitas eksternal (bagian lain, sebuah perusahaan, seseorang atau sebuah mesin) yang dapat mengirim data atau menerima data dari sistem. Entitas eksternal, atau hanya entitas, disebut juga sumber atau tujuan data, dan di anggap eksternal terhadap sistem yang sedang digambarkan. Setiap entitas diberi label dengan nama yang sesuai. Meskipun berinteraksi dengan sistem, namun di anggap di luar batas-batas sistem. Entitas-entitas tersebut harus diberi nama dengan suatu kata benda. Entitas yang sama bisa digunakan lebih dari sekali atas suatu *Data Flow Diagram* tertentu untuk menghindari persilangan antara jalur-jalur aliran data.

Bentuk simbol kotak rangkap dua dapat di lihat pada gambar 2.4 berikut.

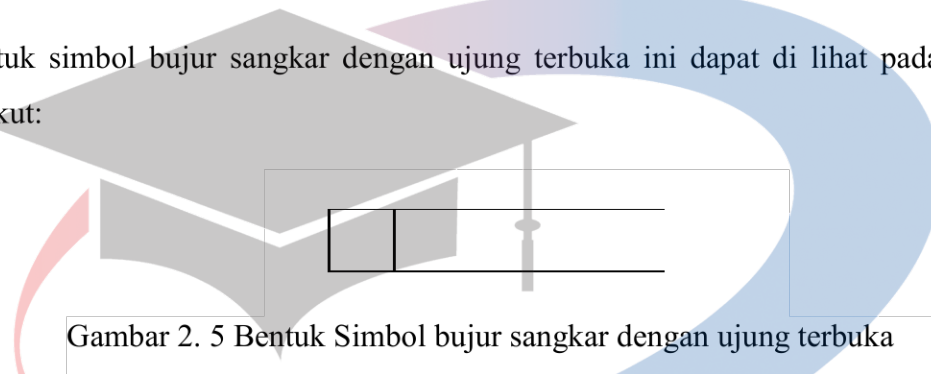


Gambar 2. 4 Simbol kotak rangkap dua

4. Simbol bujur sangkar dengan ujung terbuka

Simbol bujur sangkar dengan ujung terbuka, yang menunjukkan penyimpanan data, digambarkan dengan dua garis paralel yang ditutup oleh sebuah garis pendek di sisi kiri dan ujungnya terbuka di sisi sebelah kanan. Simbol-simbol ini digambarkan secukupnya sehingga memungkinkan menandai bentuk huruf-huruf di antara garis-garis paralel yang ada.

Bentuk simbol bujur sangkar dengan ujung terbuka ini dapat di lihat pada gambar 2.5 berikut:

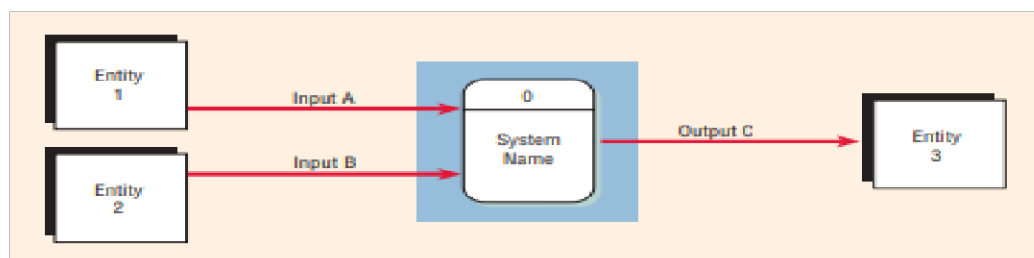


Gambar 2. 5 Bentuk Simbol bujur sangkar dengan ujung terbuka

DFD dapat dimanfaatkan untuk menjelaskan sistem pada tingkat kedetilan yang berbeda. Adapun tahapan-tahapan DFD untuk menjelaskan tingkat kedetilan dari sistem [1]:

1. Diagram Konteks

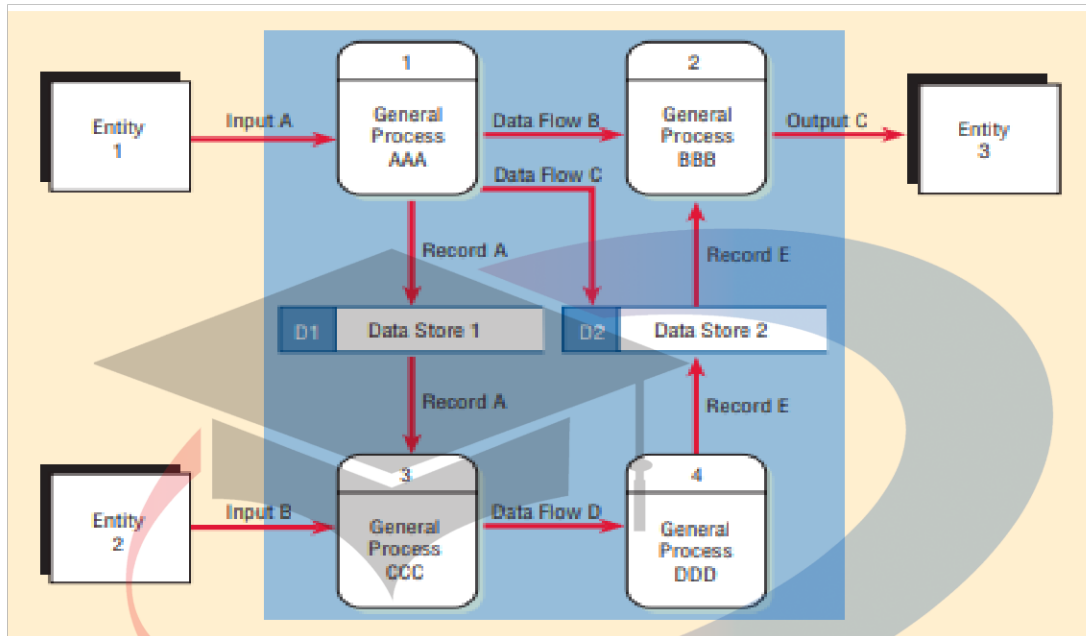
Diagram konteks merupakan tahap tertinggi dari sebuah DFD. Diagram ini hanya terdiri dari satu proses yang mewakili sistem (diberi nomor 0). Diagram ini menampilkan semua entitas luar dan aliran data yang mengalir menuju ke dan dari entitas luar tersebut. Pada tahap ini, simpanan data tidak ditampilkan.



Gambar 2. 6 Diagram Konteks

2. DFD level 0

DFD level 0 menjabarkan proses-proses apa saja yang harus dilakukan dalam sistem, tetapi kita perlu menghindari terlalu banyak proses dalam DFD tahap ini. Kita dapat melakukan ekspansi terhadap proses untuk menunjukkan kedetilan sistem di DFD level selanjutnya. Pada tahap ini penyimpanan data sudah dapat ditampilkan.



Gambar 2. 7 Diagram level 0

3. DFD level 1 ... N

Pada tahap penggambaran DFD level 1 ... N, kita mengekspansi setiap proses dari DFD level di atasnya. Pada tahap ini entitas luar tidak perlu ditampilkan lagi, kita menggambarkan proses-proses yang terlibat di dalam sistem dengan semakin rinci, menampilkan aliran data yang mengalir dan penyimpanan data.

2.4 Kamus Data

Kamus data adalah suatu aplikasi khusus dari jenis kamus-kamus yang digunakan sebagai referensi kehidupan setiap hari. Kamus data merupakan hasil referensi data mengenai data, suatu data yang disusun penganalisis sistem untuk membimbing mereka selama melakukan analisis dan desain. Sebagai suatu dokumen, kamus data mengumpulkan dan mengkoordinasi istilah-istilah data tertentu, dan menjelaskan apa arti dari setiap istilah yang ada. [1]

Sekalipun kamus data juga memuat informasi mengenai data dan prosedur-prosedur, kumpulan informasi mengenai proyek dalam jumlah besar disebut gudang. Konsep gudang adalah salah satu dari berbagai pengaruh perangkat CASE dan bisa berisikan hal-hal sebagai berikut [1] :

1. Informasi mengenai data-data yang dipertahankan oleh sistem, meliputi aliran data, simpanan data, struktur record dan elemen data.
2. Logika prosedural
3. Desain layar dan laporan
4. Keterkaitan data
5. Penyampaian syarat-syarat proyek dan sistem final.
6. Informasi manajemen proyek

Struktur data biasanya digambarkan menggunakan notasi aljabar. Metode ini memungkinkan penganalisis membuat suatu gambaran mengenai elemen-elemen yang membentuk struktur data bersama-sama dengan informasi-informasi mengenai elemen-elemen tersebut. Sebagai contoh, penganalisis akan menunjukkan apakah ada beberapa elemen yang sama di dalam struktur data tersebut (kelompok berulang) atau apakah dua elemen saling berpisah satu sama lain. Notasi aljabar menggunakan simbol-simbol sebagai berikut [1] :

1. Tanda sama dengan (=), artinya "terdiri dari".
2. Tanda plus (+), artinya "dan".
3. Tanda kurung { }, menunjukkan elemen-elemen repetitif, juga disebut kelompok berulang atau tabel-tabel. Kemungkinan bisa ada satu atau beberapa elemen berulang di dalam kelompok tersebut. Kelompok berulang bisa mengandung keadaan-keadaan tertentu seperti misalnya, jumlah pengulangan yang pasti atau batas tertinggi dan batas terendah untuk jumlah pengulangan.
4. Tanda kurung [], menunjukkan salah satu dari dua situasi tertentu. Satu elemen bisa ada sedangkan elemen lainnya juga ada, tetapi tidak bisa kedua-duanya ada secara bersamaan. Elemen-elemen yang ada di dalam tanda kurung ini saling terpisah satu sama lain.
5. Tanda kurung (), menunjukkan suatu elemen yang bersifat pilihan. Elemen-elemen yang bersifat pilihan ini bisa dikosongkan pada layar masukan atau bisa juga dengan memuat spasi atau nol untuk *field-field* numerik pada struktur *file*.

2.5 Basis Data

Basis Data adalah sumber data yang caranya dipakai oleh banyak pemakai untuk berbagai aplikasi. Inti dari basis data adalah *database management system* (DBMS), yang membolehkan pembuatan, modifikasi, dan pembaharuan basisdata; mendapatkan kembali data; dan membangkitkan laporan. Orang yang memastikan bahwa basisdata memenuhi tujuannya disebut administrator basis data [1]. Tujuan basis data yang efektif termuat di bawah ini [1]:

1. Memastikan bahwa data dapat dipakai di antara pemakai untuk berbagai aplikasi.

2. Memelihara data baik keakuratan maupun kekonsistennannya.
3. Memastikan bahwa semua data yang diperlukan untuk aplikasi sekarang dan yang akan datang akan disediakan dengan cepat.
4. Membolehkan basis data untuk berkembang dan kebutuhan pemakai untuk berkembang.
5. Membolehkan pemakai untuk membangun pandangan personalnya tentang data tanpa memperhatikan cara data disimpan secara fisik..

Keuntungan basis data [1]:

1. Data dibagi pakai berarti data hanya perlu disimpan satu kali saja.
2. Ketika user memerlukan data tertentu, database yang didesain dengan baik akan mengantisipasi kebutuhan dari data tersebut.
3. Menyediakan fasilitas kepada user untuk melihat data.
4. User tidak perlu memikirkan struktur basisdata atau penyimpanan fisiknya.

Kelemahan basis data [1] :

1. Semua data tersimpan pada satu tempat, maka perlu sering di *backup*.
2. Dari segi efisiensi maka:
 - Memerlukan cukup banyak waktu untuk menyisip, *update*, hapus, dan *retrieve*.
 - Memerlukan biaya untuk menyediakan tempat penyimpanannya.

2.6 Normalisasi

Normalisasi adalah transformasi tinjauan pemakai yang kompleks dan data tersimpan kesekumpulan bagian-bagian struktur data yang kecil dan stabil. Di samping menjadi lebih sederhana dan lebih stabil, struktur data yang dinormalisasikan lebih mudah diatur daripada struktur data lainnya. [1]

Dimulai dengan tiap sebuah pandangan atau data tersimpan yang dikembangkan untuk suatu kamus data, penganalisis menormalisasikan struktur data dalam tiga tahap. Setiap tahap meliputi prosedur yang sangat penting yang mensesederhanakan struktur data.

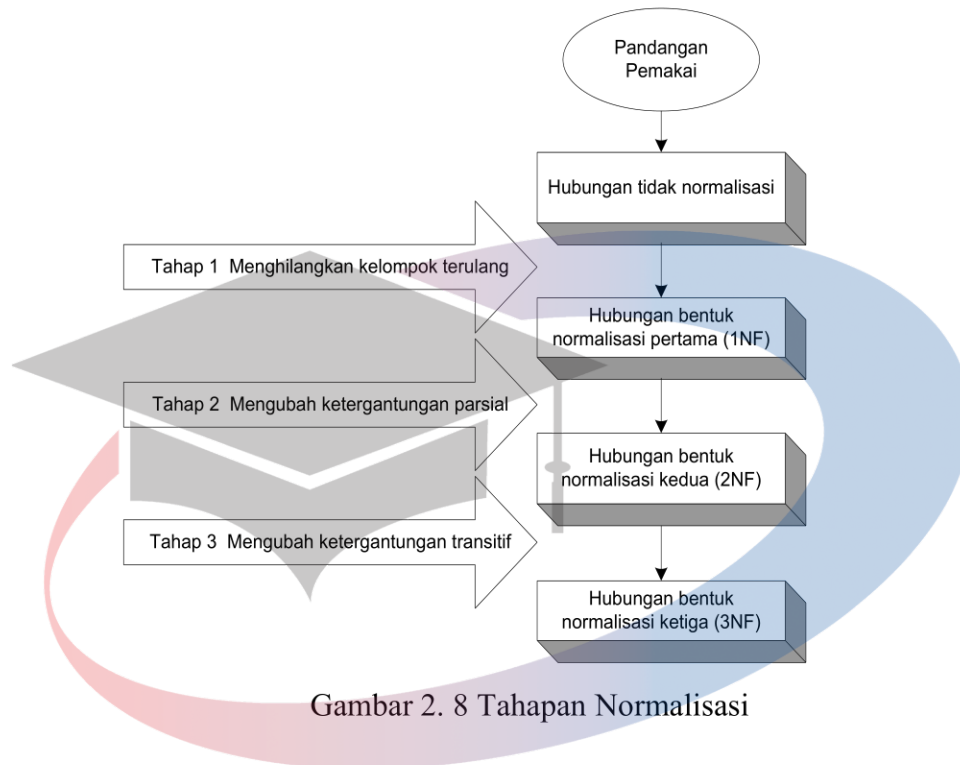
Hubungan diperoleh dari tinjauan pemakai atau data tersimpan sebagian besar akan menjadi tidak normal. Tahap pertama dari proses meliputi menghilangkan semua kelompok terulang dan mengidentifikasi kunci utama. Untuk mengerjakannya, hubungan perlu dipecah ke dalam dua atau lebih hubungan. Pada titik ini, hubungan mungkin sudah menjadi bentuk normalisasi ketiga, bahkan lebih banyak tahap akan diperlukan untuk mentransformasi hubungan ke bentuk normalisasi ketiga.

Tahap kedua menjamin bahwa semua atribut bukan kunci sepenuhnya tergantung pada kunci utama. Semua ketergantungan parsial diubah dan diletakkan dalam hubungan lain.

Tahap ketiga mengubah ketergantungan transitif manapun. Suatu ketergantungan transitif adalah sesuatu dimana atribut bukan kunci tergantung pada atribut bukan kunci lainnya.

[1]

Berikut menunjukkan hubungan dari ketiga tahapan normalisasi:



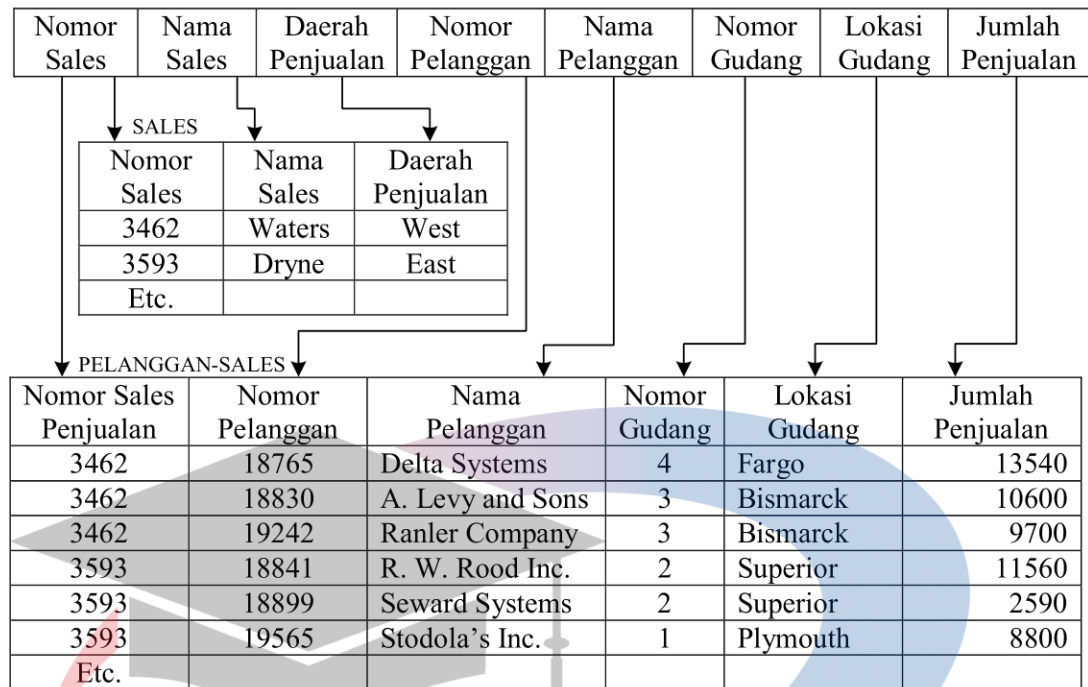
Gambar 2. 8 Tahapan Normalisasi

Tahapan normalisasi yaitu [1]:

1. Bentuk Normalisasi Pertama [1NF]

Tahap pertama dari proses meliputi menghilangkan semua kelompok terulang dan mengidentifikasi kunci utama. Untuk mengerjakannya, hubungan perlu dipecah ke dalam dua atau lebih hubungan. Contoh dari proses normalisasi tahapan pertama dapat dilihat pada gambar 2.9 berikut:

LAPORAN-PENJUALAN

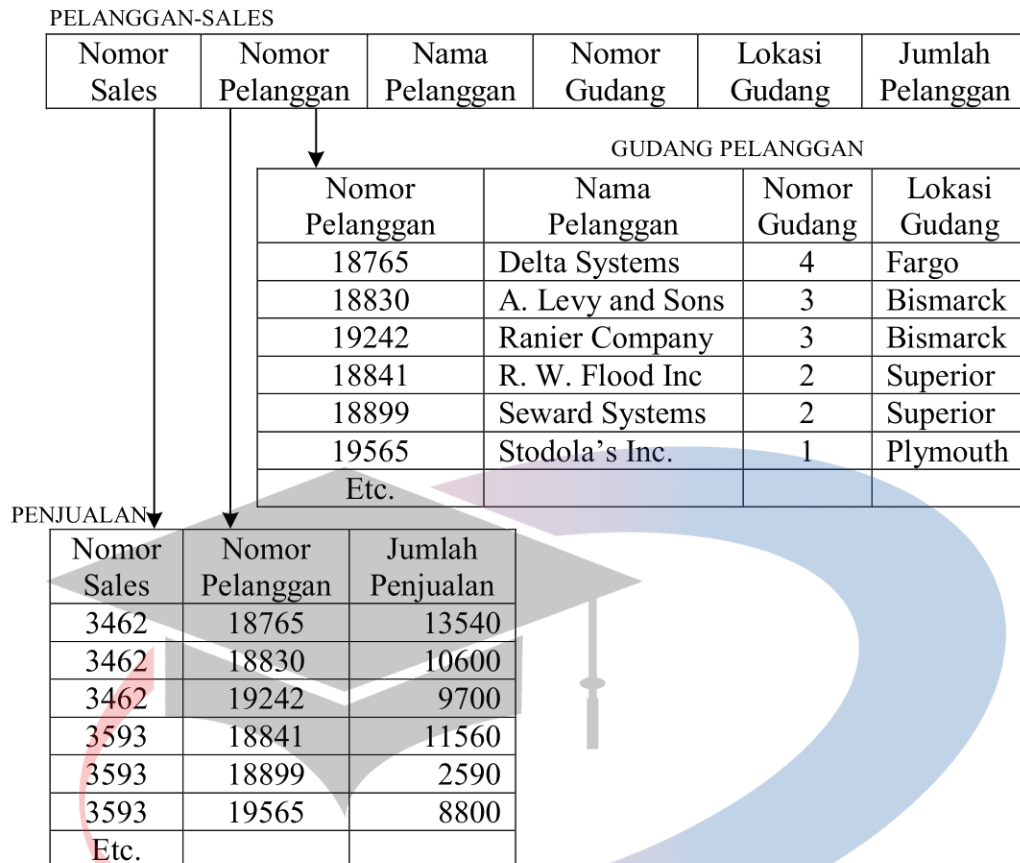


Gambar 2. 9 Contoh bentuk normalisasi pertama (1NF)

Hubungan tidak normal yang asli dari LAPORAN-PENJUALAN dipisah ke dalam dua hubungan, SALES (3NF) dan PELANGGAN-SALES (1NF)

2. Bentuk Normalisasi Kedua [2NF]

Tahap kedua menjamin bahwa semua atribut bukan kunci sepenuhnya tergantung pada kunci utama. Semua ketergantungan parsial diubah dan diletakkan dalam hubungan lain. Contoh dari proses normalisasi tahapan kedua dapat dilihat pada gambar 2.10 berikut:

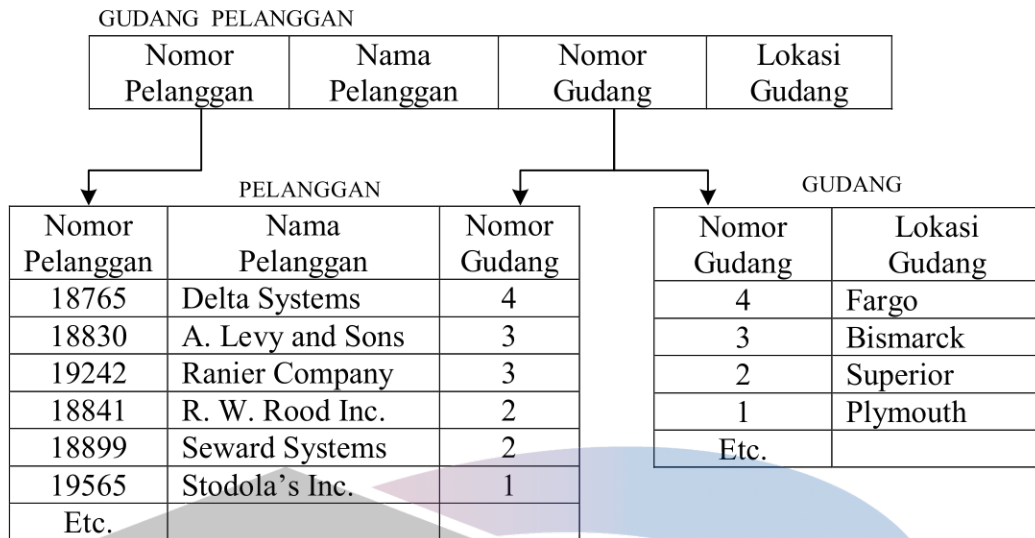


Gambar 2. 10 Contoh bentuk normalisasi kedua (2NF)

Hubungan PELANGGAN-SALES dipisah ke dalam hubungan yang dinamakan GUDANG PELANGGAN (2NF) dan hubungan yang dinamakan PENJUALAN (1NF)

3. Bentuk Normalisasi Ketiga [3NF]

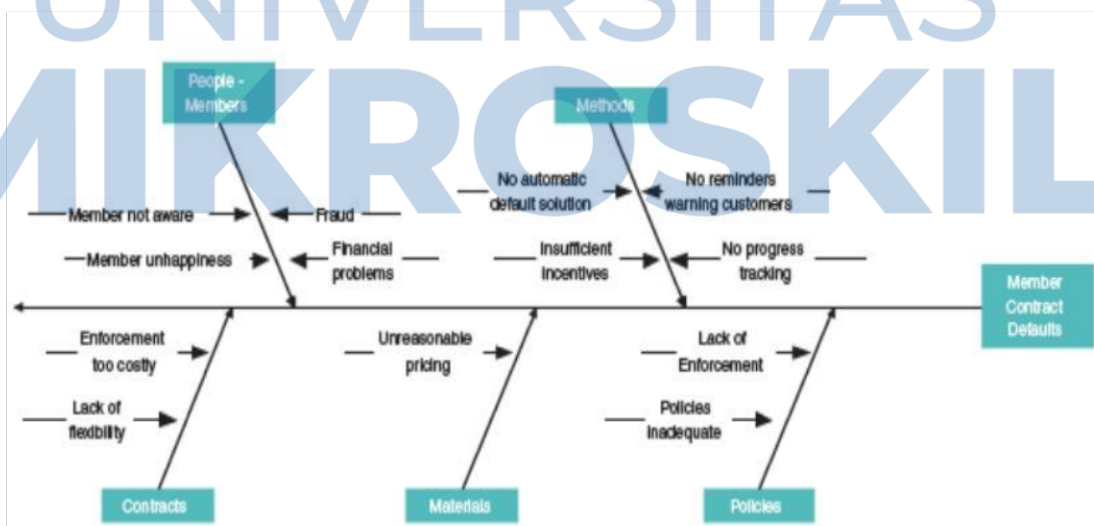
Tahap ketiga mengubah ketergantungan transitif dimana semua atribut bukan kunci sepenuhnya tergantung secara fungsional pada kunci utama dan tidak terdapat ketergantungan bukan kunci. Contoh dari proses normalisasi tahapan ketiga dapat di lihat pada gambar 2.11 berikut:



Gambar 2. 11 Contoh bentuk normalisasi ketiga (3NF)
 Hubungan GUDANG-PELANGGAN dipisah ke dalam dua hubungan yang dinamakan
 PELANGGAN (1NF) dan GUDANG (1NF)

2.7 Diagram Sebab -Akibat (Diagram *Fishbone* / Diagram Ishikawa)

Diagram Ishikawa adalah diagram yang digunakan untuk mengidentifikasi, mengeksplorasi dan menggambarkan masalah serta sebab dan akibat dari masalah tersebut. Diagram ini sering disebut sebagai diagram sebab-akibat atau diagram *fishbone* karena bentuknya yang menyerupai tulang ikan [5].




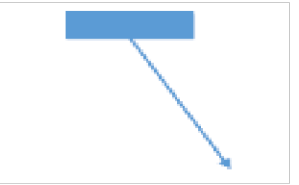
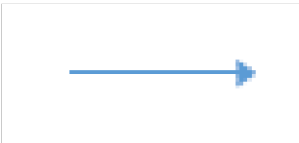
Gambar 2. 12 Contoh Diagram Ishikawa

Diagram *fishbone* pertama kali dicetuskan oleh Kaoru Ishikawa, yang membuat terobosan mengenai proses manajemen kualitas di galangan kapal Kawasaki, Jepang, dan dalam prosesnya, Ishikawa menjadi salah satu pencetus manajemen modern [5].

Penggambaran diagram *fishbone* dimulai dengan menentukan nama masalah yang digambarkan pada sebelah kanan diagram atau kepala ikan. Kemudian, penyebab-penyebab masalah tersebut digambarkan sebagai tulang ikan disepanjang tulang belakang, dengan setiap sebab masalah digambarkan dalam sebuah anak panah yang menunjuk ke arah tulang belakang. Biasanya, “tulang” ini dilabeli empat kategori utama yaitu *materials* (barang), *machines* (mesin), *manpower* (tenaga manusia) dan *methods* (Metode), yang disebut sebagai *the four Ms*. Kategori alternatif atau kategori tambahan mencakup *places*, *procedures*, *policies* dan *people* (4P) atau *surroundings*, *suppliers*, *systems* dan *skills* (4S). Kuncinya adalah memiliki tiga sampai enam kategori utama untuk mencakup semua area kemungkinan masalah. Setelah semua tulang-tulang lengkap maka *fishbone* akan menggambarkan semua kemungkinan yang bisa menjadi akar penyebab dari masalah. Kemudian pengembang bisa menggunakan diagram tersebut untuk menentukan pokok permasalahannya dan bagaimana masalah tersebut bisa diatasi [5].

Tabel 2.1 menunjukkan notasi pada diagram *fishbone*.

Tabel 2. 1 Notasi diagram fishbone

NO	Notasi	Keterangan
1		<i>Effect</i> , merupakan simbol yang menunjukkan akibat dari masalah yang digambarkan pada diagram <i>fishbone</i> dan akibat tersebut digambarkan di sebelah kanan anak panah.
2		<i>Category</i> , menunjukkan kategori dari masalah yang digambarkan pada diagram <i>fishbone</i> , dimana anak panah simbol ini menunjuk langsung ke anak panah yang mengarah ke <i>effect</i> .
3		<i>Cause</i> , merupakan sebab dari masalah yang digambarkan pada diagram <i>fishbone</i> , dimana anak panah simbol ini menunjuk anak panah simbol <i>category</i> .

2.8 Flowchart

Sebuah *flowchart* adalah representasi grafikal dari sebuah sistem yang menjelaskan relasi fisik diantara entitas-entitas kuncinya. *Flowchart* dapat digunakan untuk menyajikan kegiatan manual, kegiatan pemrosesan komputer atau keduanya. Sebuah *flowchart* dokumen digunakan untuk menggambarkan elemen-elemen dari sebuah sistem manual, termasuk *record-record* akuntansi (dokumen, jurnal, buku besar dan *file*). Departemen organisasional yang terlibat dalam proses dan kegiatan-kegiatan (baik klerikal maupun fiskal) yang dilakukan dalam departemen tersebut. [5]

Adapun simbol-simbol dari *Flowchart Document* [5] :

Tabel 2. 2 Simbol *Flowchart Document*

Simbol	Keterangan
	Catatan Akuntansi (jurnal, <i>Register</i> , Catatan, buku besar).
	Proses.
	Garis Arus.
	Deskripsi Proses dan Komentar.
	Operasi Manual.
	Dokumen sumber atau laporan.
	Terminal menunjukkan sumber atau tujuan dari dokumen atau laporan.
	Konektor halaman.
	Konektor <i>off-page</i> .
	<i>File</i> untuk menyimpan dokumen sumber penyimpanan dan laporan.

2.9 PIECES

Kerangka kerja PIECES dikembangkan oleh James Wetherbe, yang ditunjukkan untuk mengklasifikasi masalah. Sebutannya PIECES, dikarenakan kerangka kerja tersebut merupakan singkatan dari enam kategori masalah yang diidentifikasi. Masing-masing kepanjangan dari PIECES yaitu [5]:

1. P → *performance* (kinerja)
2. I → *information* (informasi)
3. E → *economics* (ekonomi)
4. C → *control* (pengendalian)
5. E → *efficiency* (efisiensi)
6. S → *service* (layanan)

Kerangka kerja PIECES juga sering digunakan dalam penentuan kebutuhan sistem. Tabel dibawah ini menjabarkan hal-hal yang harus diperhatikan pada setiap kategori dalam PIECES, ketika akan menentukan kebutuhan sistem. Kebutuhan tersebut sifatnya nonfungsional [5].

Tabel 2. 3 Klasifikasi PIECES dalam penentuan kebutuhan sistem

Jenis Kebutuhan nonfungsional	Penjelasan
<i>Performance</i> (kinerja)	Kebutuhan kinerja merepresentasikan bahwa kinerja sistem dibutuhkan untuk memenuhi kebutuhan para pengguna. Contoh masalah yang dapat dikategorikan pada kinerja adalah waktu respon.
<i>Information</i> (informasi)	Kebutuhan informasi merepresentasikan informasi yang sesuai dengan kebutuhan pengguna dalam hal isi, kronologis, akurasi dan format. Contoh masalah yang dapat dikategorikan pada informasi adalah masukan dan keluaran yang dibutuhkan, atau seberapa update informasi yang diberikan.
<i>Economics</i> (ekonomi)	Kebutuhan ekonomi merepresentasikan kebutuhan untuk mengurangi biaya atau menambah pendapatan. Contoh masalah yang dapat dikategorikan pada ekonomi adalah area sistem yang biayanya perlu dikurangi, atau batasan anggaran yang harus dikeluarkan

<p><i>Control</i> (pengendalian)</p>	<p>Kebutuhan pengendalian merepresentasikan lingkungan dimana sistem harus beroperasi, bersama dengan jenis dan tingkat keamanan yang harus diberikan. Contoh masalah yang dikategorikan pada pengendalian adalah kebutuhan privasi yang diperlukan atau pembatasan akses terhadap sistem atau informasi</p>
<p><i>Efficiency</i> (efisiensi)</p>	<p>Kebutuhan efisiensi merepresentasikan kemampuan sistem untuk menghasilkan keluaran dengan "usaha" sedikit. Contoh masalah yang dapat dikategorikan pada efisiensi adalah apakah terdapat langkah berulang yang harus dihilangkan atau mengurangi pemakaian <i>resource</i> oleh sistem yang berujung pada kinerja yang ringan</p>
<p><i>Service</i> (layanan)</p>	<p>Kebutuhan sistem merepresentasikan kebutuhan pada sistem untuk menjadi dapat dipercaya, fleksibel dan dapat dikembangkan lebih jauh. Contoh masalah yang dapat dikategorikan pada layanan adalah pengguna sistem, jenis-jenis pengguna, atau dokumentasi yang dibutuhkan</p>

2.10 Persediaan

Persediaan barang (*inventory*) merupakan salah satu aktiva lancar yang jumlah cukup besar dan selalu berputar secara terus menerus serta mengalami perubahan pada suatu perusahaan, terutama perusahaan industri [6].

Pada perusahaan industri terdiri dari 3 (tiga) jenis persediaan, yaitu [6]:

1. Persediaan bahan dasar/bahan mentah
2. Persediaan barang dalam proses
3. Persediaan barang jadi

Bagi perusahaan industri atau perusahaan manufaktur, persediaan ini dianggap cukup penting, karena kesalahan dalam menentukan persediaan, akan mengganggu kelancaran operasi perusahaan. Sedangkan perusahaan dagangan hanya satu jenis persediaan yaitu persediaan barang dagangan. Persediaan bahan dasar atau bahan mentah bagi perusahaan industri, adalah persediaan bahan untuk diproses dalam memproduksi barang jadi [6].

Persediaan barang dalam proses adalah barang yang masuk dalam proses produksi yang akhirnya menjadi barang jadi. Sedangkan barang jadi adalah barang yang siap dipakai atau dijual, yang akan memberikan fleksibilitas bagi perusahaan untuk menjadwalkan produksi dan

pemasarannya. Persediaan barang jadi yang cukup juga dapat menjamin efektifitas kegiatan pemasarannya, karena apabila persediaan kurang, berakibat perusahaan akan kehilangan kesempatan untuk merebut pasar. Dengan persediaan barang jadi yang cukup, perusahaan dapat memenuhi permintaan atau pesanan dengan cepat. Tetapi dengan persediaan yang besar akan membawa konsekuensi berupa biaya yang timbul untuk mempertahankan persediaan tersebut. Biaya yang terkait dengan persediaan, adalah biaya pemesanan dan biaya penyimpanan [6]

