

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Data dan Informasi**

##### **2.1.1 Pengertian Data dan Informasi**

Menurut Hoffer, Prescott dan McFadden (2005, p5) Data merupakan gambaran obyek dan peristiwa yang mempunyai arti dan penting di lingkungan pemakai. Sedangkan informasi merupakan data yang telah diproses dengan cara meningkatkan pengetahuan seseorang yang menggunakan data tersebut.

Menurut Turban (2003, p15), data adalah fakta-fakta yang belum diolah atau gambaran-gambaran lebih lanjut dari benda-benda, kejadian-kejadian, kegiatan-kegiatan dan transaksi-transaksi yang ditangkap, direkam, disimpan dan diklarifikasikan, tetapi tidak disusun untuk menyampaikan arti khusus lainnya. Sedangkan Informasi merupakan sebuah kumpulan dari fakta-fakta yang disusun di dalam beberapa cara, jadi kumpulan fakta tersebut bisa berarti bagi penerimanya.

##### **2.1.2 Karakteristik Informasi**

Dalam sebuah sistem informasi, informasi mempunyai beberapa karakteristik. Menurut Romney dan Steinbart (2003, p10) karakteristik informasi yang berguna adalah sebagai berikut :

1. Relevan

Informasi dikatakan relevan apabila informasi itu mengurangi ketidakpastian, meningkatkan kemampuan pembuat keputusan untuk

membuat prediksi, konfirmasi atau membenarkan dugaan mereka sebelumnya

2. Dapat dipercaya

Suatu informasi dapat dipercaya jika bebas dari kesalahan dan akurat mewakili aktifitas dari suatu organisasi

3. Lengkap

Informasi yang lengkap merupakan informasi yang tidak mengurangi aspek penting dari suatu aktifitas yang mendasari pengukuran informasi tersebut

4. Tepat waktu

Informasi yang tepat waktu merupakan informasi yang tersedia tepat pada waktunya sehingga memungkinkan pembuat keputusan untuk membuat suatu keputusan

5. Dapat dimengerti

Informasi dapat dimengerti jika informasi tersebut dipresentasikan dalam format yang benar dan mudah dipahami

6. Dapat diuji

Informasi dapat diuji jika ada dua orang yang mempunyai pengalaman bekerja secara terpisah, tapi masing-masing menghasilkan informasi yang sama

## **2.2 *File-Based System***

Sebelum adanya aplikasi basis data seperti DBMS, penyimpanan data masih menggunakan penyimpanan dalam suatu *file*. Namun aplikasi basis data saat ini

merupakan suatu aplikasi umum yang digunakan sehari-hari. Misalnya peminjaman komik, pembelian barang di pasar swalayan dan lain-lain.

Menurut Connolly (2005, p7), *File-Based System* adalah suatu kumpulan dari program aplikasi yang memberikan servis kepada pemakai untuk membuat suatu laporan. Setiap program dapat mendefinisikan dan mengatur datanya masing-masing. Namun terdapat banyak keterbatasan dari *File-Based System*, antara lain:

1. Data yang terpisah dan terisolasi
2. Data yang terduplikasi
3. Ketergantungan Data
4. Format data yang tidak kompatibel
5. Query yang tetap
6. Tidak dapat mengantisipasi perkembangan program aplikasi

Oleh karena itu, saat penyimpanan data dilakukan dengan menggunakan pendekatan basis data atau DBMS (*Database Management System*).

## **2.3 Basis Data**

### **2.3.1 Pengertian Basis Data**

Menurut Connolly dan Begg (2005, p15) *database* merupakan suatu kumpulan logikal data yang terhubung satu dengan yang lainnya dan deskripsi dari suatu data yang dirancang sebagai informasi yang dibutuhkan.

Menurut Hoffer, Prescott dan McFadden (2005, p4), basis data adalah sebuah kumpulan terorganisir dari data-data yang berhubungan secara logikal. Basis data biasanya dirancang untuk memenuhi kebutuhan-kebutuhan informasi dari *multiple user* dalam sebuah organisasi.

### 2.3.2 Pengertian Sistem Basis Data

Menurut Michael V. Mannino (2004, p7), *database* adalah bahasa dan alat-alat grafis untuk mendefinisikan entiti-entiti, hubungannya, *integrity constraint* dan hak otorisasi.

Menurut Hoffer, Prescott dan McFadden (2005, p27), Basis data mempunyai komponen-komponen di dalamnya yaitu :

1. *Computer-aided software engineering (CASE) tools.*

*CASE tools* merupakan *tools* automata yang digunakan untuk mendesign basis data dan program aplikasi

2. *Repository*

Merupakan pusat penyimpanan definisi data, relasi antar data, format layar dan laporan

3. *Database Management System (DBMS)*

*Software* komersial yang digunakan untuk mendefinisikan, membuat, memelihara dan menyediakan akses terkontrol terhadap basis data dan *repository*

4. Basis Data

Merupakan suatu kelompok data-data yang terorganisir yang saling berhubungan, dan didesain untuk memenuhi kebutuhan informasi banyak pemakai dalam suatu organisasi

5. Program Aplikasi

Program komputer yang digunakan untuk membuat dan memelihara basis data serta menyediakan informasi kepada pemakai

6. Antar-muka pemakai

Bahasa, menu dan fasilitas lainnya yang membuat pemakai dapat berinteraksi dengan beragam komponen dalam sistem

7. Administrator data

Orang yang bertanggung jawab atas semua sumber informasi pada organisasi

8. *System Developer*

Orang-orang seperti sistem analis dan programmer yang mendesain program aplikasi baru

9. *End Users*

Orang-orang yang berada pada organisasi yang menambah, mengurangi serta mengubah data dalam basis data dan orang yang meminta atau menerima informasi dari basis data.

### **2.3.3 Database Management System (DBMS)**

Menurut Hoffer, Prescott dan McFadden (2005, p7) DBMS merupakan sebuah sistem piranti lunak yang digunakan untuk membuat, memelihara dan menyediakan akses kontrol kepada pengguna basis data. Sebuah DBMS menyediakan metode yang sistematis untuk menciptakan, memperbaharui, menyimpan dan mengambil kembali data di dalam basis data.

Menurut Connolly dan Begg (2005, p16) DBMS merupakan sistem piranti lunak dimana pemakainya dapat mendefinisikan, menciptakan, memelihara dan mengontrol akses ke basis data. DBMS menyediakan beberapa fasilitas :

1. Mengijinkan pengguna untuk menentukan basis data, biasanya melalui *Data Definition Language* (DDL). DDL menyediakan fasilitas bagi *Database*

*Administrator* (DBA) dan pemakai untuk menentukan tipe data, entitas, atribut dan relasi yang diperlukan aplikasi serta batasan integritas dan keamanan yang bisa disimpan di basis data.

2. Mengizinkan pengguna untuk melakukan pengoperasian data seperti memasukkan, memperbaharui, menghapus dan mengambil data dari basis data.
3. DBMS juga menyediakan akses kontrol terhadap basis data. Contoh akses kontrol yang disediakan :
  - a. *Security System*, yang dapat mencegah pengguna yang belum terotorisasi yang hendak mengakses basis data
  - b. *Integrity System*, memelihara konsistensi dalam penyimpanan data
  - c. *Concurrency and Control System*, yang memungkinkan pembagian akses basis data
  - d. *Recovery Control System*, yang dapat mengembalikan basis data ke keadaan awal apabila terjadi kesalahan pada piranti lunak ataupun keras.
  - e. *User Accesible Catalog*, yang berisi deskripsi data yang terdapat dalam basis data.

### **2.3.3.1 Komponen-komponen dalam lingkungan DBMS**

Menurut Connolly dan Begg (2005, p18), komponen yang ada dalam lingkungan DBMS adalah sebagai berikut :

#### 1. *Hardware*

Terdiri dari :

- a. Penyimpanan permanen (*magnetic disk* atau *hard disk*), perangkat I/O, *Device Controller*, *I/O Channels*, dan lainnya

b. *Hardware Processor* dan *main memory*, digunakan untuk mendukung saat eksekusi sistem *software* basis data.

## 2. *Software*

DBMS, sistem operasi (seperti Microsoft Windows atau Linux), *network software* (jika diperlukan) dan aplikasi program pendukung lainnya.

## 3. *Data*

Data pada sebuah sistem basis data harus terintegrasi dan dapat digunakan bersama baik itu sistem *single-user* maupun sistem *multi-user*.

## 4. *Procedures*

Prosedur yang digunakan dalam DBMS, misalnya saat *log-on* DBMS atau saat memulai dan mengakhiri DBMS

## 5. *People*

a. *Application Programmer*, bertanggung jawab untuk membuat aplikasi basis data dengan menggunakan bahasa pemrograman seperti VB, Java, dll.

b. *End-User*, orang yang berinteraksi dengan sistem melalui *workstation* atau terminal.

c. *DA (Data Administrator)*, seorang yang berwenang untuk membuat keputusan strategis dan kebijakan mengenai data yang ada, *DBA (Database Administrator)*, menyediakan dukungan teknis untuk implementasi keputusan tersebut, dan bertanggung jawab atas keseluruhan kontrol sistem pada tingkatan teknis.

d. *Database Designer*

Dalam sebuah proyek basis data yang besar, seharusnya terdapat dua tipe *designer* yaitu *logical database designer* dan *physical database designer*. Dimana, *Logical database Designer* bertugas untuk mengidentifikasi data (entitas dan atribut), hubungan antar data dan hambatan penyimpanan data ke dalam basis data. Sedangkan *Physical database designer* menentukan bagaimana basis data logikal untuk difisikkan.

### **2.3.3.2 Keuntungan DBMS**

Dalam proses penggunaannya, Menurut Connolly (2005, p26), DBMS memiliki keuntungan dan kekurangan yang mendukungnya.

Keuntungan dari DBMS, antara lain :

1. Kontrol terhadap pengulangan data
2. Data yang dihasilkan konsisten
3. Pada beberapa data yang sama, semakin banyak informasi yang diperoleh
4. Data dapat dipakai secara bersama-sama
5. Meningkatkan integritas data
6. Meningkatkan keamanan
7. Penetapan standarisasi
8. Perbandingan skala ekonomi
9. Mengatasi konflik kebutuhan
10. Memperbaiki pengaksesan data
11. Meningkatkan produktivitas
12. Memperbaiki pemeliharaan data melalui data yang tidak tergantung dengan data yang lain

13. Memperbaiki *concurrency*

14. Memiliki *backup* data

### 2.3.3.3 Kerugian DBMS

Kerugian penggunaan DBMS, antara lain :

1. Memiliki sistem yang kompleks
2. Karena sistem yang kompleks mengakibatkan DBMS memiliki ukuran yang semakin besar
3. DBMS memiliki harga yang bervariasi tergantung fungsi dan kebutuhan
4. Penambahan biaya untuk perangkat keras yang dibutuhkan
5. Penambahan biaya konversi
6. Karena DBMS dirancang untuk mengakses lebih dari satu aplikasi sehingga performanya menurun
7. Kegagalan dalam DBMS mengakibatkan operasi tidak dapat berjalan

Fungsi dan servis yang disediakan oleh DBMS menurut Connolly (2005, p48)

- *Data Storage, retrieval* dan *update*

Sebuah DBMS harus menyediakan fasilitas kepada pengguna untuk menyimpan, mengambil dan memperbaharui data dalam sebuah basis data

- *User-accessible catalog*

Sebuah DBMS harus menyediakan katalog yang berisi penjelasan dari *data items* yang tersimpan dalam basis data

- *Transaction support*

Sebuah DBMS menyediakan mekanisme yang memastikan bahwa *update* pada suatu transaksi terjadi seluruhnya atau tidak terjadi sama sekali

- *Concurency control service*

## 2.4 Diagram Hubungan antar Entitas (*Entity-Relationship Diagram*)

Menurut Connolly (2005, p342), diagram hubungan antar entitas terbentuk dari identifikasi data-data penting dalam perusahaan yang menjadi entitas-entitas. ERD dibuat untuk menunjukkan hubungan-hubungan antar entitas. Entitas itu sendiri adalah sesuatu hal, baik nyata atau abstrak tentang penyimpanan data. Berikut ini dijelaskan konsep dasar dari ER Modelling, yaitu entiti, *relationship*, atribut, dll sebagai berikut :

### 1. Tipe Entiti

Tipe entiti merupakan kumpulan dari obyek yang memiliki properti yang sama. Sebuah tipe entiti ada secara independen dan bisa berbentuk obyek dengan keberadaan fisik, nyata atau obyek yang tidak nyata atau abstrak.

### 2. Tipe Relasi

Tipe relasi adalah kumpulan hubungan yang mempunyai arti antara tipe-tipe entiti. Setiap tipe relasi akan diberikan nama yang menggambarkan fungsinya, misalnya tipe relasi yang dinamakan *Powns*, yaitu relasi yang menghubungkan entiti *PrivateOwner* dan *PropertyForRent*.

### 3. Atribut

Merupakan sifat dari sebuah entiti atau tipe relasi, misalnya entiti staff bisa digambarkan dengan atribut staffNo, nama, jabatan dan gaji. Atribut ini mempunyai nilai yang menggambarkan seriap entiti dan merepresentasikan bagian utama dari data yang akan disimpan dalam basis data.

#### 1) Domain Atribut

Merupakan kumpulan dari nilai-nilai yang diperbolehkan untuk satu atau lebih atribut, misalnya untuk atribut noKursi harus diisi dengan nilai antara 1 sampai 10.

#### 2) *Simple and Composite Attribute*

- a. *Simple attribute* adalah atribut yang tersusun dari sebuah komponen yang ada secara independen. *Simple attribute* tidak dapat dipecah lagi menjadi atribut yang lebih kecil, biasanya disebut dengan *atomic attribute*, contohnya adalah jabatan, gaji pada entiti staff.
- b. *Composite Attribute* adalah atribut yang tersusun dari banyak komponen, setiap komponen itu ada secara independen. *Composite attribute* dapat dipecah menjadi komponen-komponen independen yang lebih kecil, misalnya entiti cabang dengan nilai (163 Main St, Glasgow, G119QX) dapat dipecah menjadi atribut jalan (163 Main St), kota (Glasgow), kodePos (G119QX)

### 3) *Single-value and Multi-value Attributes*

a. *Single-value attribute* adalah atribut yang hanya mempunyai sebuah nilai untuk setiap tipe entiti. Hampir sebagian besar atribut adalah *single-value*, misalnya pada entiti cabang mempunyai noCabang (B003), noCabang ini adalah *single-value* atribut.

b. *Multi-valued attribute* merupakan atribut yang mempunyai banyak nilai untuk setiap tipe entiti, misalnya pada noCabang B003 mempunyai noTelp 0141-339-2178 dan 0141-339-4439. Pada kasus ini, noTelp merupakan *multi-value attribute*.

#### c. *Derived Attributes*

Yaitu suatu atribut yang merepresentasikan sebuah nilai yang berasal dari nilai sebuah atribut yang berhubungan atau set atribut, dan tidak harus berada dalam tipe entiti yang sama.

#### d. *Keys*

a) *Candidate key* merupakan kumpulan minimal dari atribut yang secara unik mengidentifikasi tipe entiti tertentu.

b) *Primary key* adalah *candidate key* yang dipilih secara unik untuk mengidentifikasi suatu tipe entiti. *Candidate key* lainnya yang tidak dipilih menjadi *primary key* disebut dengan *alternate key*.

c) *Composite key* adalah *candidate key* yang terdiri dari dua atau lebih atribut.

#### 4. *Strong and weak entity types*

1) *Strong entity* adalah entiti yang keberadaannya tidak tergantung dengan entiti lain, misalnya entiti staff, cabang. Ciri dari *strong entity* adalah tiap entiti dapat diidentifikasi dengan *primary key* dan tipe entiti itu.

2) *Weak entity* adalah entiti yang keberadaannya tergantung dari entiti lain. Karakteristik dari *weak entity* adalah atribut yang terdapat pada entiti tersebut tidak dapat mengidentifikasi tipe entiti itu secara unik.

#### 5. *Structural Constraints*

*Multiplicity* adalah jumlah kemunculan yang mungkin dari sebuah tipe entiti yang berhubungan dengan kemunculan tunggal dari sebuah tipe entiti yang berhubungan melalui relasi tertentu (Connolly, 2005, p356). Contohnya batasan (*constraint*) bahwa sebuah properti sewa harus mempunyai seorang pemilik dan setiap cabang harus mempunyai staff. Tipe utama dari batasan dalam relasi ini disebut dengan *multiplicity*. *Multiplicity* membatasi cara entiti-entiti berhubungan. Batasan ini dapat dibuat oleh organisasi ataupun user.

## 2.5 *Structured Query Language (SQL)*

*Structured Query Language* atau SQL merupakan salah satu contoh dari bahasa *transform-oriented*, atau bahasa yang diciptakan dengan menggunakan relasi untuk mengubah masukan menjadi keluaran yang dibutuhkan (Connolly, 2005, p113). Standar ISO SQL memiliki dua komponen utama :

1. DDL (*Data Definition Language*), untuk mencari struktur basis data dan mengontrol akses data
2. DML (*Data Manipulation Language*), untuk menerima dan mengubah data

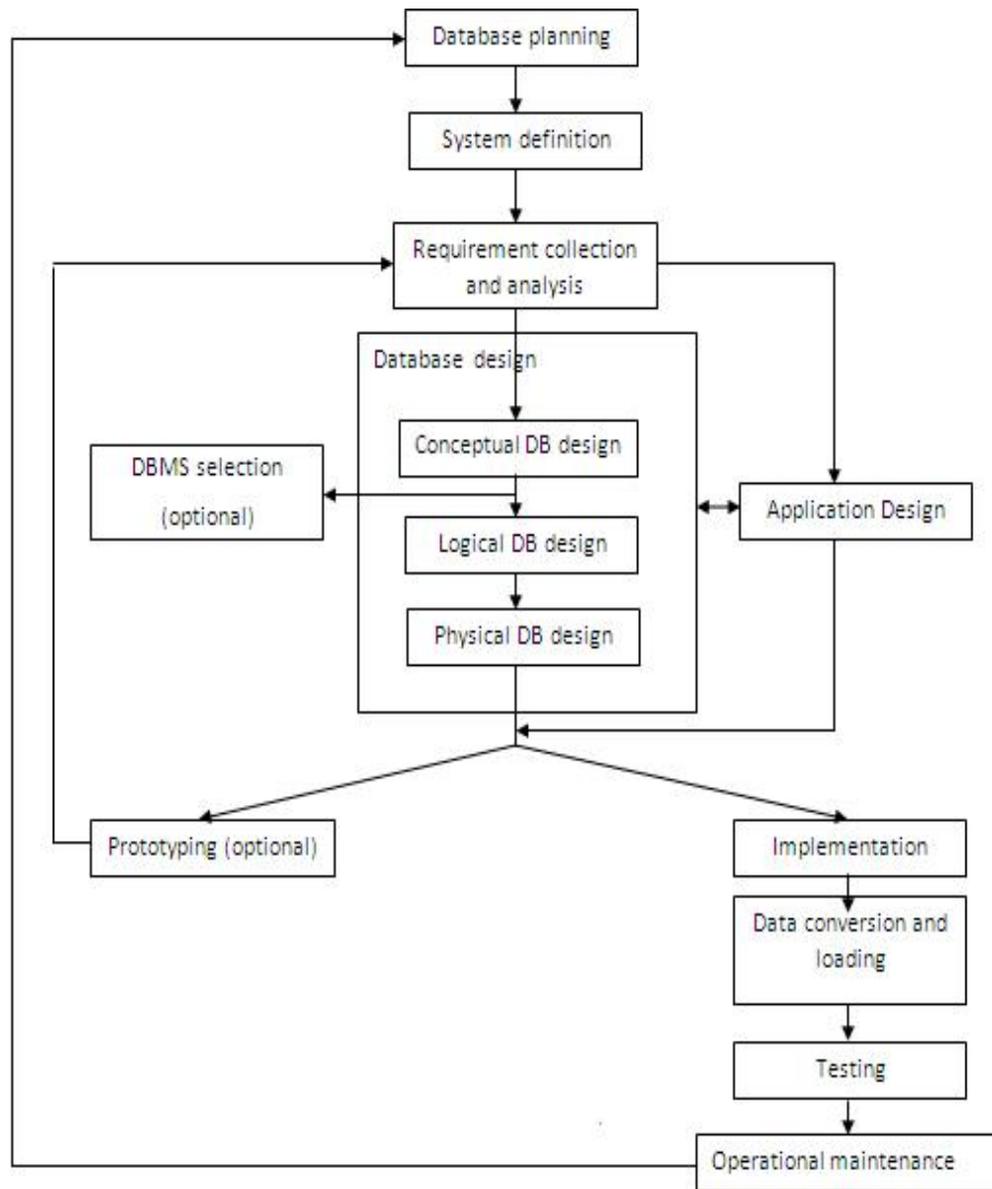
Seperti kebanyakan bahasa modern, SQL pada dasarnya memiliki format bebas (*free-format*), yang berarti bahwa bagian-bagian dari *statement* tidak harus diketik pada lokasi tertentu pada layar. Tetapi meskipun *free-format*, *statement* SQL akan lebih menarik dan mudah dibaca jika menggunakan *indentation* dan *lineation*.

Menurut Connolly (2005, p117), secara umum *statement* DML antara lain :

- a. SELECT → untuk *query* dalam basis data
- b. INSERT → untuk memasukkan data ke dalam tabel
- c. UPDATE → untuk memperbaharui data dalam tabel
- d. DELETE → untuk menghapus data dari tabel

## **2.6 Siklus Hidup Aplikasi Basis Data**

Untuk merancang sebuah sistem basis data, kita juga harus memperhatikan Siklus Hidup Aplikasi Basis Data. Menurut Connolly (2005, p283), Sistem Basis Data merupakan suatu dasar bagi komponen dari organisasi dengan sistem informasi yang besar.



**Gambar 2.1 Siklus Hidup Aplikasi Basis Data**

### 2.6.1 Perencanaan Basis Data

Perencanaan basis data adalah kegiatan pengaturan yang memungkinkan tahap-tahap dalam aplikasi basis data dapat diwujudkan secara efisien dan efektif (Connolly, 2005, p285). Tahap perencanaan basis data juga harus menjelaskan :

1. *Mission statement*, menjelaskan tujuan utama aplikasi basis data dan membantu menjelaskan tujuan proyek basis data (Connolly, 2005, p286). Dengan rumusan ini diharapkan dapat memfokuskan pada langkah dan pekerjaan selanjutnya untuk menjadikannya efektif dan efisien.
2. *Mission Objectives*, untuk menjelaskan mengenai tugas apa saja yang harus didukung oleh basis data tersebut. Setiap *mission objective* akan menjelaskan tugas tertentu yang harus didukung oleh basis data, dengan asumsi jika basis data yang mendukung *mission objectives*, maka *mission statementnya* juga akan sesuai (Connolly, 2005, p286).

### 2.6.2 Pendefinisian Sistem

Pendefinisian sistem menggsambarkan ruang lingkup dan batasan aplikasi basis data dan pandangan pengguna (Connolly, 2005, p286). Hal ini sangat penting dilakukan dalam proses perancangan basis data agar lebih fokus pada proyek basis data yang sedang dikerjakan.

Pandangan pemakai (*user view*) sangat diperlukan untuk mengidentifikasi informasi-informasi yang dibutuhkan oleh pemakai. Pandangan pengguna menggambarkan apa yang dibutuhkan oleh aplikasi basis data dari sudut pandang

tertentu, seperti manajer atau pengawas, maupun Dari sudut pandang area aplikasi perusahaan, seperti pemasaran atau personalia, dalam hubungannya dengan data yang akan disimpan dan transaksi yang akan dijalankan terhadap data itu (Connolly, 2005, p287)

### **2.6.3 Analisa dan Pengumpulan Informasi**

Langkah selanjutnya adalah menganalisis dan mengumpulkan informasi sesuai dengan permintaan pemakai. Proses ini didukung oleh aplikasi basis data dan digunakan untuk mengidentifikasi kebutuhan pemakai, yang meliputi uraian mengenai data yang digunakan atau yang dihasilkan, rincian mengenai bagaimana data digunakan atau yang dihasilkan, kebutuhan-kebutuhan tambahan untuk aplikasi basis data yang baru.

Informasi ini kemudian dimasukkan ke dalam aplikasi basis data yang baru. Untuk itu digunakan teknik yang disebut *fact-finding techniques*. Terdapat lima teknik yang umum digunakan :

1. Mengevaluai dokumen
2. *Interview*
3. Mengobservasi jalannya kegiatan kerja pada perusahaan
4. *Research riil*
5. *Quesioner*

### **2.6.4 Perancangan Basis Data**

Perancangan basis data (*database design*) adalah proses pembuatan sebuah rancangan untuk basis data yang akan mendukung operasi dan tujuan perusahaan

(Connolly, 2005, p291).. Dalam perancangannya dibutuhkan metodologi yang benar sehingga tahap perencanaan dapat berjalan sesuai dengan konsep yang telah dibuat. Metodologi perancangan (*design methodology*) adalah pendekatan terstruktur yang menggunakan prosedur-prosedur, teknik-teknik, peralatan dan dokumentasi untuk mendukung dan memudahkan proses perancangan (Connolly , 2005, p438). Teknik ini digunakan untuk membantu, merancang, mengatur, mengontrol dan mengevaluasi proyek pengembangan basis data. Menurut Connolly (2005, p440) terdapat 3 tahapan dalam metodologi perancangan basis data, antara lain:

### **1. Perancangan Basis Data Konseptual**

Merupakan proses membangun model informasi yang digunakan organisasi, bebas dari semua pertimbangan fisik (Connolly, 2005, p439). Pertimbangan fisik yang dimaksud meliputi *DBMS* yang akan digunakan, program aplikasi, bahasa pemrograman, *platform* perangkat keras, dan pertimbangan fisik lainnya.

#### **Langkah 1 : Membuat model data konseptual untuk setiap bagian**

Tujuan dari perancangan basis data konseptual lokal adalah untuk memproses pembuatan suatu model informasi yang digunakan di dalam suatu organisasi, dimana model tersebut tidak tergantung pada perangkat keras lainnya. Langkah-langkah pembuatannya sebagai berikut :

##### **Langkah 1.1 : Mengidentifikasi tipe entitas**

Pendefinisian tipe entitas sangat dibutuhkan oleh pemakai dalam membangun suatu model data konseptual. Salah satu metode untuk mengidentifikasi tipe entitas yang utama adalah dengan mengidentifikasi

kata benda atau frase kata benda yang telah disebutkan user. Sebagai contoh : NoStaff, NamaStaff, TelpStaff, QtyBarang.

### **Langkah 1.2 : Mengidentifikasi tipe relasi**

Tujuan dari langkah ini adalah untuk mengidentifikasi relasi yang penting antara berbagai tipe entitas yang telah diidentifikasi. Biasanya, relasi diidentifikasi dengan menggunakan kata kerja atau frase kata kerja.

Sebagai contohnya :

- a) *Staff Manages PropertyForRent*
- b) *PrivateOwner Owns PropertyForRent*

Relasi paling umum adalah relasi *binary*, artinya relasi antar entitas yang persis antara dua entitas saja.

Langkah-langkah dalam mengidentifikasi tipe relasi adalah sebagai berikut :

- a) Gunakan *Entity Relationship Diagram (ERD)*

*ERD* dapat digunakan untuk mempresentasikan entitas dan relasi antar entitas, sehingga didapat gambaran dari perancangan basis data yang sedang dikembangkan.

- b) Tentukan pembatas *multiplicity* dari tipe relasi

Setelah didapat relasi antara entitas, maka langkah berikutnya adalah menentukan *multiplicity* setiap relasi. Jika terdapat suatu nilai spesifik dari *multiplicity* maka lebih baik apabila didokumentasikan.

c) Memeriksa *Fan* dan *Chasm Traps*

Setelah mendefinisikan relasi yang dibutuhkan antar entitas, maka langkah berikutnya adalah memeriksa *fan* dan *chasm traps*.

d) Memeriksa setiap entitas yang mempunyai relasi minimal satu

Pada saat pembuatan *ERD*, pastikan bahwa setiap entitas mempunyai minimal satu relasi dengan entitas lain. Jika setiap entitas sudah mempunyai minimal satu relasi dengan entitas yang lain maka langkah berikutnya adalah memperhatikan kamus data. Jika sudah memiliki relasi semua, berarti setiap entitas telah memiliki relasi.

**Langkah 1.3 : Mengidentifikasi dan mengasosiasikan atribut suatu entitas atau tipe relasi**

Tujuan dari langkah ini adalah untuk mengidentifikasi dan mengasosiasikan atribut dari entitas atau tipe relasi, apakah atribut itu *Simple/Composite Attributes*, *Single/Multi-valued Attributes*, *Derived Attributes*.

**Langkah 1.4 : Menentukan domain atribut**

Langkah ini bertujuan untuk menentukan domain dari atribut yang ada di dalam model data konseptual lokal. Sebagai contoh antara lain :

1. Atribut dari No Staff terdiri dari lima karakter tipe string dimana dua karakter utama adalah huruf dan tiga karakter sisanya berupa angka.

2. Nilai yang mungkin untuk atribut jenis kelamin adalah 'M' atau 'F'.

Ini merupakan dominan dari atribut yang menggunakan karakter tunggal.

### **Langkah 1.5 : Menentukan *candidate key* dan *primary key***

Tujuan dari langkah ini adalah untuk mengidentifikasi kandidat key dari setiap entitas. Jika terdapat lebih dari satu kandidat key, pilihlah salah satunya untuk menjadi *primary key*. Pentunjuk berikut dapat membantu penyeleksian saat pemilihan *primary key* :

- 1) Merupakan kandidat key dengan jumlah set atribut paling sedikit.
- 2) Merupakan kandidat key yang nilainya jarang sekali berubah.
- 3) Merupakan kandidat key dengan jumlah karakter paling sedikit.
- 4) Merupakan kandidat key dengan jumlah paling sedikit dari nilai maksimumnya (untuk tipe atribut dengan tipe numerik).
- 5) Merupakan kandidat key yang paling mudah digunakan dari sudut pandang pengguna.

### **Langkah 1.6 : Menggunakan Enhanced Modeling Concepts**

#### **(langkah optional)**

Langkah ini untuk mempertimbangkan penggunaan model enchanted, seperti spesialisasi / generalisasi, agregasi dan komposisi (Connolly, 2005, p453). Jika kita menggunakan pendekatan spesialisasi, yang akan dibuat adalah perbedaan antar entiti dengan mendefinisikan satu atau lebih subclass dari entiti superclass. Sedangkan dengan

pendekatan generalisasi, yang akan dilakukan adalah mengidentifikasi fitur-fitur antar entiti untuk mendefinisikan generalisasi entiti superclass.

### **Langkah 1.7 : Memeriksa redudansi**

Tujuan dari langkah ini adalah untuk memeriksa apakah ada redudansi dalam model basisdata. Pada langkah ini, anda akan menguji data dan model konseptual lokal dengan melihat secara spesifik, apabila ada redudansi maka dapat dihilangkan dengan dua langkah berikut :

- 1) Menguji kembali hubungan one-to-one
- 2) Menghilangkan relasi redudansi

### **Langkah 1.8 : Validasi model konseptual dengan transaksi user**

Bertujuan untuk memastikan *local conceptual data model* mendukung transaksi yang dibutuhkan oleh pandangan pengguna, dengan menggunakan dua pendekatan berikut (Connolly, 2005, p456)

- 1) Menggambarkan transaksi (*describing the transaction*)  
Memeriksa semua informasi (entitas, *relationship*, dan atribut) yang dibutuhkan setiap transaksi yang disediakan oleh model.
- 2) Menggunakan transaksi *pathways*  
Memvalidasi *data model* terhadap kebutuhan transaksi dengan menggambar diagram yang mewakili *pathways* yang diambil oleh setiap transaksi secara langsung pada ERD

### **Langkah 1.9 : Mereview model data konseptual dengan user**

Tujuannya adalah untuk mengkaji ulang model data konseptual lokal bersama pemakai untuk memastikan bahwa model yang ada sudah sesuai dengan yang diminta.

### **3. Perancangan Basis Data Logikal**

Merupakan proses dalam membangun suatu model informasi yang digunakan didalam perusahaan, tetapi tidak tergantung oleh suatu DBMS dan pertimbangan fisik tertentu, tetapi berdasarkan pada model data spesifik yang diperlukan. Dalam tahap ini, model data yang telah diperoleh dalam *conceptual database design* diubah dalam bentuk *logical model* dimana data yang ada dipengaruhi oleh model data yang menjadi tujuan basis data.

#### **Langkah 2 : Memebuat dan memvalidasi model data logikal untuk setiap bagian**

Tujuan dari langkah ini adalah untuk membangun sebuah model data logikal lokal dari suatu model data konseptual lokal yang merepresentasikan perusahaan kemudian memvalidasi model ini untuk memastikan strukturnya benar, dan memastikan bahwa model tersebut mendukung transaksi yang diminta.

#### **Langkah 2.1 : Menghilangkan bagian yang tidak sesuai dengan model**

##### **Relasi (langkah optional)**

Tujuan langkah ini adalah untuk memperbaiki model data konseptual lokal dengan menghilangkan fitur yang tidak sesuai dengan model relasi.

Bagian yang akan dibahas pada langkah ini antara lain :

1. Menghilangkan many-to-many (\*..\*) tipe relasi binary.
2. Menghilangkan many-to-many (\*..\*) tipe relasi rekursif.
3. Menghilangkan tipe relasi kompleks.
4. Menghilangkan atribut-atribut multi-valued.

### **Langkah 2.2 : Membuat relasi untuk model data logikal**

Tujuan dari langkah ini adalah untuk membuat suatu relasi untuk model data logikal lokal yang merepresentasikan suatu entitas, relasinya dan juga atribut yang telah diidentifikasi. Anda dapat mendeskripsikan bagaimana relasi diturunkan dari struktur model data antara lain :

1. Tipe entitas kuat.
2. Tipe entitas lemah.
3. One-to-many (1..\*) tipe relasi binary.
4. One-to-one (1..1) tipe relasi binary.
5. One-to-one (1..1) tipe relasi rekursif.
6. Tipe relasi *superclass / subclass*.
7. Many-to-many tipe relasi binary
8. Tipe relasi kompleks.
9. Atribut-atribut *multi-valued*.

### **Langkah 2.3 : Memvalidasi relasi menggunakan normalisasi**

Langkah ini merupakan sebuah teknik untuk menghasilkan sejumlah relasi dengan sifat-sifat yang diinginkan, dengan memberikan kebutuhan data dari sebuah perusahaan (Connolly, 2005, p388)

Tujuan dari langkah ini adalah untuk memvalidasi relasi dalam *Local Logical Data Model* dengan menggunakan teknik normalisasi.

Tujuan dari normalisasi adalah sebagai berikut:

- a. Menghilangkan kumpulan relasi dari insert, update dan delete *dependency* yang tidak diharapkan.
- b. Mengurangi kebutuhan restrukturisasi kumpulan relasi dan meningkatkan *life span* dari program aplikasi.
- c. Membuat model relasional lebih informatif.

Tahapan dari normalisasi terdiri dari :

1. Normalisasi tahap pertama (1NF) untuk menghilangkan perulangan.
2. Normalisasi tahap kedua (2NF), syarat dari bentuk ini adalah data harus memenuhi kriteria 1NF dan setiap *data item* yang bukan kunci harus bergantung secara fungsional (*functional dependent*) pada *primary key*.
3. Normalisasi tahap ketiga (3NF), tahap ini untuk menghilangkan *transitive dependency*.
4. Boyce-Codd Normal Form (BCNF), suatu relasi dikatakan memenuhi BCNF, jika dan hanya jika, setiap determinan adalah *candidate key*.

#### **Langkah 2.4 : Memvalidasi relasi dengan transaksi user**

Tujuan dari langkah ini adalah untuk memastikan bahwa relasi didalam model data logikal lokal mendukung transaksi yang diminta user. Pada langkah ini, anda akan memastikan bahwa relasi yang dibuat di langkah sebelumnya juga mendukung transaksi ini, dan juga pastikan bahwa tidak ada kesalahan dalam relasi yang telah dibuat.

### **Langkah 2.5 : Memeriksa integritas basis data**

Tujuan dari langkah ini adalah untuk mendefinisikan integritas konstrain yang diberikan dalam view.

Terdapat lima tipe integriti konstrain yang harus dipertibangkan antara lain :

- a. *Reuqired data*
- b. *Attribute domain constraint*
- c. *Entity integrity*
- d. *Referential integrity*
- e. *Enterprise constraint*

### **Langkah 2.6 : Mereview model data logikal dengan user**

Tujuan dari langkah ini adalah untuk memastikan bahwa model data lokal logikal dan dokumentasi yang mendukung yang menjelaskan model merupakan representasi yang benar dari view.

### **Langkah 3 : Membangun dan memvalidasi model data logikal**

Pada langkah ini model data lokal logikal individual dikombinasikan menjadi sebuah model data global logikal tunggal yang menggambarkan perusahaan.

#### **Langkah 3.1 : Menggabungkan model data logikal menjadi model model global**

Tahap ini akan menggabungkan model data lokal logikal individual menjadi sebuah model data logikal tunggal dari perusahaan. Beberapa tugas dari pendekatan ini adalah :

- 1) Mereview nama dan isi dari entiti / relasi dari candidate key nya.

- 2) Mereview nama dan isi dari relasi / foreign key.
- 3) Menggabungkan entiti / relasi dari model data lokal.
- 4) Memasukkan entiti / relasi secara unik ke setiap model data lokal.
- 5) Menggabungkan relationship / foreign key dari model data lokal.
- 6) Memasukkan relationship / foreign key secara unik ke setiap model data lokal.
- 7) Mengecek entiti / relasi dan relationship / foreign key yang hilang.
- 8) Mengecek foreign key.
- 9) Mengecek barisan integritas.
- 10) Menggambar diagram ER / relasi global.
- 11) Meng-update dokumentasi.

### **Langkah 3.2 : Memvalidasi model data logikal global**

Langkah ini bertujuan untuk memvalidasi relasi yang dibuat dari model data global logikal menggunakan teknik normalisasi dan untuk memastikan mendukung transaksi yang diperlukan.

### **Langkah 3.3 : Memeriksa kemungkinan adanya pengembangan di masa mendatang**

Langkah ini bertujuan untuk memastikan apakah ada perubahan yang signifikan di masa yang akan datang dan untuk memastikan apakah model data global logikal dapat mengakomodasi perubahan tersebut.

### **Langkah 3.4 : Mereview model data logikal global dengan user**

Langkah ini bertujuan untuk memastikan bahwa model data logikal merupakan representasi yang benar dari perusahaan.

### **3. Perancangan Basis Data Fisikal**

Proses menghasilkan deskripsi dari implementasi basis data pada *secondary storage*, yang menjelaskan relasi dasar, *file* pada perusahaan dan index yang digunakan untuk mendapatkan akses yang efisien ke dalam data, dan semua pertimbangan keamanan dari *integrity constraint* yang berhubungan. Tujuan utama dari perancangan basis data fisikal adalah menentukan bagaimana struktur logikal diimplementasikan secara fisikal ke dalam DBMS (Connolly, 2005, p417)

Beberapa langkah dalam perancangan basis data logikal (Connolly, 2005, p479)

#### **Langkah 4 : Menterjemahkan model data logikal global untuk DBMS yang digunakan**

Tujuan dari langkah ini adalah untuk menghasilkan skema relational basis data model data global logikal yang dapat diimplementasikan sebagai target DBMS

##### **Langkah 4.1 : Merancang basis data relasional**

Tujuannya adalah untuk mendapatkan relasi diantara *logical global data* saat diterapkan pada suatu DBMS.

#### **Langkah 4.2 : Mendesain representasi dari data yang sudah diturunkan**

Tujuannya adalah untuk membuat representasi dengan menerapkan data-data ke dalam DBMS

#### **Langkah 4.3 : Merancang batasan perusahaan**

Tujuannya adalah untuk menentukan pengaturan *file* yang optimal untuk menyimpan tabel-tabel dan sistem *index* sehingga sistem dapat memenuhi kebutuhan yang diinginkan.

#### **Langkah 5 : Merancang representasi fisik**

Tujuan dari langkah ini adalah untuk menentukan organisasi *file* yang optimal untuk menyimpan relasi dasar (*base relation*) dan indeks yang diperlukan.

#### **Langkah 5.1 : Menganalisis transaksi**

Tujuan dari langkah ini adalah untuk memahami kegunaan transaksi yang dijalankan pada basis data dan untuk menganalisa transaksi yang penting.

Dalam menganalisa transaksi, kita berusaha untuk mengidentifikasi kriteria kinerja, seperti :

- 1) Transaksi yang sering dilakukan dan akan memiliki pengaruh yang berarti pada kinerja
- 2) Transaksi yang kritis terhadap operasi bisnis
- 3) Waktu selama satu hari/minggu dimana akan ada permintaan yang tinggi terhadap basis data (*peak load*).

**Langkah 5.2 : Memilih organisasi file**

Tujuannya untuk menentukan sebuah organisasi *file* yang efisien untuk setiap relasi dasar (*base relations*).

Untuk membantu memahami organisasi *file* pada setiap relasi disediakan petunjuk untuk memilih sebuah organisasi *file* berdasarkan tipe *file* berikut:

- 1) Heap
- 2) Hash
- 3) Indexed Sequential Office Access Method (ISAM)
- 4) B<sup>+</sup>-tree
- 5) Cluster

**Langkah 5.3 : Memilih indeks**

Tujuannya untuk menentukan apakah menambahkan indeks meningkatkan kinerja pada sistem.

Pada kasus ini, pilih atribut untuk mengurutkan atau mengelompokkan *tuple* sebagai:

- 1) Atribut yang paling sering digunakan pada operasi *join*, karena atribut ini membuat operasi *join* lebih efisien, atau
- 2) Atribut yang paling sering digunakan untuk mengakses *tuple* dalam sebuah relasi untuk atribut tersebut

**Langkah 5.4 : Mengestimasi kapasitas disk yang dibutuhkan**

Langkah ini bertujuan untuk memperkirakan jumlah kapasitas disk yang dibutuhkan oleh basis data. Merupakan sebuah kebutuhan bahwa implementasi basis data dapat ditangani oleh konfigurasi perangkat keras yang sedang digunakan. Tujuan dari tahapan ini adalah

untuk memperkirakan jumlah dari kapasitas disk yang diperlukan untuk mendukung implementasi basis data pada penyimpanan sekunder.

### **Langkah 6 : Merancang tampilan layar untuk user**

Tujuan dari langkah ini adalah untuk merancang pandangan user yang didefinisikan selama pengumpulan kebutuhan dan menganalisa tahapan dari siklus hidup aplikasi basis data relasional.

#### **2.6.5 Pemilihan DBMS**

Pemilihan BMS dipengaruhi oleh beberapa faktor sebagai berikut :

1. Kemudahan dalam menggunakannya (*case of use*)
2. Keandalan (*reliability*)
3. Biaya (*cost*)
4. Keamanan (*security*)
5. *Compability*
6. Persyaratan Minimum (*Minimum Requirement*)
7. Kedekatan dengan basis data (*familiarity with database*)
8. Skalability

Tahapan utama dalam memilih DBMS antara lain :

1. Mendefinisikan syarat-syarat sebagai referensi
2. Daftar singkat dua atau tiga produk
3. Evaluasi produk
4. Merekomendasikan pilihan dan memproduksi laporan

### **2.6.6 Perancangan Aplikasi Basis Data**

Desain aplikasi dibagi menjadi dua aspek, yaitu :

1. Desain transaksi
  - *Retrieve Transaction*
  - *Update Transaction*
  - *Mixed Transaction*

### **2.6.7 Prototyping**

Pada langkah ini, perancang dan pemakai memvisualisasikan dan mengevaluasi gambaran sistem secara menyeluruh. Tujuannya adalah untuk memungkinkan pemakai menggunakan *prototype*, sehingga dapat mengidentifikasi fitur sistem dan jika memungkinkan, pemakai dapat memberikan saran untuk perbaikan atau menambahkan fitur baru dalam aplikasi tersebut.

### **2.6.8 Implementasi**

Merupakan realisasi fisik dari *database* dan desain aplikasi (Connolly, 2005, p304). Dilakukan dengan menggunakan *Data Definition Language* (DDL) dari DBMS yang dipilih. Aplikasinya terdapat transaksi basis data yang diimplementasikan menggunakan *Data Manipulation Language* (DML).

### **2.6.9 Konversi Data dan Loading**

Memindahkan data yang ada ke dalam basis data yang baru dan mengubah aplikasi yang ada untuk dijalankan di dalam basis data yang baru (Connolly, 2005, p305). Dewasa ini, DBMS mempunyai fungsi untuk mengambil basis data yang ada

untuk dimasukkan ke basis data yang baru yang secara otomatis merubah data tersebut ke dalam format yang sesuai dengan basis data yang baru.

#### **2.6.10 Testing**

Pengujian dilakukan terhadap aplikasi basis data dengan tujuan untuk melihat kesalahan-kesalahan dan memvalidasi sesuai dengan kebutuhan *user* (Connolly, 2005, p305). Pengujian dilakukan secara terencana dan hati-hati. Standar dalam sasaran pengujian antara lain :

1. Pengujian merupakan pengekseskusion program dengan tujuan pencarian kesalahan
2. *Test case* yang baik adalah yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya
3. Pengujian yang sukses merupakan pengujian yang mampu mengungkapkan semua kesalahan yang belum pernah ditemukan sebelumnya

#### **2.6.11 Operational Maintenance**

*Operational Maintenance* merupakan proses pemantauan dan pemeliharaan sistem basis data (Connolly, 2005, p306). Saat dibutuhkan, maka permintaan-permintaan baru akan dimasukkan ke dalam aplikasi basis data melalui tahap-tahap daur hidup sebelumnya. Kegiatan-kegiatan pemeliharaan tersebut dilaksanakan dalam beberapa tahap berikut :

1. Memantau performa sistemnya. Jika kemampuan menurun di bawah standar yang seharusnya, maka perbaikan diperlukan

2. Memelihara dan meningkatkan sistem basis data jika dibutuhkan.

### 2.7 State Transition Diagram

*State Transition Diagram* (STD) adalah diagram yang menggambarkan perubahan dari suatu keadaan selama pemrosesan suatu *finite-state*.

Menurut Mulyadi (2001, p60), ada 2 macam simbol yang menggambarkan proses dalam STD, yaitu :

1. State



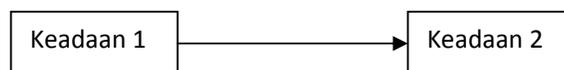
State adalah kumpulan keadaan atau atribut yang mencirikan seseorang atau suatu benda pada waktu tertentu, bentuk kebradaan tertentu atau kondisi tertentu. State disimbolkan dengan segi empat

2. Perubahan State (*State Transition*)



Gambar panah menunjukkan transisi antar state. Tiap panah diberi label dengan ekspresi aturan. Label yang di atas menunjukkan kejadian atau kondisi yang menyebabkan transisi yang terjadi. Label yang di bawah menunjukkan aksi yang terjadi akibat dari kejadian tadi.

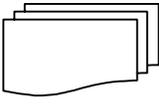
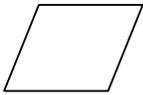
Perubahan State

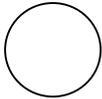


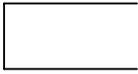
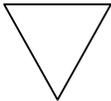
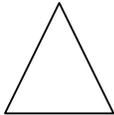
## 2.8 Bagan Alir (*Flow Chart*)

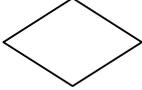
Berikut ini adalah sebuah simbol-simbol dengan maknanya masing-masing :

**Tabel 2.1 Komponen Flow Chart**

Simbol	Nama	Keterangan
	Dokumen	Menggambarkan semua jenis dokumen, formulir yang digunakan untuk merekam data terjadinya suatu transaksi.
	Dokumen dan tembusannya	Menggambarkan dokumen asli dan tembusannya. Nomor lembar dokumen dicantumkan di sudut kanan atas.
	Berbagai dokumen	Menggambarkan berbagai jenis dokumen yang digabungkan bersama. Nama dokumen ditulis di dalam masing-masing simbol dokumen yang dicantumkan di sudut kanan atas
	Catatan	Menggambarkan catatan akuntansi yang digunakan untuk mencatat data

		sebelumnya di dalam dokumen.
	Penghubung pada halaman yang sama ( <i>on-page connector</i> )	Karena keterbatasan kertas untuk menggambar, maka diperlukan simbol penghubung untuk memungkinkan aliran dokumen tertentu di suatu lokasi pada halaman tertentu dan kembali berjalan di lokasi lain pada halaman yang sama. Dengan memperlihatkan nomor yang tercantum di dalam simbol penghubung pada halaman yang sama.
	Penghubung pada halaman yang berbeda ( <i>off-page connector</i> )	Menunjukkan kemana dan bagaimana bagan alir terkait satu dengan yang lainnya. Nomor yang tercantum dalam simbol penghubung menunjukkan hubungan bagan alir terkait dengan halaman yang lain.
	Kegiatan manual	Uraian singkat kegiatan manual dicantumkan didalam simbol ini.

	Keterangan, komentar	Untuk menambahkan keterangan, guna memperjelas pesan yang disampaikan dalam bagan alir.
	Arsip sementara	Menunjukkan tempat penyimpanan dokumen. Arsip sementara adalah tempat penyimpanan dokumen yang nantinya akan diambil kembali dari arsip tersebut untuk keperluan pengolahan lebih lanjut terhadap dokumen tersebut. Untuk pengurutan pengarsipan dokumen digunakan simbol berikut ini : A = menurut abjad. N = menurut nomor urut. T = kronologis, menurut tanggal
	Arsip permanen	Menggambarkan arsip permanen yang merupakan tempat penyimpanan dokumen yang tidak akan diproses lagi dalam sistem yang

		bersangkutan.
	Mulai / berakhir ( <i>terminal</i> )	Menggambarkan awal dan akhir suatu sistem.
	Keputusan	Menggambarkan keputusan yang harus dibuat dalam proses pengolahan data. Keputusan yang dibuat ditulis dalam bentuk simbol.

## 2.9 Pembelian, Persediaan dan Penjualan

### 2.9.1 Pengertian Pembelian

Menurut Render (2001, p414), pembelian adalah perolehan barang dan jasa. Secara umum, definisi pembelian adalah suatu usaha pengadaan barang dan jasadengan tujuan yang akan digunakan sendiri, untuk kepentingan proses produksi maupun untuk dijual kembali.

Tujuan pembelian menurut Render (2001, p414), tujuan dari kegiatan pembelian adalah:

- a. Membantu identifikasi produk dan jasa yang dapat diperoleh secara eksternal.
- b. Mengembangkan, mengevaluasi dan menentukan pemasok, harga dan pengiriman yang terbaik bagibarang dan jasa tersebut.

Jenis transaksi dalam pembelian dibedakan menjadi dua, yaitu :

- a. Pembelian tunai adalah proses pembayaran yang dilakukan secara langsung pada saat barang diterima
- b. Pembelian kredit adalah proses pembayaran yang tidak dilakukan langsung pada saat barang diterima, tetapi pembayaran dilakukan selang beberapa waktu setelah barang diterima sesuai dengan perjanjian kedua belah pihak.

Hal-hal yang harus diperhatikan dalam proses pembelian adalah :

- a. Kualitas
- b. Harga
- c. Waktu proses

Pengawasan perlu dilakukan terhadap pelaksanaan proses pembelian ini, karena pembelian menyangkut investasi dana dalam persediaan dan kelancaran arus barang ke dalam perusahaan. Menurut Mulyadi (2001, p299) fungsi yang terkait dalam sistem pembelian adalah sebagai berikut :

- a. Fungsi Gudang

Bertanggung jawab untuk memperoleh informasi mengenai harga barang, menentukan pemasok yang dipilih dalam pengadaan barang dan mengeluarkan order pembelian kepada pemasok yang dipilih.

- b. Fungsi Pembelian

Bertanggung jawab untuk memperoleh informasi mengenai harga barang, menentukan pemasok yang dipilih dalam pengadaan barang, dan mengeluarkan order pembelian kepada pemasok yang dipilih.

c. Fungsi Penerimaan

Bertanggung jawab untuk melakukan pemeriksaan terhadap jenis, mutu, dan kuantitas bahan yang diterima dari pemasok guna menentukan dapat atau tidaknya barang tersebut diterima oleh perusahaan.

d. Fungsi Akuntansi

Bertanggung jawab dalam pencatatan transaksi pembelian yang menjadi hutang sebagai dokumen bukti kas keluar.

Menurut Mulyadi (2001, p301), jaringan prosedur uang membentuk sistem pembelian adalah sebagai berikut :

1. Prosedur Permintaan Pembelian

Dalam prosedur ini fungsi gudang mengajukan permintaan pembelian dalam formulir surat permintaan pembelian kepada fungsi pembelian.

2. Prosedur Permintaan Penawaran Harga dan Pemilihan Pemasok

Fungsi pembelian mengirimkan surat permintaan penawaran harga kepada pemasok untuk memperoleh informasi mengenai harga barang dan berbagai syarat pembelian yang lain untuk memungkinkan pemilihan pemasok yang akan ditunjuk sebagai pemasok barang yang diperlukan oleh perusahaan.

3. Prosedur Order Pembelian

Fungsi pembelian mengirim surat order pembelian kepada pemasok yang dipilih dan memberitahukan kepada unit-unit lain dalam perusahaan mengenai order pembelian yang sudah dikeluarkan oleh perusahaan.

4. Prosedur Penerimaan barang

Dalam prosedur ini fungsi penerimaan barang melakukan pemeriksaan mengenai jenis, kuantitas dan mutu bahan yang diterima dari pemasok,

kemudian membuat laporan penerimaan barang untuk menyatakan penerimaan barang dari pemasok tersebut.

#### 5. Prosedur Pencatatan Hutang

Dalam prosedur ini akuntansi memeriksa dokumen-dokumen yang berhubungan dengan pembelian (surat order pembelian, laporan penerimaan barang, faktur dari pemasok) dan menyelenggarakan pencatatan hutang atau pengarsipan dokumen sumber sebagai catatan hutang.

#### 6. Prosedur Distribusi Pembelian

Prosedur ini meliputi distribusi rekening yang didebet dari transaksi pembelian untuk kepentingan pembuatan laporan manajemen.

### **2.9.2 Pengertian Persediaan**

Menurut Epstein (2004, p208), persediaan merupakan aset pengadaan barang di dalam sebuah bisnis, atau yang sedang dalam proses produksi untuk penjualan tertentu, atau dalam wujud material atau pendukung untuk digunakan dalam proses produksi atau penyumbangan jasa.

Dengan demikikian dapat disimpulkan bahwa persediaan barang-barang yang dibeli untuk produksi, serta dijual kembali dan habis dipakai oleh perusahaan. Karena pentingnya peranan persediaan barang bagi perusahaan, kekayaan tersebut harulah berada dalam pengendalian dan pengawasan yang ketat.

Tujuan pengawasan persediaan adalah sebagai berikut:

- a. Menjaga agar jangan sampai kehabisan persediaan.

- b. Supaya pembentukan persediaan stabil.
- c. Menghindari pembelian secara kecil-kecilan karena akan menjadi tidak efisien.
- d. Pemesanan yang ekonomis.

Persediaan dapat memiliki berbagai fungsi penting yang menambah fleksibilitas dari operasi suatu perusahaan. Menurut Render (2001, p314), ada enam penggunaan persediaan, yaitu :

1. Untuk memberikan suatu stok barang-barang agar dapat memenuhi permintaan yang diantisipasi akan timbul dari konsumen.
2. Untuk memasangkan produksi dengan distribusi. Misalnya, bisa permintaan produknya tinggi hanya dengan pada musim panas, suatu perusahaan dapat membentuk stok selama musim dingin, sehingga biaya kekurangan stok dan kehabisan stok dapat dihindari
3. Untuk mengambil keuntungan dari potongan jumlah besar dapat secara substansial menurunkan biaya produk.
4. Untuk menghindari inflasi dan perubahan harga
5. Untuk menghindari kekurangan stok yang dapat terjadi karena cuaca, kekurangan pasokan, masalah mutu atau pengiriman yang tidak tepat.
6. Untuk menjaga agar operasi dapat berlangsung dengan baik dengan menggunakan “barang-dalam-proses” dalam persediaannya.

### **2.9.3 Pengertian Penjualan**

Menurut Mulyadi (2001, p204), kegiatan penjualan terdiri dari transaksi penjualan barang atau jasa, baik secara kredit atau tunai. Dari pengertian tersebut dapat disimpulkan penjualan adalah suatu aktifitas perusahaan yang utama dalam

memperoleh pendapatan, baik untuk perusahaan besar maupun perusahaan kecil. Penjualan merupakan sasaran akhir dari kegiatan pemasaran. Pada bagian ini terjadi penetapan harga melalui perundingan dan perjanjian serah terima barang, cara pembayaran yang disepakati oleh kedua belah pihak, sehingga terjadi suatu titik kepuasan.

Fungsi yang terkait dengan sistem penjualan adalah:

1. Fungsi Penjualan

Bertanggung jawab untuk menerima order, mengedit order, meminta otorisasi kredit, menentukan tanggal pengiriman dan transaksi penjualan.

2. Fungsi Kredit

Bertanggung jawab untuk meneliti status kredit dan memberikan otorisasi kredit kepada pelanggan.

3. Fungsi Gudang

Bertanggung jawab untuk menyimpan dan menyiapkan barang yang dipesan, dan mengirimkannya ke bagian pengiriman.

4. Fungsi Pengiriman

Bertanggung jawab untuk menyerahkan barang ke pelanggan berdasarkan surat order pengiriman yang diterima dari bagian penjualan.

5. Fungsi Penagihan

Bertanggung jawab untuk melakukan pencatatan atas transaksi yang timbul dari kegiatan penjualan.

6. Fungsi Akuntansi

Bertanggung jawab untuk mencatat piutang yang timbul dari transaksi penjualan kredit dan membuat serta mengirimkan pernyataan piutang kepada para debitur, serta membuat laporan penjualan.

Sistem informasi penjualan membentuk prosedur-prosedur yang bekerja satu dengan yang lainya untuk mencapai tujuan. Prosedur-prosedur yang membentuk sistem informasi penjualan adalah :

1. Prosedur Order Penjualan

Dalam prosedur ini bagian order penjualan menerima order dari pembeli dan menambah informasi penting pada surat order. Bagian ini kemudian mengirimkan kepada bagian kredit untuk meminta persetujuan kredit. Setelah disetujui, bagian penjualan mengisi faktur penjualan dan surat order pengiriman ke bagian terkait.

2. Prosedur Persetujuan Kredit

Bagian kredit menganalisis kredit yang diajukan oleh pelanggan kemudian memberikan persetujuan atau membatalkan penjualan kredit yang diajukan.

3. Prosedur Pengiriman

Bagian pengiriman mengirimkan barang kepada pembeli sesuai dengan surat order pengiriman yang diterima dari bagian order penjualan.

4. Prosedur Penagihan

Bagian ini membuat surat penagihan dan melampirkan faktur penjualan untuk dikirimkan kepada pembeli.

5. Prosedur Pencatatan Piutang

Bagian piutang mencatat tembusan faktur penjualan ke dalam kartu piutang untuk mengadakan pengakuan piutang.

6. Prosedur Distribusi Penjualan

Bagian ini mendistribusikan data-data penjualan menurut informasi yang diperlukan manajemen.

#### 7. Prosedur Pencatatan Harga Pokok Penjualan

Dalam prosedur ini bagian kartu persediaan dan kartu biaya membuat rekap harga pokok penjualan dan bukti memorial sebagai dasar bagi bagian akuntansi untuk melakukan pencatatan harga pokok produk yang dijual.