

## BAB III

### LANDASAN TEORI

#### 1.1 Definisi Sistem

Menurut Herlambang dan Tanuwijaya (2005:116), definisi dari sistem dapat dilakukan dengan dua pendekatan, yaitu pendekatan prosedur dan pendekatan komponen. Dengan pendekatan prosedur, sistem didefinisikan sebagai kumpulan dari beberapa prosedur yang mempunyai tujuan tertentu. Sedangkan berdasarkan pendekatan komponen, sistem merupakan kumpulan dari komponen-komponen yang saling berkaitan untuk mencapai tujuan tertentu.

Menurut O'Brien (2002:8) sistem diartikan sebagai sekumpulan komponen yang berhubungan yang bekerja bersama-sama menuju tujuan umum dengan menerima *Input* dan memproduksi output dalam suatu proses perubahan yang terorganisasi.

Dalam perkembangan sistem yang ada, sistem dibedakan menjadi dua jenis, yaitu sistem terbuka dan sistem tertutup. Sistem terbuka (*open systems*) merupakan sistem yang dihubungkan dengan arus sumber daya luar dan tidak mempunyai elemen pengendali. Sedangkan sistem tertutup (*closed systems*) tidak mempunyai elemen pengontrol dan dihubungkan pada lingkungan sekitarnya.

#### 1.2 Sistem Informasi

Menurut Herlambang dan Tanuwijaya (2005:121), data sebenarnya merupakan fakta-fakta atau kejadian yang dapat berupa angka-angka atau berupa kode-kode tertentu. Data berupa angka atau wujud yang lain masih belum mempunyai arti atau kegunaan bagi penggunaannya, sehingga harus diolah

sedemikian rupa dengan menggunakan prosedur-prosedur tertentu untuk menghasilkan sebuah informasi bagi penggunanya. Secara singkat, informasi adalah data yang diolah dan mempunyai arti bagi penggunanya.

Dari penjelasan diatas, maka diperoleh kesimpulan bahwa sistem informasi didefinisikan sebagai prosedur-prosedur yang digunakan untuk mengolah data sehingga dapat digunakan oleh penggunanya.

### 1.3 Inventori

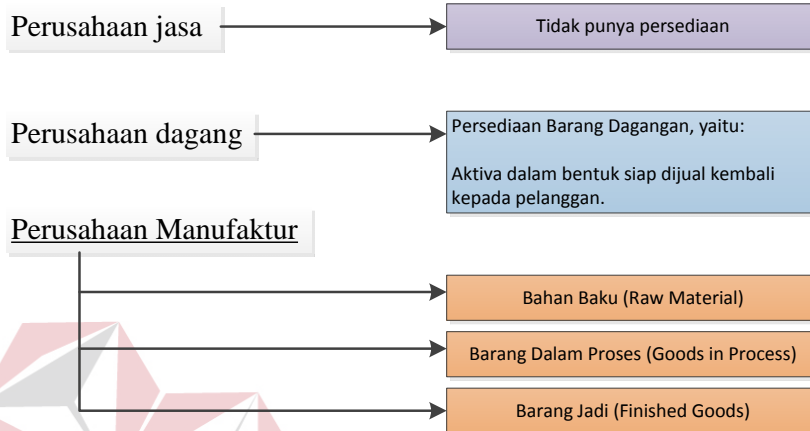
Menurut Richardus (2003:3), inventori adalah barang-barang yang biasanya dapat dijumpai di gudang tertutup, lapangan, gudang terbuka, atau tempat-tempat penyimpanan lainnya, baik berupa bahan baku, barang setengah jadi, barang jadi, barang-barang untuk keperluan operasi, atau barang-barang untuk keperluan suatu proyek.

Jenis-jenis inventori akan berbeda sesuai dengan bidang atau kegiatan normal usaha perusahaan tersebut. Berdasarkan bidang usaha perusahaan dapat terbentuk perusahaan jasa, perusahaan dagang, ataupun perusahaan industry (*manufacture*). Dijelaskan pada Gambar 1.1. di halaman. Untuk dapat memahami perbedaan dan keberadaan dari tiap-tiap jenis persediaan tersebut maka dapat dilihat dari penggolongan inventori secara garis besar, menurut Richardus (2003:5), yaitu:

1. Inventori bahan baku (*raw material*), merupakan barang-barang yang diperoleh untuk digunakan dalam proses.
2. Inventori bahan dalam proses (*goods in process*), yang juga disebutkan sebagai pekerjaan dalam proses (*work in process*) terdiri dari bahan baku yang sebagian telah diproses dan perlu dikerjakan lebih lanjut.

Inventori barang jadi (*finished goods*), merupakan produk/barang yang telah selesai diproduksi dan menjadi persediaan perusahaan untuk dijual atau digunakan untuk operasi perusahaan.

**Penggolongan persediaan:**



Gambar 1.1 Klasifikasi Inventori.

#### 1.4 Konsep Analisis dan Perancangan Sistem

Menurut McLeod and Schell (2001:128), analisis sistem adalah penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem baru atau diperbaharui atau merancang sistem yang baru. Didalam tahap analisis sistem, analisis sistem terus bekerja sama dengan manajer, komite pengarah sistem informasi manajemen yang terlibat dalam titik-titik yang penting. Analisis sistem juga dilakukan dengan tujuan untuk mengidentifikasi dan mengevaluasi permasalahan yang terjadi dan kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikannya.

Menurut McLeod dan Schell (2001:192), perancangan sistem merupakan sebuah penentuan proses dan data yang diperlukan oleh sistem yang baru.

Jika sistem itu berbasis komputer, maka rancangan dapat menyertai spesifikasi jenis peralatan yang akan digunakan.

Selanjutnya, menurut Kendall dan Kendall (2003:7), analisis dan perancangan sistem digunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang bisa dicapai melalui penggunaan sistem informasi terkomputerisasi.

### 1.5 Data Flow Diagram (DFD)

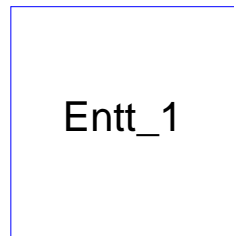
Menurut Kendall dan Kendall (2003:241), *data flow diagram* (DFD) menggambarkan pandangan sejauh mungkin mengenai masukan, proses dan keluaran sistem, yang berhubungan dengan masukan, proses, dan keluaran dari model sistem yang dibahas. Serangkaian diagram alir data berlapis juga bisa digunakan untuk merepresentasikan dan menganalisis prosedur-prosedur mendetail dalam sistem. Prosedur-prosedur tersebut yaitu konseptualisasi bagaimana data-data berpindah di dalam organisasi, proses-proses atau transformasi dimana data-data melalui, dan apa keluarannya. Jadi, melalui suatu teknik analisa data terstruktur yang disebut *data flow diagram* (DFD), penganalisis sistem dapat merepresentasikan proses-proses data di dalam organisasi.

Menurut Kendall dan Kendall (2003:265), dalam memetakan *data flow diagram* (DFD), terdapat beberapa simbol yang digunakan antara lain:

#### 1. *External Entity*

Suatu *External entity* atau entitas merupakan orang, kelompok, departemen, atau sistem lain di luar sistem yang dibuat dapat menerima atau

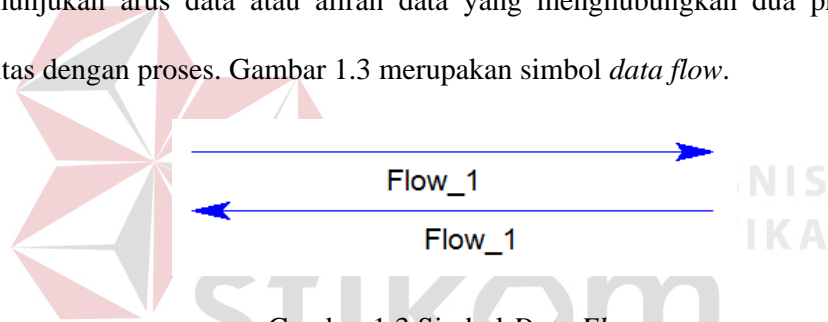
memberikan informasi atau data ke dalam sistem yang dibuat. Gambar 1.2 merupakan simbol entitas dalam DFD dalam model *gene and sarson*.



Gambar 1.2 Simbol *External Entity*

## 2. *Data Flow*

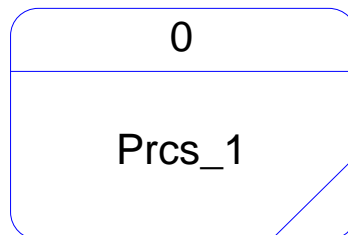
*Data flow* atau aliran data disimbolkan dengan tanda panah. Aliran data menunjukkan arus data atau aliran data yang menghubungkan dua proses atau entitas dengan proses. Gambar 1.3 merupakan simbol *data flow*.



Gambar 1.3 Simbol *Data Flow*

## 3. *Process*

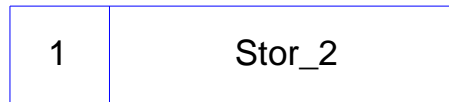
Suatu proses dimana beberapa tindakan atau sekelompok tindakan dijalankan. Gambar 1.4 merupakan simbol *process*.



Gambar 1.4 Simbol *Process*

#### 4. *Data Store*

*Data Store* adalah simbol yang digunakan untuk melambangkan proses penyimpanan data. Gambar 1.5 merupakan simbol *data store*/Penyimpanan data.



Gambar 1.5 Simbol *Data Store*

### 1.6 Entity Relationship Diagram (ERD)

*Entity relationship diagram* (ERD) adalah gambaran pada sistem dimana di dalamnya terdapat hubungan antara *entity* beserta relasinya. *Entity* merupakan sesuatu yang ada dan didefinisikan di dalam suatu organisasi, dapat abstrak dan nyata. Untuk setiap *entity* biasanya mempunyai *attribute* yang merupakan ciri *entity* tersebut. *Attribute* yaitu uraian dari entitas dimana mereka dihubungkan atau dapat dikatakan sebagai *identifier* atau *descriptors* dari entitas. Sedangkan entitas (*entities*) menurut Simarmata (2007:97) adalah objek data prinsip tentang informasi yang dikumpulkan.

Entitas digolongkan menjadi *independent* atau *dependent entity*. *Independent entity* adalah apa yang tidak bersandar pada yang lain sebagai identifikasi. Suatu *dependent entity* adalah apa yang bersandar pada yang lain sebagai identifikasi. Selain digolongkan menjadi *independent* atau *dependent entity*, terdapat jenis-jenis entitas khusus yaitu:

#### 1. *Associateive Entity*

*Associative entity* (dikenal juga sebagai *intersection entity*) adalah entitas yang digunakan oleh rekanan dua entitas atau lebih untuk menyatukan suatu hubungan –ke –banyak (*Many to Many*) .

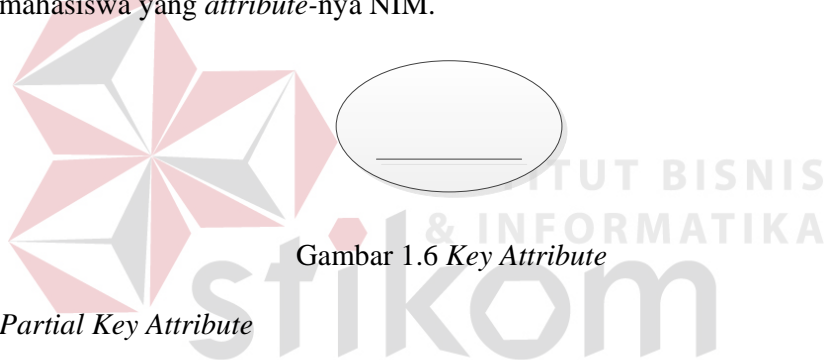
## 2. *Subtypes entity*

*Subtypes entity* digunakan di dalam hierarki generalisasi (*generalization hierarchies*) untuk menyajikan suatu subset kejadian dari entitas orangtua, yang disebut *supertype*, tetapi yang memiliki atribut atau hubungan yang berlaku hanya untuk subset.

Menurut Marlinda (2004:28), *attribute* sebagai kolom disebuah relasi mempunyai macam-macam jenis *attribute* yaitu:

### 1. *Key Attribute*

Adalah *attribute* unik dan tidak dimiliki oleh *attribute* lainnya, misalnya *entity* mahasiswa yang *attribute*-nya NIM.

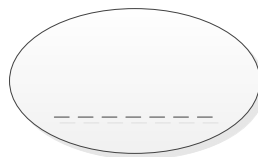


Gambar 1.6 *Key Attribute*

### 2. *Partial Key Attribute*

Adalah *attribute* yang tidak menjadi atau merupakan anggota dari *Key primer*.

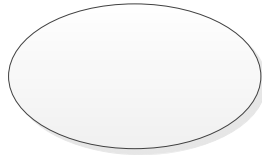
Misal antara cabang (toko) dan kode cabang.



Gambar 1.7 *Particial Key Attribute*

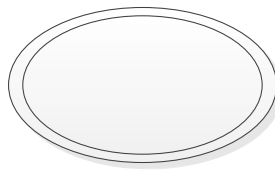
### 3. *Single Value Attribute*

*Attribute* yang hanya memiliki satu nilai harga, misalnya *entity* mahasiswa dengan *attribute*-nya umur (tanggal lahir).

Gambar 1.8 *Single Value Attribute*

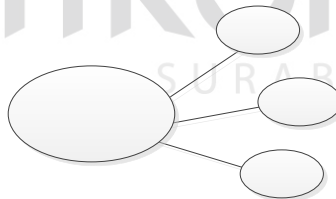
#### 4. *Multi Value Attribute*

*Attribute* yang banyak memiliki nilai harga, misalnya *entity* mahasiswa dengan *attribute*-nya pendidikan (SD,SMP,SMA).

Gambar 1.9 *Multi Value Attribute*

#### 5. *Composite Attribute*

*Attribute* yang memiliki dua harga, misalnya nama besar (nama kerja) dan nama kecil (nama asli).

Gambar 1.10 *Composite Attribute*

#### 6. *Derived Attribute*

*Attribute* yang nilai-nilainya diperoleh dari pengolahan atau dapat diturunkan dari tabel *attribute* atau dari tabel lain yang berhubungan.

Gambar 1.11 *Derived Attribute*



Menurut Simarmata (2007:96), model *Entity-Relationship* (ER) pertama kali diusulkan oleh Peter pada tahun 1976 sebagai cara untuk mempersatukan pandangan basis data jaringan relasional. Langkah sederhana dari model ER adalah model data konseptual yang memandang dunia nyata sebagai kesatuan (*entities*) dan hubungan (*relationships*).

Menurut Simarmata (2007:98), *Entity Relationship Diagram* (ERD) mengilustrasikan struktur logis dari basis data yang mempunyai metodologi sebagai berikut:

Tabel 1.1 Ilustrasi Pembuatan ERD.

1. Menentukan Entitas	Menentukan peran, kejadian, lokasi, hal nyata, dan konsep di mana pengguna akan menyimpan data.
2. Menentukan relasi	Tentukan hubungan antara pasangan entitas menggunakan matriks relasi.
3. Gambar ERD Sementara	Entitas digambar dengan kotak dan relasi dengan garis yang menghubungkan entitas.
4. Isi Kardinalitas	Tentukan jumlah kejadian dari satu entitas untuk sebuah kejadian pada entitas yang berhubungan.
5. Tentukan Kunci Utama	Tentukan atribut yang mengidentifikasi satu dan hanya satu kejadian pada masing-masing entitas.

6. Gambar ERD berdasar Kunci	Hilangkan relasi Many-toMany dan masukkan primari dan kunci foreign pada masing-masing entitas.
7. Menentukan Atribut	Tuliskan field-field yang diperlukan oleh sistem.
8. Pemetaan Atribut	Pasangkan atribut dengan satu entitas yang sesuai pada masing-masing atribut.
9. Gambar ERD dengan Atribut	Aturlah ERD dari langkah 6 dengan menambahkan entitas atau relasi yang ditemukan pada langkah 8.
10. Periksa Hasil	Apakah ERD sudah menggambarkan sistem yang akan dibangun.

*Entity relationship diagram* (ERD) ini diperlukan untuk menggambarkan hubungan antar *entity* dengan jelas, dapat menggambarkan batasan jumlah *entity* dan partisipasi antar *entity*, mudah dimengerti pemakai dan mudah disajikan oleh perancang *database*. Untuk itu, *entity relationship diagram* dibagi menjadi dua jenis model, yaitu:

1. *Conceptual Data Model*

*Conceptual data model* (CDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara konseptual.

2. *Physical Data Model*

*Physical data model* (PDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara fisikal.

## 1.7 Database

Menurut Connolly (2005:15) *database* adalah sebuah koleksi data yang dipakai bersama dan terhubung secara logis dan deskripsi dari data ini dirancang untuk memenuhi kebutuhan informasi dari sebuah organisasi.

Menurut Herlambang dan Tanuwijaya (2005:77), Pengolahan data (*Database*) adalah penggunaan perangkat lunak untuk memelihara, memanipulasi dan menampilkan (*query*) data-data yang diperlukan untuk dicetak dalam bentuk laporan. Operator menggunakan *keyboard* suatu terminal atau *computer mikro* untuk memelihara (*maintenance*) data-data yang dibutuhkan, yang ditampilkan di layar. Data-data tersebut dapat disimpan dalam penyimpanan sekunder dan diolah sesuai dengan kebutuhan laporan yang ingin dicetak. Demikian menurut Yuswanto (2005:2), *database* merupakan sekumpulan data yang berisi informasi yang saling berhubungan. Pengertian ini sangat berbeda antara *database* relasional dan non-relasional. Pada *database* non-relasional sebuah *database* hanya merupakan *file*.

Menurut Marlinda (2004:1), *database* adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu, menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan oleh pemakainya. *Database* digunakan untuk mengatasi masalah-masalah yaitu redundansi data, kesulitan pengaksesan data, isolasi data untuk standarisasi, *multiple user*, masalah keamanan, masalah integrasi, dan masalah data *independence*.

Maka dapat disimpulkan bahwa *database* merupakan kumpulan data dalam *file-file* yang terdiri dari *atribut-atribut*, *entity-entity*, dan *relationship* dari informasi sebuah perusahaan yang dirancang untuk memenuhi kebutuhan informasi dari sebuah organisasi.

## 1.8 SQL Server 2008

Menurut Nugroho (2008:1), SQL Server 2008 merupakan DBMS (*Database Management System*) yang handal dalam mengolah data dengan disertai *User Interface* yang cukup mudah untuk digunakan. Microsoft sebagai vendor dari SQL Server terus mengembangkan produk mereka yang satu ini untuk menjadikannya sebagai *Tools database* terbaik.

Jika dilihat dari tampilannya, SQL Server 2008 tidak berbeda jauh dengan SQL Server 2005, dibandingkan dengan SQL Server 2000 ke 2005 yang mengalami perubahan signifikan dengan menyatukan *Enterprise Manager* dan *Query Analyzer*. Menjadi satu, hanya saja di SQL Server 2008, Microsoft mengembangkan beberapa fitur yang telah ada di produk SQL Server sebelumnya dan menambah beberapa fitur baru untuk meningkatkan *performance*. Beberapa fitur baru yaitu:

1. *Data Compression*
2. *Change Data Capture*
3. *Filtered Indexes*
4. *Table-Valued parameter*
5. *Sparse column*
6. Data type baru (seperti Date, Time dan Filestream).

## 1.9 Visual Basic .NET 2010

Menurut Yuswanto dan Subari (2010:1), Visual Basic.NET 2010 adalah bahasa pemrograman yang tergabung dalam Microsoft Visual Studio 2010. Microsoft Visual Studio 2010 diperkenalkan pertama kali ke masyarakat umum pada pertengahan Mei 2009.

Dalam Visual Studio 2010 diperkenalkan beberapa kelebihan, diantaranya:

1. Teknologi yang ada mendukung “*parallel programming*” untuk manajemen developer dengan hadirnya fitur proyek manajemen, *work item tracking*, *simple server reporting service*, *version control*.
2. Visual Studio 2010 sudah mendukung analisis dan desain UML bukan hanya *coding*, *compile*, dan *system*.
3. Visual Studio 2010 dapat bekerja dengan baik pada *platform* Windows untuk 32 bit dan 64 bit khusus Vista.
4. Visual Studio 2010 dan Microsoft .NET *Framework* 4.0 membantu *developer* menghasilkan performansi yang lebih baik dan menghasilkan aplikasi-aplikasi *scalable*.
5. Pada pemrograman *database*, Visual Studio 2010 mampu bekerja dengan baik dengan program IBM DB2, Oracle *Database*, apalagi dengan Microsoft SQL Server.