

## **BAB III**

### **LANDASAN TEORI**

Pada bab ini akan dijelaskan dasar-dasar teori yang berhubungan dengan permasalahan yang dibahas. Hal ini sangat penting karena teori-teori tersebut digunakan sebagai landasan pemikiran dalam kerja praktek ini.

#### **3.1 Konsep Dasar Sistem Informasi**

Informasi sangat penting bagi suatu organisasi. Menurut Hartono (1990:8) informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan berarti bagi yang menerimanya. Informasi dapat dihasilkan dari sistem informasi atau disebut juga *processing system* atau *information processing system* atau *information generation system*.

Menurut Robert A. Leitch dan K. Roscoe Davis dalam Hartono (1990:11), sistem informasi adalah suatu sistem 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. Laporan tersebut dapat digunakan sebagai informasi untuk mengambil keputusan.

### 3.1.1 Sistem

Menurut Herlambang (2005:116), definisi sistem dapat dibagi menjadi dua pendekatan, yaitu pendekatan secara prosedur dan pendekatan secara komponen. Berdasarkan pendekatan prosedur, sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem. Prosedur (*procedure*) itu sendiri didefinisikan oleh Richard F. Neuschel sebagai berikut:

“Prosedur adalah suatu urutan operasi klerikal (tulis-menulis), biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi.”

Sedangkan berdasarkan pendekatan komponen, sistem merupakan kumpulan dari komponen-komponen yang saling berkaitan untuk mencapai tujuan tertentu.

Dalam perkembangan sistem yang ada, sistem dibedakan menjadi dua jenis, yaitu sistem terbuka dan sistem tertutup. Sistem terbuka merupakan sistem yang dihubungkan dengan arus sumber daya luar dan tidak mempunyai elemen pengendali, sedangkan sistem tertutup tidak mempunyai elemen pengontrol dan dihubungkan pada lingkungan sekitarnya.

Suatu kesatuan yang terdiri dari dua atau lebih komponen yang saling berinteraksi untuk mencapai suatu tujuan tertentu.

Jenis-jenis Sistem Menurut tingkatannya :

1. ESS, GDSS, CSCWS
2. Sistem Ahli DSS, SIM
3. Knowledge Work Systems, Office Automation Systems
4. Transaction Processing Systems

### **3.1.2 Sistem Informasi**

Robert A Szymanski dkk, dalam bukunya “Computer & Information System”

Sistem informasi adalah sekumpulan fungsi yang bekerja secara bersama-sama dalam mengelola : pengumpulan, penyimpanan, pemrosesan, serta pendistribusian informasi.

Menurut Burch dan Strater dalam bukunya “Information System :Theory and Practise” System Informasi adalah suatu kumpulan fungsi-fungsi yang bergabung secara formal dan secara sistematis: (a) melaksanakan pengelolaan data transaksi operasional, (b) menghasilkan informasi untuk mendukung manajemen dalam melaksanakan aktifitas perencanaan, pengendalian dan pengambilan keputusan, (c) menghasilkan berbagai laporan bagi kepentingan eksternal organisasi.

### **3.2 Analisa dan Perancangan Sistem**

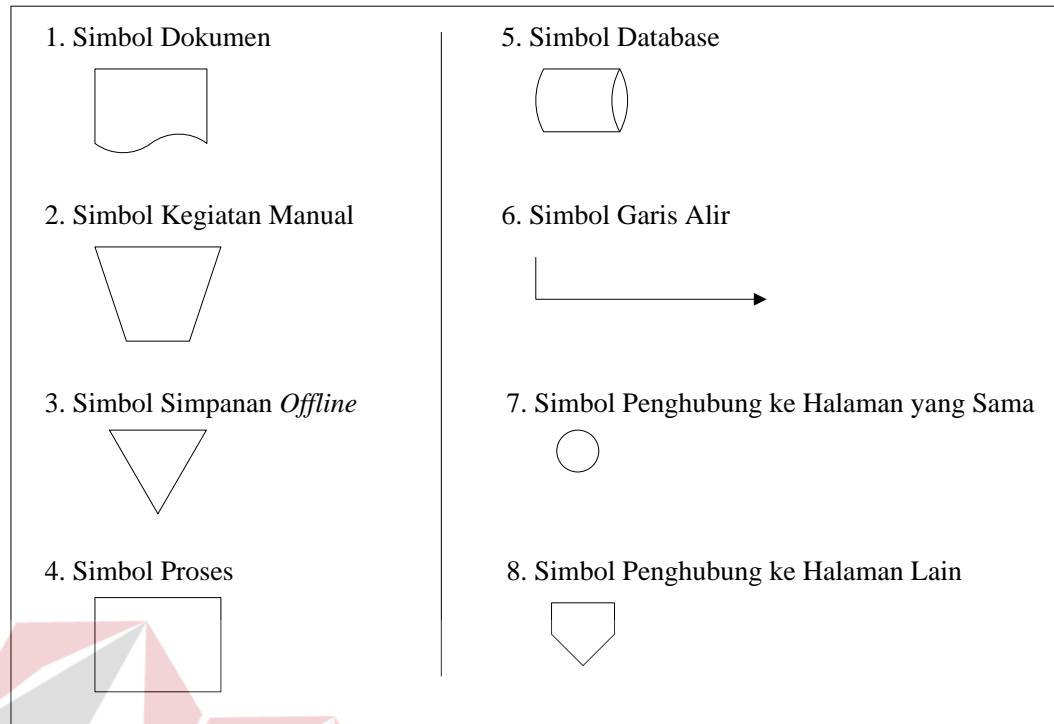
Tujuannya untuk memperbaiki berbagai fungsi di dalam sistem agar lebih efisien, untuk mengubah sasaran sistem, untuk mengganti output, untuk mencapai tujuan yang sama dengan seperangkat input yang lain atau untuk melakukan beberapa perbaikan serupa.

Analisis sistem dilakukan dengan tujuan untuk dapat mengidentifikasi dan mengevaluasi permasalahan yang terjadi dan kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikannya.

Perancangan sistem merupakan penguraian suatu sistem informasi yang utuh ke dalam bagian komputerisasi yang dimaksud, mengidentifikasi dan mengevaluasi permasalahan, menentukan kriteria, menghitung konsistensi terhadap kriteria yang ada, serta mendapatkan hasil atau tujuan dari masalah tersebut serta mengimplementasikan seluruh kebutuhan operasional dalam membangun aplikasi.

### **3.2.1 System Flow**

*System flow* atau bagan alir sistem merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. System flow menunjukkan urutan-urutan dari prosedur yang ada di dalam sistem dan menunjukkan apa yang dikerjakan sistem. Simbol-simbol yang digunakan dalam system flow ditunjukkan pada Gambar 3.1 di halaman 24.



Gambar 3.1 Simbol-simbol pada System Flow

1. Simbol dokumen

Menunjukkan dokumen input dan output baik untuk proses manual atau komputer.

2. Simbol kegiatan manual

Menunjukkan pekerjaan manual.

3. Simbol simpanan offline

Menunjukkan file non-komputer yang diarsip.

4. Simbol proses

Menunjukkan kegiatan proses dari operasi program komputer.

5. Simbol database

Menunjukkan tempat untuk menyimpan data hasil operasi komputer.

#### 6. Simbol garis alir

Menunjukkan arus dari proses.

#### 7. Simbol penghubung

Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain.

### 3.2.2 Data Flow Diagram (DFD)

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur dan dapat mengembangkan arus data di dalam sistem dengan terstruktur dan jelas.

DFD fokus pada aliran data dari dan ke dalam sistem serta memproses data tersebut (Kendall, 2003:241).

Simbol-simbol dasar dalam DFD antara lain :

#### a. *Eksternal Entity*

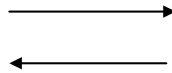
Suatu *Eksternal Entity* atau entitas merupakan orang, kelompok, departemen, atau sistem lain di luar sistem yang dibuat dapat menerima atau memberikan informasi atau data ke dalam sistem yang dibuat. Gambar 3.2 merupakan simbol entitas dalam DFD dalam model Gane dan Sarson.



Gambar 3.2 Simbol Eksternal Entity

*b. Data Flow*

*Data Flow* atau aliran data disimbolkan dengan tanda panah. *Data Flow* menunjukkan arus data atau aliran data yang menghubungkan dua proses atau entitas dengan proses. Gambar 3.3 merupakan simbol *Data Flow*.



Gambar 3.3 Simbol Data Flow

*c. Process*

Suatu Proses dimana beberapa tindakan atau sekelompok tindakan dari arus data yang masuk untuk dijalankan atau diproses agar menghasilkan arus data yang akan keluar dari proses. Gambar 3.4 merupakan simbol *Process*.



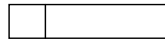
Gambar 3.4 Simbol Process

*d. Data Store*

*Data Store* adalah simbol yang digunakan untuk melambangkan proses penyimpanan data. Suatu nama perlu diberikan pada data store untuk menunjukkan nama dari filenya. Gambar 3.5 pada halaman 27 merupakan simbol file penyimpanan/*data store* yang dapat berupa hal-hal sebagai berikut, sebagai gambaran:

1. Suatu file atau *database* di sistem komputer.
2. Suatu arsip atau catatan manual.

3. Suatu tabel acuan manual.



Gambar 3.5 Simbol Data Store

### A. Context Diagram

*Context Diagram* merupakan langkah pertama dalam pembuatan *Data Flow Diagram*. Pada *context diagram* dijelaskan sistem apa yang dibuat dan *entity* apa saja yang digunakan. Dalam *context diagram* harus ada arus data yang masuk dan arus data yang keluar.

### B. Data Flow Diagram Level 0

DFD level 0 adalah langkah selanjutnya setelah *context diagram*. Hal yang digambarkan dalam Diagram level 0 ini adalah proses utama dari sistem serta hubungan *Entity*, *Proses*, *alur data* dan *data store*.

### C. Data Flow Diagram Level 1

DFD Level 1 merupakan penjelasan dari DFD level 0. Pada proses ini dijelaskan proses apa saja yang dilakukan pada setiap proses yang terdapat di DFD level 0.

### 3.2.3 Entity Relational Diagram

*Entity Relationship Diagram* (ERD) adalah gambaran pada sistem dimana di dalamnya terdapat hubungan antara *entity* beserta relasinya. *Entity* merupakan sesuatu yang ada dan terdefiniskan di dalam suatu organisasi, dapat abstrak dan nyata. Untuk setiap *entity* biasanya mempunyai *atribute* yang merupakan ciri *entity* tersebut. Menurut Marlinda (2004:28), *Atribute* adalah kolom di sebuah relasi. Macam-macam *atribute* yaitu:



a. *Simple Attribute*

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

b. *Composite Attribute*

*Composite attribute* adalah *attribute* yang memiliki dua nilai harga, misalnya nama besar (nama keluarga) dan nama kecil (nama asli).

c. *Single Value Attribute*

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

d. *Multi Value Attribute*

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

e. *Null Value Attribute*

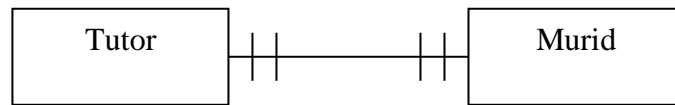
*Null value attribute* adalah *attribute* yang tidak memiliki nilai harga, misalnya *entity* tukang becak dengan *attribute*-nya pendidikan (tanpa memiliki ijazah).

Sedangkan relasi adalah hubungan antar *entity* yang berfungsi sebagai hubungan yang mewujudkan pemetaan antar *entity*. Macam-macam relasi itu sendiri antara lain:

1. *One to One (1:1)*

Relasi dari *entity* satu dengan *entity* dua adalah satu berbanding satu.

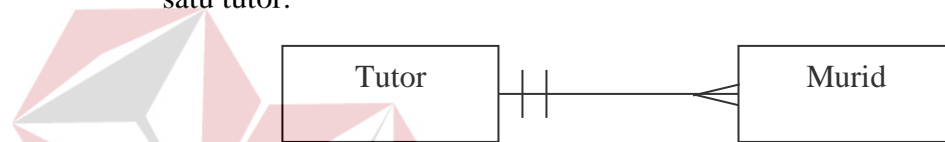
Contoh: Pada pelajaran privat, satu tutor mengajar satu murid dan satu murid hanya diajar oleh satu tutor.

Gambar 3.6 Relasi *One to One*

## 2. *One to Many* (1: m)

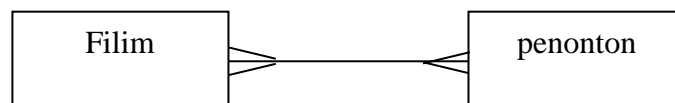
Relasi antara *entity* yang pertama dengan *entity* yang kedua adalah satu berbanding banyak atau dapat pula dibalik, banyak berbanding satu.

Contoh: Satu tutor mengajar banyak murid dan banyak murid diajar oleh satu tutor.

Gambar 3.7 Relasi *One to Many*

## 3. *Many to Many*

Relasi antara *entity* yang satu dengan *entity* yang kedua adalah banyak berbanding banyak. Contoh: Pada bioskop, satu film ditonton banyak penonton dan satu penonton melihat banyak film pula.

Gambar 3.8 Relasi *Many to Many*

*Entity Relationship Diagram* ini diperlukan agar dapat 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:

a. *Conceptual Data Model (CDM)*

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

b. *Physical Data Model (PDM)*

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

