

## BAB III

### LANDASAN TEORI

#### 3.1 Pengertian Sistem Informasi

Menurut (Wilkinson, 2007:3-4) Sistem informasi berasal dari dua kata yang saling berhubungan yaitu antara sistem dan informasi. Sistem adalah suatu kerangka kerja yang sangat terpadu serta mempunyai satu sasaran atau lebih. Informasi berbeda dengan data, data adalah keadaan yang ada dan belum diproses belum lanjut, sedangkan informasi adalah data-data yang telah diproses dan dibentuk sebagaimana mungkin agar lebih bernilai bagi penggunanya. Maka sistem informasi adalah suatu kerangka kerja dimana sumber daya manusia dan teknologi dikoordinasikan untuk mengubah *input* (data) menjadi *output* (informasi) guna mencapai sasaran perusahaan.

Menurut Herlambang (2008:121), data adalah fakta-fakta atau kejadian-kejadian yang dapat berupa angka atau kode tertentu. Data belum mempunyai arti bagi penggunanya, sehingga harus diolah sedemikian rupa hingga menjadi suatu informasi. Secara ringkas, informasi adalah data yang telah diolah dan mempunyai arti bagi penggunanya.

#### 3.2 Konsep Sistem Informasi

Menurut Herlambang (2005:47) sistem informasi terdiri dari *input*, proses dan *output*. Pada proses terdapat hubungan timbal balik dengan dua elemen yaitu control dari kinerja sistem dan sumber-sumber penyimpanan data. *Input* yang akan di proses berupa data, baik berupa karakter-karakter huruf maupun berupa *numeric*. Saat ini data bisa berupa suara atau audio maupun gambar atau

*video*. Data ini diproses dengan metode-metode tertentu dan akan menghasilkan *output* yang berupa informasi. Informasi yang dihasilkan bisa berupa laporan atau *report* maupun solusi dari proses yang telah di jalankan.

### 3.3 Analisa dan Perancangan Sistem

Menurut Jogiyanto (1999:129) analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya. Tahap analisis dilakukan setelah tahap perencanaan sistem dan sebelum tahap desain sistem.

### 3.4 Data Flow Diagram

*Data Flow Diagram* (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alir data baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama *Bubble Chart* atau *diagram*, model proses, diagram alur kerja atau model fungsi. DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks daripada data yang digunakan untuk menjelaskan aliran informasi dan transformasi data yang bergerak dari pemasukan data hingga keluaran.

Untuk memudahkan pembacaan DFD, maka penggambaran DFD disusun berdasarkan tingkatan atau level dari atas ke bawah, yaitu:

## 1. Diagram Konteks

Merupakan diagram paling atas yang terdiri dari suatu proses dan menggambarkan ruang lingkup proses. Hal yang digambarkan dalam diagram konteks adalah hubungan terminator dengan sistem dan juga sistem dalam suatu proses. Sedangkan hal yang tidak digambarkan dalam diagram konteks adalah hubungan antar terminator dan data store.

## 2. Diagram Zero (Level 0)

Merupakan diagram yang berada diantara Diagram Konteks dan Diagram Detail serta menggambarkan proses utama dari DFD. Hal yang digambarkan dalam *Diagram Zero* adalah proses utama dari sistem serta hubungan *Entity*, Proses, alur data dan data store.

## 3. Diagram Detail (Primitif)

Merupakan penguraian dalam proses yang ada dalam *Diagram Zero*. Diagram yang paling rendah dan tidak dapat diuraikan lagi.

*Data Flow Diagram* (DFD) memiliki tiga komponen, yaitu:

### 1. Terminator atau External Entity atau Kesatuan Luar

Terminator mewakili entitas *external* yang berkomunikasi dengan sistem yang sedang dikembangkan. Terminator merupakan kesatuan di lingkungan sistem. Yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luar *system* yang akan memberikan *input* maupun *output* dari sistem. Biasanya terminator ini dikenal dengan nama entitas (*external*), sumber atau tujuan (*source and sink*). Terminator dapat juga berupa departemen, divisi atau sistem diluar sistem yang berkomunikasi dengan

sistem yang dikembangkan. Ada tiga hal penting yang harus di ingat tentang terminator:

- a. Terminator merupakan bagian atau lingkungan luar sistem. Alur data yang menghubungkan terminator dengan berbagai proses *system* menunjukkan hubungan sistem dengan dunia luar.
- b. Profesional sistem tidak dapat mengubah isi atau cara kerja, organisasi atau prosedur yang berkaitan dengan terminator.
- c. Hubungan yang ada antar terminator yang satu dengan yang lain tidak dapat digambarkan pada DFD.

## 2. Proses

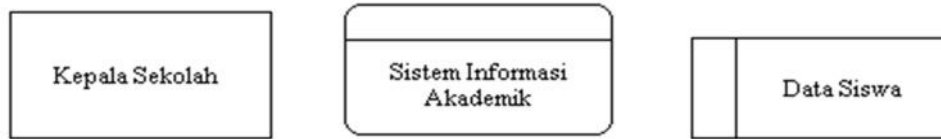
Proses sering dikenal dengan nama *Bubble*, fungsi atau informasi. Komponen proses menggambarkan bagian dari sistem. Yang mentransformasikan *input* ke *output*, atau dapat dikatakan bahwa komponen proses menggambarkan transformasi satu *input* atau lebih menjadi *output*. Dilambangkan dengan lingkaran atau empat persegi panjang tegak dengan sudut tumpul.

## 3. Data Store (Penyimpanan Data)

Data store digunakan sebagai saran untuk pengumpulan data. Data store disimbolkan dengan dua garis horizontal yang paralel dimana tertutup pada salah satu ujungnya atau dua garis horizontal. Suatu nama perlu diberikan pada data store menunjukkan nama dari filenya.

Data store ini biasanya berkaitan dengan penyimpanan seperti: file atau database yang berkaitan dengan penyimpanan secara komputerisasi,

contohnya: file pita magnetik, file disket dan file *hard disk*. Data Store juga berkaitan dengan penyimpanan data.



**Gambar 3. 1** Simbol-simbol dalam DFD

### 3.5 Entity Relationship Diagram

*Entity Relationship Diagram* (ERD) adalah sebuah diagram yang digunakan untuk menganalisa dan mengidentifikasi semua data, properti data dan hubungan antar data tersebut kemudian menggambarannya dalam suatu diagram.

*Entity Relationship Diagram* (ERD) digunakan untuk menginterpretasikan, menentukan dan mendokumentasikan kebutuhan sistem dalam proses database.

ERD ialah sebuah pemodelan untuk mendesain database yang baik karena tanpa ERD, bisa dipastikan pembuatan database berjalan lama dan tidak teratur (Budiharto, 2005). ERD menyediakan bentuk untuk menunjukkan struktur keseluruhan kebutuhan data dari aplikasi. Adapun elemen dari ERD ini adalah:

1. Entitas, adalah sesuatu yang dapat diidentifikasi di dalam lingkup pemakai, sesuatu yang penting bagi pemakai dari sistem yang akan dikembangkan.
2. Atribut, entitas memiliki atribut yang berfungsi untuk menjelaskan karakteristik dari entitas.

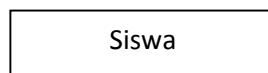
3. Identifikasi, data-data entitas memiliki nama yang berfungsi untuk mengidentifikasikan mereka. Sebuah identifikasi dapat bersifat unik atau tidak unik. (Kendall & Kendall, 2003:46)

Hubungan atau relasi berfungsi untuk menunjukkan hubungan satu entitas dengan entitas yang lain. Hubungan ini boleh memiliki atribut. Banyaknya entitas dalam suatu relasi menunjukkan tingkat dari relasi yang bersangkutan, namun yang banyak digunakan dalam aplikasi-aplikasi adalah model yang menggunakan relasi tingkat dua atau yang disebut dengan hubungan biner. Hubungan biner ini memiliki tipe yaitu hubungan biner satu ke satu, hubungan biner satu ke banyak, hubungan biner banyak ke banyak dan hubungan biner banyak ke satu (Kendall & Kendall, 2002:52).

Berikut ini komponen-komponen dalam ERD:

1. Entity

*Entity* adalah segala hal nyata maupun abstrak yang berhubungan dengan masukan dan keluaran data. Contoh: Siswa, matapelajaran, dan sebagainya.



**Gambar 3. 2** Simbol Entity

2. Attribute

*Attribute* adalah identifikasi dari suatu entitas atau entity. Contoh: *Entity* mahasiswa mempunyai attribute NIM, Nama, dan seterusnya.

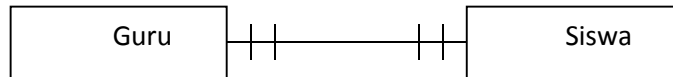
3. Relation

*Relation* adalah gambaran dari hubungan natural antara sebuah *entity* dengan *entity* lainnya. Macam-macam relation antara lain:

a. One to One (1:1)

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

Contoh: Pada pelajaran privat, satu guru mengajar satu siswa dan satu siswa hanya diajar oleh satu guru.



**Gambar 3. 3** Relasi One to One

b. One to Many (1:m)

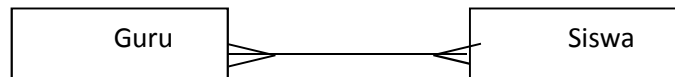
Relasi antara *entity* yang pertama dengan *entity* yang kedua adalah satu berbanding banyak atau dapat pula dibalik, banyak berbanding satu. Contoh: Pada sekolah, satu guru mengajar banyak siswa dan banyak siswa diajar oleh satu guru.



**Gambar 3. 4** Relasi One to Many

c. Many to Many

Relasi antara *entity* yang satu dengan *entity* yang kedua adalah banyak berbanding banyak. Contoh: Pada perkuliahan, satu dosen mengajar banyak mahasiswa dan satu mahasiswa diajar oleh banyak dosen pula.



**Gambar 3. 5** Relasi Many to Many