

## **BAB III**

### **LANDASAN TEORI**

Pada bab ini akan dijelaskan dasar-dasar teori yang berhubungan dengan permasalahan yang dibahas dan juga menjelaskan aplikasi-aplikasi yang digunakan pada kerja praktek ini. Hal ini sangat penting karena teori-teori tersebut digunakan sebagai landasan pemikiran dalam kerja praktek ini, adapun teori-teori yang digunakan adalah sebagai berikut:

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

Konsep dasar dari Sistem Informasi terbagi atas dua pengertian. Yang pertama adalah sistem, dan yang kedua adalah sistem informasi itu sendiri.

##### **3.1.1. Sistem**

Definisi sistem dapat dibagi menjadi dua pendekatan, yaitu pendekatan secara prosedur dan pendekatan secara komponen. Berdasarkan 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.

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. (Herlambang, 2005)

### **3.1.2. Sistem informasi**

Data adalah fakta-fakta atau kejadian-kejadian yang dapat berupa angka-angka atau kode-kode tertentu. Data masih belum mempunyai arti bagi penggunanya. Untuk dapat mempunyai arti data diolah sedemikian rupa sehingga dapat digunakan oleh penggunanya. Hasil pengolahan data inilah yang disebut sebagai informasi. Secara ringkas, Informasi adalah data yang telah diolah dan mempunyai arti bagi penggunanya. Sehingga sistem informasi dapat didefinisikan sebagai prosedur-prosedur yang digunakan untuk mengolah data sehingga dapat digunakan oleh penggunanya. (Herlambang, 2005).

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

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.

Menurut (Kendall, 2003), Analisa dan Perancangan Sistem dipergunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi terkomputerisasi.

### 3.2.1 System Flow

*System flow* adalah bagan yang menunjukkan arus pekerjaan secara menyeluruh dari suatu sistem dimana bagan ini menjelaskan urutan prosedur-prosedur yang ada dalam sistem dan biasanya dalam membuat *system flow* sebaiknya ditentukan pula fungsi-fungsi yang melaksanakan atau bertanggung jawab terhadap sub-sistem yang ada (Jogiyanto, 1998).

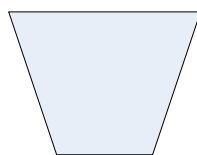
Terdapat berbagai macam bentuk simbol yang digunakan untuk merancang sebuah desain dari sistem, diantaranya adalah *terminator*, *manual operation*, *document*, *process*, *database*, *manual input*, *decision*, *off-line storage*, *on-page reference*, dan *off-page reference*.

*Terminator* merupakan bentuk simbol yang digunakan sebagai tanda dimulainya jalan proses sistem ataupun tanda akhir dari sebuah pengerjaan suatu sistem. Bentuk dari *terminator* adalah sebagai berikut:



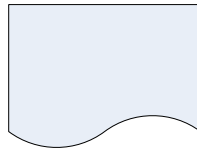
Gambar 3.1. *Simbol Terminator*

*Manual operation* digunakan untuk menggambarkan sebuah proses kerja yang dilakukan tanpa menggunakan komputer sebagai medianya (menggunakan proses manual). Bentuk simbolnya adalah:



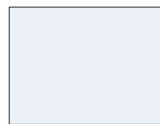
Gambar 3.2. *Simbol Manual Operation*

*Document* merupakan simbol dari dokumen yang berupa kertas laporan, surat-surat, memo, maupun arsip-arsip secara fisik. Bentuk dari *document* di gambarkan dalam simbol berikut:



Gambar 3.3. *Simbol Document*

*Process* adalah sebuah bentuk kerja sistem yang dilakukan secara terkomputerisasi. *Process* disimbolkan dengan gambar:



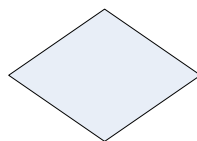
Gambar 3.4. *Simbol Process*

*Database* digunakan sebagai media penyimpanan data yang bersifat terkomputerisasi. Simbol dari *database* adalah sebagai berikut:



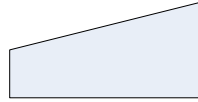
Gambar 3.5. *Simbol Database*

*Decision* merupakan operator logika yang digunakan sebagai penentu keputusan dari suatu permintaan atau proses dengan dua nilai, benar dan salah. Operator logika ini digambarkan sebagai berikut:



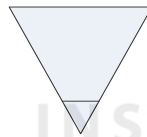
Gambar 3.6. *Simbol Decision*

*Manual input* digunakan untuk melakukan proses *input* ke dalam *database* melalui *keyboard*. *Manual input* digambarkan dengan simbol:



Gambar 3.7. *Simbol Manual Input*

*Off-line storage* merupakan bentuk media penyimpanan yang berbeda dengan *database*, dimana media penyimpanan ini menyimpan dokumen secara manual atau lebih dikenal dengan nama arsip. *Off-line storage* digambarkan dengan simbol:



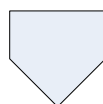
Gambar 3.8. *Simbol Off-Line Storage*

*On-page reference* digunakan sebagai simbol untuk menghubungkan bagan desain sebuah sistem apabila hubungan arus data yang ada terlalu jauh dalam permasalahan letaknya. Bentuk simbol *On-page reference* adalah sebagai berikut:



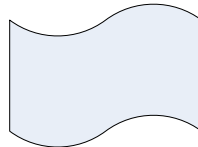
Gambar 3.9. *Simbol On-Page Reference*

*Off-page reference* memiliki sifat yang sedikit berbeda dengan *On-page reference*, karena simbol ini hanya digunakan apabila arus data yang ada dilanjutkan ke halaman yang berbeda. Bentuk simbolnya adalah:



Gambar 3.10. *Simbol Off-Page Reference*

*Paper tape* merupakan sebuah simbol yang umumnya menggantikan bentuk penggambaran jenis pembayaran yang digunakan (misal: uang) dalam transaksi yang ada pada sistem yang dirancang. Bentuk dari *paper tape* adalah dengan simbol:



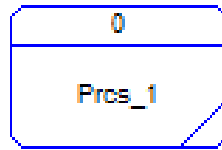
Gambar 3.11. Simbol Paper Tape

### 3.2.2 Data Flow Diagram

Menurut (Andri Kristanto, 2004), *Data Flow Diagram* (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data tersebut disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan, dan proses yang dikenakan pada data tersebut.

*Data Flow Diagram* merupakan suatu metode pengembangan sistem yang terstruktur (*structured analysis and design*). Penggunaan notasi dalam *data flow diagram* sangat membantu untuk memahami suatu sistem pada semua tingkat kompleksitas. Pada tahap analisis, penggunaan notasi ini dapat membantu dalam berkomunikasi dengan pemakai sistem untuk memahami sistem secara logika.

Di dalam *data flow diagram* terdapat empat simbol yang digunakan yaitu *process*, *external entity*, *data store*, dan *data flow*. Simbol *process* digunakan untuk melakukan suatu perubahan berdasarkan data yang diinputkan dan menghasilkan data dari perubahan tersebut. Simbol *process* dapat digambarkan sebagai bentuk berikut:



Gambar 3.12. *Simbol Process*

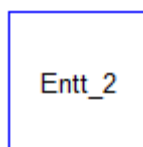
Pada bentuk gambar *process*, bagian atas berisi nomor untuk identitas proses. Suatu proses dengan nomor 0 (nol atau kosong) menandakan bahwa proses tersebut adalah sebuah *context diagram*. Diagram ini merupakan level tertinggi dari DFD yang menggambarkan hubungan sistem dengan lingkungan luarnya. Pembuatan *context diagram* dapat dilakukan dengan terlebih dahulu menentukan nama sistemnya, menentukan batasan dari sistem, dan menentukan *terminator* yang diterima atau diberikan daripada sistem untuk kemudian dilakukan penggambaran.

Nomor 1, 2, 3, dan seterusnya menandakan bahwa proses tersebut diartikan sebagai proses level-0 (nol) yang merupakan hasil turunan atau *decompose* dari proses *context diagram*. Proses level-0 membahas sistem secara lebih mendetil, baik dipandang dari segi kegiatan dari sebuah bagian, alur data yang ada, maupun *database* yang digunakan didalamnya. Pembuatannya dapat dilakukan dengan cara menentukan proses utama yang ada dalam sistem, menentukan alur data yang diterima dan diberikan masing-masing proses daripada sistem sambil memperhatikan konsep keseimbangan (alur data yang masuk atau keluar dari suatu level harus sama dengan alur data yang masuk dan keluar pada level berikutnya), memunculkan *data store* sebagai sumber maupun tujuan data (*optional*), menggambarkan diagram level-0, menghindari perpotongan arus data,

dan melakukan pemberian nomor pada proses utama (nomor tidak menunjukkan urutan proses).

Nomor 1.1, 1.2, 2.1, 2.2, dan seterusnya merupakan sebuah proses turunan atau *decompose* dari proses level-0 yang disebut sebagai proses level-1 (satu). Proses level-1 menggambarkan detail kerja dari sebuah bagian dalam sebuah sistem. Penggambarannya dilakukan dengan cara menentukan proses yang lebih kecil (sub-proses) dari proses utama yang ada di level-0, menentukan apa yang diterima atau diberikan masing-masing sub-proses daripada sistem dan tetap memperhatikan konsep keseimbangan, memunculkan *data store* sebagai sumber maupun tujuan alur data (*optional*), menggambar DFD level-1, dan berusaha untuk menghindari perpotongan arus data. Hasil turunan akhir disebut sebagai *the lowest level*, dimana hasil akhir ini tergantung dari kompleksitas sistem yang ada.

*External entity* disimbolkan dengan bentuk persegi yang digunakan untuk menggambarkan pelaku-pelaku sistem yang terkait, dapat berupa orang-orang, organisasi maupun instansi. *External entity* dapat memberikan masukan kepada *process* dan mendapatkan keluaran dari *process*. *External entity* digambarkan dalam bentuk sebagai berikut:

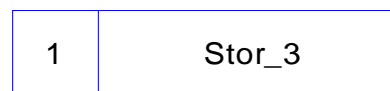


Gambar 3.13. *Simbol External Entity*

*Data store* digunakan sebagai media penyimpanan suatu data yang dapat berupa *file* atau *database*, arsip atau catatan manual, lemari *file*, dan tabel-tabel dalam *database*. Penamaan *data store* harus sesuai dengan bentuk data yang

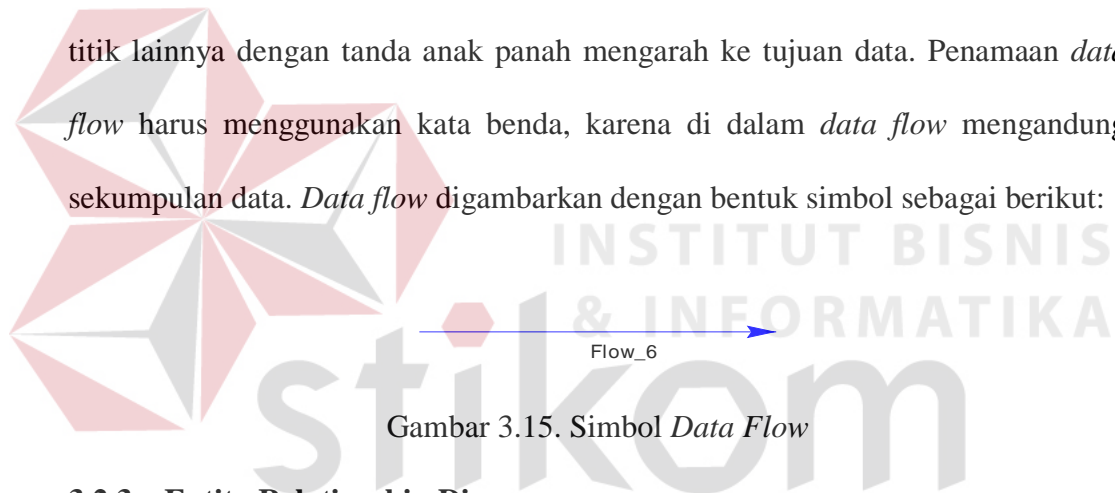


tersimpan pada *data store* tersebut, misalnya tabel pelamar, tabel pendidikan, tabel lulus seleksi, dan lain-lain. *Data store* digambarkan dalam bentuk simbol sebagai berikut:



Gambar 3.14. Simbol *Data Store*

*Data flow* merupakan penghubung antara *external entity* dengan *process* dan *process* dengan *data store*. *Data flow* menunjukkan aliran data dari satu titik ke titik lainnya dengan tanda anak panah mengarah ke tujuan data. Penamaan *data flow* harus menggunakan kata benda, karena di dalam *data flow* mengandung sekumpulan data. *Data flow* digambarkan dengan bentuk simbol sebagai berikut:



Gambar 3.15. Simbol *Data Flow*

### 3.2.3 Entity Relationship Diagram

*Entity Relationship Diagram* (ERD) adalah gambaran pada sistem dimana di dalamnya terdapat hubungan antara *entity* beserta relasinya. *Entity* merupakan sesuatu yang ada dan terdefinisikan di dalam suatu organisasi, dapat abstrak dan nyata. Untuk setiap *entity* biasanya mempunyai *attribute* yang merupakan ciri *entity* tersebut. *Attribute* adalah kolom di sebuah relasi. Macam-macam *attribute* 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 guru mengajar satu siswa dan satu siswa hanya diajar oleh satu guru.



Gambar 3.16 Relasi *One to One*

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

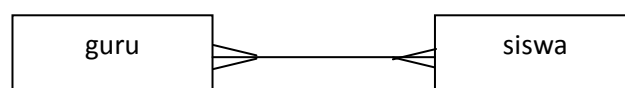
Relasi antara *entity* yang pertama dengan *entity* yang kedua adalah satu berbanding banyak atau dapat pula dibalik, banyak berbanding satu. Contoh: Pada sekolah, satu guru mengajar banyak siswa dan banyak siswa diajar oleh satu guru.



Gambar 3.17 Relasi *One to Many*

### 3. *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 3.18 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.

### 3.3 Website

*Website* merupakan kumpulan halaman web yang saling terhubung dan file - filenya saling terkait. *Web* terdiri dari *page* atau halaman, dan kumpulan halaman yang dinamakan *homepage*. *Homepage* berada pada posisi teratas, dengan halaman - halaman terkait berada di bawahnya. Biasanya setiap halaman di bawah *homepage* disebut *child page*, yang berisi *hyperlink* ke halaman lain dalam *web*. (Gregorius, 2000). *Website* awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep *hyperlink*, yang memudahkan *surfer* atau pengguna internet melakukan penelusuran informasi di internet. Informasi yang disajikan dengan *web* menggunakan konsep multimedia, informasi dapat disajikan dengan menggunakan banyak media, seperti teks, gambar, animasi, suara, atau film.

### 3.3.1 World Wide Web

WWW (*World Wide Web*) adalah sebuah metode baru yang berjalan di dunia internet yang akhir - akhir ini berkembang dengan cepat (Bunafit Nugroho, 2004).

### 3.4 Database

*Database* merupakan suatu kumpulan dari data – data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras (*hardware*) komputer dan digunakan perangkat lunak (*software*) untuk memanipulasinya. Data disimpan didalam *database* untuk keperluan penyediaan informasi, diorganisasikan untuk efisiensi kapasitas penyimpanan supaya informasi yang dihasilkan berkualitas. *Database* diakses atau dimanipulasikan dengan menggunakan paket software *Database Mangement System* (DBMS). (Jogiyanto, 1993).

### 3.5 Aplikasi

Aplikasi dapat diartikan sebagai program komputer yang di buat unuk menolong manusia dalam melaksanakan tugas tertentu. ( Kamus Komputer, 1996). Aplikasi *software* yang di rancang untuk penggunaan praktisi khusus klasifikasi luas ini dapat di bagi menjadi dua. (Santoso, 2000).

1. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas.
2. Aplikasi paket, suatu program dengan dokumentasi tergabung yang digunakan untuk sejenis masalah tertentu, misalnya penggajian (*payroll*).

### 3.6 PHP

PHP adalah kependekan dari *Hypertext Preprocessor*, bahasa interpreter yang mempunyai kemiripan dengan bahasa C dan Perl yang mempunyai kesederhanaan dalam perintah, yang digunakan untuk pembuatan aplikasi web. (Sidik, 2004)

PHP/FI merupakan nama awal dari PHP (*Personal Home Page/Form Interface*). Dibuat pertama kali oleh Rasmus Lerdoff. PHP awalnya merupakan program CGI yang dikhususkan untuk menerima *input* melalui form yang ditampilkan dalam *browser* web. Dengan menggunakan PHP maka *maintenance* suatu situs web menjadi lebih mudah. Proses *update* dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan script PHP.

PHP merupakan *script* untuk pemrograman *script* web *server-side*, *script* yang membuat dokumen HTML secara *on the fly*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML.

### 3.7 Web Server

*Web Server* adalah perangkat keras dan perangkat lunak yang dipakai untuk menyimpan dan mengirim dokumen HTML (*Hypertext Markup Language*) untuk digunakan dalam *World Wide Web* (WWW) (Dhanta, 2009).

### 3.8 MySQL

Menurut (Nugroho, 2004). MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL (*Structured Query Language*). MySQL

juga dapat berperan sebagai *client/server*, yang *open source* dengan kemampuan dapat berjalan baik di OS (*Operating System*) manapun. Selain itu *database* ini memiliki kelebihan dibanding database lain, diantaranya adalah:

1. MySQL sebagai *Database Management System (DBS)*
2. MySQL sebagai *Relation Database Management System (RDBMS)*
3. MySQL adalah sebuah *software database* yang bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.
4. MySQL merupakan *database server*, jadi dengan menggunakan *database* ini, dapat dihubungkan ke media internet sehingga dapat diakses dari jauh.
5. Selain menjadi server yang melayani permintaan, MySQL juga dapat melakukan *query* yang mengakses *database* pada server.
6. Mampu menerima *query* yang bertumpuk dalam satu permintaan atau yang disebut *Multi-Threading*.
7. Mampu menyimpan data yang berkapasitas besar hingga berukuran *gigabyte* sekalipun.
8. Memiliki kecepatan dalam pembuatan tabel maupun *update* tabel.
9. Menggunakan bahasa permintaan standar yang bernama SQL (*Structure Query Language*) yaitu sebuah bahasa permintaan yang distandarkan pada beberapa *database server* seperti *oracle*.

Dengan beberapa kelebihan yang dimiliki, MySQL menjadi sebuah program *database* yang sangat terkenal digunakan. Pada umumnya MySQL digunakan sebagai *database* yang diakses melalui web.

### 3.9 Browser

*Browser* merupakan program aplikasi yang digunakan untuk *browsing*. Sebuah program yang memungkinkan pengguna internet mengakses dan membaca dokumen yang ditulis dalam *hypertext* pada *world wide web* (WWW) yang terkoneksi dengan internet. *Browser* yang paling populer saat ini adalah *Internet Explorer*, *Opera*, *Mozilla* dan *Netscape* (Dhanta, 2009). Sedangkan menurut (Chendramata, 2008), *web browser* adalah sebuah *server web* yang berfungsi untuk menerima permintaan HTTP dari sebuah klien. Jadi dapat disimpulkan bahwa *web browser* adalah aplikasi yang digunakan sebagai media untuk menampilkan permintaan HTTP dari klien.

