

## **BAB III**

### **LANDASAN TEORI**

#### **3.1 Rancang Bangun**

##### **3.1.1 Pengertian Rancang**

Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan (Pressman, 2002).

##### **3.1.2 Pengertian Bangun**

Bangun atau pembangunan sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian (Pressman, 2002). Bangun sistem adalah membangun sistem informasi dan komponen yang didasarkan pada spesifikasi desain (Whitten et al, 2004).

#### **3.2 Perancangan Sistem**

Pengertian perancangan sistem menurut Jogiyanto (2005), yaitu:

*“Perancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan. Tahap ini menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancangan bangun yang telah ditetapkan pada akhir tahap analisa sistem”.*

#### **3.3 Analisis Sistem**

Menurut Jogiyanto (2005) analisis sistem adalah penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan,

hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan.

Menurut Kristanto (2003) analisis sistem adalah suatu proses mengumpulkan dan menginterpretasikan kenyataan-kenyataan yang ada, mendiagnosa persoalan dan menggunakan keduanya untuk memperbaiki sistem.

Menurut Jogiyanto (2005) analisis sistem (analisis informasi) adalah orang yang menganalisis sistem (mempelajari masalah-masalahan yang timbul dan menentukan kebutuhan pemakai sistem) untuk mengidentifikasi pemecahan permasalahan tersebut.

Menurut Kristanto (2003) analisis sistem adalah orang yang mempunyai kemampuan untuk menganalisis sebuah sistem, memilih alternatif pemecahan masalah dan menyelesaikan masalah tersebut dengan menggunakan komputer.

### **3.4 Perancangan sistem**

Perancangan sistem adalah merancang atau mendesain suatu sistem yang baik, yang isinya adalah langkah-langkah operasi dalam proses pengolahan data dan prosedur untuk mendukung operasi sistem.

Menurut Jogiyanto (2005), dalam bukunya Analisis Dan Disain Sistem, Perancangan sistem dapat diartikan sebagai berikut :

1. Tahap setelah analisis dari siklus pengembangan sistem.
2. Pendefinisian dari kebutuhan-kebutuhan fungsional.
3. Persiapan untuk rancang bangun implementasi.
4. Menggambarkan bagaimana suatu sistem dibentuk.


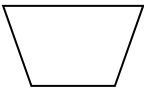
5. Yang dapat berupa penggambaran perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.
6. Termasuk menyangkut mengkonfigurasi dari komponen perangkat keras dari suatu sistem.


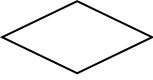

Adapun langkah-langkah dalam perancangan sistem meliputi *Document Flow*, *System Flow*, *Data Flow Diagram (DFD)*, *Conceptual Data Model (CDM)*, *Physical Data Model*, dan *Database Management System (DBMS)*.

### 3.4.1 Document Flow

Bagan alir dokumen (*document flow*) atau disebut bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya (Jogiyanto, 2005). Bagan alir dokumen digambar dengan symbol-simbol yang tampak sebagai berikut :

Tabel 3.1 Simbol *Document Flow*


No	Simbol	Nama Simbol	Keterangan
1.		Terminator	Digunakan pada awal pembuatan <i>document flow</i> sebagai mengawali ( <i>Start</i> ) dan mengakhiri ( <i>End</i> ) <i>flow</i> .
2.		Proses manual	Dinyatakan sebagai proses manual karena dalam notasi <i>document flow</i> segala bentuk proses masih belum dilakukan

No	Simbol	Nama Simbol	Keterangan
			oleh komputer.
3.		Dokumen	Notasi dokumen ini umumnya digambarkan sebagai bentuk lain dari arsip, laporan atau dokumen lainnya yang berbentuk kertas.
4.		<i>Decision</i> (keputusan)	Dalam penggambaran notasi <i>decision</i> ini selalu menghasilkan dengan keputusan “ya” atau “tidak”.
5.		<i>Connector</i>	Difungsikan sebagai penunjuk arah aliran dari satu proses ke proses yang lainnya yang saling berkaitan.

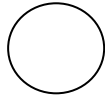
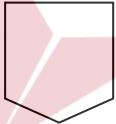
### 3.4.2 System Flow

Bagan alir sistem (*system flow*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan menjelaskan urutan-urutan dari prosedur-prosedur yang ada dalam sistem (Jogiyanto, 2005). Bagan alir sistem menunjukkan apa yang dikerjakan sistem. Bagan alir sistem ini menggunakan symbol-simbol yang dapat dilihat pada tabel 3.2.

Tabel 3.2 Simbol *System Flow*

No.	Simbol	Nama Simbol	Keterangan
1.		Proses Komputerisasi	Menggambarkan sistem yang dikerjakan oleh komputer.

No.	Simbol	Nama Simbol	Keterangan
2.		<i>Database</i>	Gambar disamping adalah representasi dari <i>database</i> yang mana fungsinya untuk menyimpan data dari proses sebelumnya.
3.		<i>Sub-process</i>	Simbol <i>Sub-proscess</i> difungsikan untuk menunjukkan adanya proses yang lebih rinci dari suatu proses utama.
4.		Dokumen	Simbol dokumen menunjukkan tentang dokumen yang dihasilkan.
5.		<i>Decision</i> (keputusan)	Dalam penggambaran notasi <i>decision</i> ini selalu menghasilkan dengan keputusan “ya” atau “tidak”.
6.		<i>Connector</i>	Difungsikan sebagai penunjuk arah aliran dari satu proses ke proses yang lainnya yang saling berkaitan
7.		<i>Input/Output</i>	Simbol disamping difungsikan untuk menunjukkan masukan data ( <i>input</i> ) dan data yang

No.	Simbol	Nama Simbol	Keterangan
			dihasilkan ( <i>output</i> ).
8.		<i>Connector (On Page Reference)</i>	Untuk menunjukkan hubungan simbol yang saling terkait dalam <i>system flow</i> . Selain itu juga sebagai pengganti garis <i>connector</i> untuk menghubungkan simbol yang saling berjauhan.
9.		<i>Connector (Off-Page Reference)</i>	Untuk menghubungkan simbol yang berada pada halaman yang berbeda.

### 3.4.3 Data Flow Diagram (DFD)

Pengertian Data Flow Diagram (DFD) menurut Kristanto (2003) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluaran dari sistem, dimana data di simpan, proses apa yang menghasilkan data tersebut, dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. DFD atau diagram arus digunakan untuk sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika.

Jenis DFD meliputi *Context Diagram* dan *Diagram Level n / Data Flow Diagram Levelled*.

#### A. *Context Diagram*

Menurut Jogiyanto (2005) pengertian diagram konteks adalah :

“Sebuah diagram sederhana yang menggambarkan hubungan antara entity luar, masukan dan keluaran dari sistem. Diagram konteks dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem.”

*Context diagram* menggambarkan sistem dalam satu lingkaran dan hubungan dengan entitas luar. Lingkaran tersebut menggambarkan keseluruhan proses dalam sistem.

Beberapa hal yang harus diperhatikan dalam menggambar *context diagram* :

- a. Komponen-komponen yang berupa fungsi orang, aktivitas, kejadian dan sebagainya yang saling berhubungan dan melengkapi satu sama lain untuk mencapai satu atau lebih tujuan yang telah ditetapkan.
- b. Menggunakan satu simbol proses.
- c. Nama/keterangan di simbol proses tersebut sesuai dengan fungsi sistem tersebut.
- d. Antara Entitas Eksternal/Terminator tidak diperbolehkan komunikasi langsung.
- e. Jika Terminator mewakili individu (personil) sebaiknya diwakili oleh peran yang dipermainkan personil tersebut.
- f. Aliran data ke proses dan keluar sebagai output keterangan aliran data berbeda.

#### **B. Diagram Level $n$ / Data Flow Diagram Levelled**

Dalam Diagram Level  $n$  DFD dapat digunakan untuk menggambarkan diagram fisik maupun diagram-diagram logis. Dimana Diagram Level  $n$  merupakan hasil pengembangan dari *context diagram* ke dalam komponen yang lebih detail tersebut disebut dengan *top-down partitioning*. Jika

kita melakukan pengembangan dengan benar, kita akan mendapatkan DFD-DFD yang seimbang.

Simbol-simbol DFD meliputi terminator (*External Entity*), arus data (*data flow*), proses (*process*) dan simpanan data (*data store*).

### B.1 Terminator/Kesatuan Luar (*External Entity*)

Setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan input atau menerima output dari sistem (Jogiyanto, 2005). Suatu kesatuan luar dapat disimbolkan dengan suatu notasi kotak.



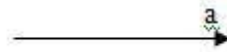
Gambar 3.1 Notasi terminator/Kesatuan Luar di DFD

Terminator dapat berupa orang, sekelompok orang, organisasi, departemen di dalam organisasi, atau perusahaan yang sama tetapi di luar kendali sistem yang sedang dibuat modelnya. Terminator dapat juga berupa departemen, divisi atau sistem di luar sistem yang berkomunikasi dengan sistem yang sedang dikembangkan.

### B.2 Arus Data (*Data Flow*)

Arus data (*data flow*) di DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (*Process*), simpanan data (*data store*) dan kesatuan luar (*external entity*). Arus data ini menunjukkan arus data yang dapat berupa masukkan untuk sistem atau hasil dari proses sistem.





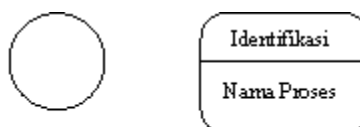
Gambar 3.2 Notasi Arus Data di DFD

Arus data data dapat dapat berbentuk berbentuk sebagai sebagai berikut :

1. Formulir atau atau dokumen dokumen yang yang digunakan digunakan perusahaan perusahaan
2. Laporan tercetak tercetak yang yang dihasilkan dihasilkan sistem sistem
3. Output dilayar komputer
4. Masukan untuk komputer komputer
5. Komunikasi ucapan
6. Surat atau memo
7. Data yang dibaca atau atau direkam di file
8. Suatu isian yang yang dicatat pada buku agenda
9. Transmisi data dari suatu komputer ke komputer lain.

### B.3 Proses (*Process*)

Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin, atau komputer dan hasil suatu arus data yang masuk ke dalam proses untuk dilakukan arus data yang akan keluar dari proses. Suatu proses dapat ditunjukkan dengan simbol lingkaran atau dengan simbol empat persegi panjang tegak dengan sudut-sudutnya tumpul.



Gambar 3.3 Notasi Proses di DFD

Ada beberapa hal yang perlu diperhatikan tentang proses :

1. Proses harus memiliki input dan output.
2. Proses dapat dihubungkan dengan komponen terminator, data store atau proses melalui alur data.
3. Sistem/bagian/divisi/departemen yang sedang dianalisis oleh profesional sistem digambarkan dengan komponen proses.

#### B.4 Simpanan Data (*Data Store*)

Simpanan data (*data store*) merupakan simpanan dari data yang dapat berupa file atau database di sistem komputer, arsip atau catatan manual, kotak tempat data di meja seseorang, tabel acuan manual, agenda atau buku. Simpanan data di DFD dapat disimbolkan dengan sepasang garis horizontal paralel yang tertutup di salah satu ujungnya.



Gambar 3.4 Simbol dari Simpanan Data di DFD

#### 3.4.4 *Conceptual Data Model (CDM)*

*Conceptual Data Model* adalah proses membangun suatu model berdasarkan informasi yang digunakan oleh perusahaan atau organisasi, tanpa pertimbangan perencanaan fisik (Connolly, 2002).

##### A. *Entity types*

Menurut Connolly (2002), *entity types* adalah kumpulan objek yang mempunyai karakteristik yang sama, dimana telah diidentifikasi oleh perusahaan. Menurut Silberschatz (2002), *entity types* adalah kumpulan dari *entity* yang memiliki tipe dan karakteristik yang sama.

*Entity* dapat dibedakan menjadi dua yaitu :

1. *Strong Entity* : *entity* yang keberadaannya tidak tergantung kepada *entity* lain (Fathansyah, 1999).
2. *Weak entity* : *entity* yang keberadaannya tergantung dari *entity* lain (Fathansyah, 1999).

Contohnya adalah *entity* mahasiswa dan orang tua. Dimana mahasiswa merupakan *strong entity* dan orang tua merupakan *weak entity* karena keberadaan *entity* orang tua tergantung dari *entity* mahasiswa.

#### B. *Relationship types*

Menurut Connolly (2002) definisi dari *relationship types* adalah kumpulan antar *entity* yang saling berhubungan dan mempunyai arti.

#### C. *Attribute dan attribute domains*

*Attribute* adalah karakteristik dari suatu *entity* atau relasi (Connolly, 2002). Setiap *attribute* diperbolehkan untuk memiliki nilai yang disebut dengan domain. *Attribute domains* adalah kumpulan dari nilai-nilai yang diperbolehkan untuk satu atau lebih *attribute*.

#### 3.4.5 *Physical Data Model (PDM)*

*Physical Data Model* adalah suatu proses untuk menghasilkan gambaran dari implementasi basis data pada tempat penyimpanan, menjelaskan dasar dari relasi, organisasi file dan indeks yang digunakan untuk efisiensi data dan menghubungkan beberapa *integrity constraints* dan tindakan keamanan (Connolly, 2002).

### 3.4.6 Database Management System (DBMS)

Menurut Connoly (2002), Database Management System adalah sistem software yang dapat mendefinisikan, membuat, memelihara dan mengontrol akses ke basis data.

Fasilitas yang disediakan oleh DBMS adalah:

- A. Dapat mendefinisikan basis data dengan menggunakan *Data Definition Language* (DDL). DDL dapat memberi fasilitas kepada pengguna untuk menspesifikasikan tipe data, struktur dan batasan aturan mengenai data yang bisa disimpan ke dalam basis data.
- B. Pengguna dapat menambah, mengedit, menghapus dan mendapatkan kembali data dengan menggunakan data manipulation language.
- C. Dapat mengontrol akses ke basis data, yaitu mencegah pengguna tanpa otoritas, sistem integrasi untuk memelihara konsistensi penyimpanan data, sistem control untuk memperbolehkan pengguna untuk akses, sistem kontrol untuk pengembalian data yang bisa mengembalikan data ke keadaan semula apabila ada kegagalan software atau hardware, catalog yang dapat diakses pengguna yang mendeskripsikan data dalam basis data.