

## BAB II

### LANDASAN TEORI

#### A. Perancangan Terkait

Intan Septavia (2015), “Sistem Informasi Penyewaan Mobil Berbasis Web Di Jasa Karunia Tour And Travel”, dengan adanya aplikasi ini produktivitas kerja perudahaan dapat berkembang lebih baik lagi. Aplikasi ini dapat memberi informasi secara detail dan akurat mengenai mobil-mobil yang disewakan oleh jasa karunia tour & travel.

Joko Tri Cahyono (2014), “Pembuatan Sistem Informasi Rental Mobil Purnama Rent Car Ploso Pacitan Berbasis Web”, Dari hasil pembuatan sistem informasi rental mobil purnama rent car Ploso Pacitan berbasis web pihak menjadi mudah dalam menginformasikan mobil Purnama Rent Car yang siap disewa melalui sistem pemesanan. Dengan adanya website masyarakat bisa mengakses dan mendapatkan informasi serta memesan mobil Purnama Rent Car dengan cepat dari rumahnya, dan dapat memilih mobil yang diinginkan untuk disewa.

Muh. Zufri Halim (2013), “Rancang Bangun Sistem Informasi Penyewaan Pada Rental Mobil Berbasis Web dan Menggunakan SMS Gateway”, dengan hasil dari penelitian berupa aplikasi sistem informasi penyewaan rental mobil berbasis web dan menggunakan sms gateway. Sistem informasi penyewaan rental mobil ini berisi aplikasi pengolahan data mobil, pengolahan data penyewa, dan disertai laporan penyewaan. Adanya sistem informasi penyewaan mobil ini bertujuan untuk memudahkan pemilik rental

mobil dalam pengecekan laporan penyewaan data sewa dan membantu dalam penyimpanan dan pengolahan data penyewa.

## **B. Pengertian Sistem Informasi**

### **1. Pengertian Sistem**

Menurut Davis (1985), sistem adalah bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud. Lucas (1989), mendefinisikan sistem sebagai suatu komponen atau variabel yang terorganisasi, saling berinteraksi, saling bergantung, satu sama lain dan terpadu. McLeod berpendapat, sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan.

Dari beberapa pendapat tentang definisi sistem tersebut, definisi sistem yang lebih luas dibandingkan pendekatan sistem yang lebih menekankan pada prosedur dan lebih banyak diterima adalah pendekatan sistem yang merupakan kumpulan dari komponen atau elemen-elemen atau subsistem-subsistem, karena pada kenyataannya suatu sistem memang terdiri dari subsistem-subsistem.

### **2. Pengertian Informasi**

Gordon.B.Davis (1985), mendefinisikan Informasi sebagai data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun yang akan datang. Informasi mempunyai ciri benar atau salah, baru, tambahan, dan korektif. Raymond McLeod (1995), mendefinisikan informasi sebagai

data yang diolah menjadi bentuk yang lebih berarti bagi penerimanya.

Sumber informasi adalah data.

Kegunaan informasi adalah untuk mengurangi ketidak pastian didalam proses pengambilan keputusan tentang suatu keadaan. Suatu informasi dikatan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkan informasi tersebut.

### 3. Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Tujuan dari sistem informasi sendiri adalah menghasilkan informasi (Jogiyanto, 2001). Manfaat:

- a. Organisasi menggunakan sistem informasi untuk mengolah transaksi-transaksi, mengurangi biaya dan menghasilkan pendapatan sebagai salah satu produk atau pelayanan mereka.
- b. Bank menggunakan sistem informasi untuk mengolah cek-cek nasabah dan membuat berbagai rekening koran dan transaksi yang terjadi.
- c. Perusahaan menggunakan sistem informasi untuk mempertahankan persediaan pada tingkat paling rendah agar konsisten dengan jenis barang yang tersedia.

### C. *Web*

*Web* adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang

menggunakan protokol HTTP (*hypertext transfer protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut *browser*.

*Browser* (perambah) adalah aplikasi yang mampu menjalankan dokumen-dokumen *web* dengan cara diterjemahkan. Prosesnya dilakukan oleh komponen yang terdapat di dalam aplikasi *browser* yang biasa disebut *web engine*. Semua dokumen *web* ditampilkan oleh *browser* dengan cara diterjemahkan. Situs *Web* adalah dokumen-dokumen *web* yang terkumpul menjadi satu kesatuan yang memiliki *Unified Resource Locator* (URL atau *domain* dan biasanya di-*publish* di *internet* atau *intranet*. (Arief, 2011:17).

#### **D. PHP (*Hypertext Preprocessor*)**

##### **1. Pengertian *PHP* (*Hypertext Preprocessor*)**

*PHP* (*Hypertext Preprocessor*) adalah bahasa *server side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis (Arief, 2011:43). Karena *PHP* merupakan *server side scripting* maka sintaks dan perintah-perintah *PHP* yang dieksekusi di *server* kemudian hasilnya dikirimkan ke *browser* dalam format *HTML*. Dengan demikian kode program yang ditulis dalam *PHP* tidak akan terlihat oleh *user* sehingga keamanan halaman *web* lebih terjamin. *PHP* dirancang untuk membentuk halaman *web* yang dinamis, yaitu halaman *web* yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman *web*.

*PHP* termasuk dalam *open source product*, sehingga *source code* *PHP* dapat diubah dan didistribusikan secara bebas. Versi terbaru *PHP*

dapat diunduh secara gratis di situs resmi *PHP*: <http://www.PHP.net>. *PHP* juga dapat berjalan pada berbagai *web* server seperti *IIS* (*Internet Information Server*), *Apache*, dan *Xitami*. *PHP* juga mampu lintas *platform*. Artinya *PHP* dapat dibangun sebagai modul pada *web* server *Apache* dan sebagai *binary* yang dapat berjalan sebagai *CGI* (*Common Gateway Interface*). (Arief, 2011:24). Keunggulan *PHP* adalah :

- a. *PHP* memiliki tingkat akses yang lebih cepat
- b. *PHP* mampu berjalan di beberapa *server web*, seperti *Apache*, *Microsoft IIS*, dan *Xitami*
- c. *PHP* mendukung akses ke beberapa *database* baik yang bersifat gratis atau komersial, seperti *mSQL*, *MySQL*, dan *MicrosoftSQL Server*
- d. *PHP* bersifat *Open Source*

## 2. *Syntaks PHP*

*Syntaks PHP* ditulis dalam apitan tanda khusus *PHP*. Ada empat macam pasangan tag *PHP* yang digunakan :

- a. `<?PHP.....?>`
- b. `<script language="PHP">...</script>`
- c. `<?.....?>`
- d. `<%.....%>`

*Point a* dan *b* merupakan cara yang paling umum yang digunakan sekalipun cara *c* lebih praktis karena cara *c* tidak selalu diaktifkan pada konfigurasi *file PHP* yang terdapat pada direktori `c:\apache\php`. Cara *d* juga memungkinkan kemudahan bagi anda yang sudah terbiasa dengan

*ASP (Active Server Pages)*. Namun, bila itu dikenal harus dilakukan pengaktifan file konfigurasi *PHP*.

## E. *MySQL*

*MySQL* merupakan *software* yang tergolong sebagai *DBMS (Database Managemen Sistem)* yang bersifat *Open Source*. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat *MySQL*), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa dengan cara *men-download* (mengunduh) di internet secara gratis. (Kadir; 2008)

*MySQL* awalnya dibuat oleh perusahaan konsultan bernama *TcX* yang berlokasi di Swedia. Saat ini pengembangan *MySQL* berada dibawah naungan perusahaan *MySQL AB*.

Sebagai *software DBMS*, *MySQL* memiliki sejumlah fitur seperti yang dijelaskan dibawah ini.

a. *Multiplatform*.

*MySQL* tersedia pada beberapa *platform* (Windows, Linux, Unix, dan lain-lain).

b. *Handal, cepat, dan mudah digunakan*.

*MySQL* tergolong sebagai *database server* (server yang melayani permintaan terhadap database) yang handal, dapat menangani database yang besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses database, dan sekaligus mudah untuk digunakan. Berbagai *tool* pendukung juga tersedia (walaupun dibuat oleh pihak lain).

Perlu diketahui, *MySQL* dapat menangani sebuah table yang berukuran dalam *terabyte* (1 *terabyte* = 1024 *gigabyte*). Namun, ukuran yang sesungguhnya sangat bergantung pada batasan sistem operasi. Sebagai contoh, pada sistem *solaris 9/10*, batasan ukuran file sebesar 16 *terabyte*.

c. Jaminan keamanan akses.

*MySQL* mendukung pengamanan *database* dengan berbagai kriteria pengaksesan. Sebagai gambaran, dimungkinkan untuk mengatur *user* tertentu agar bisa mengakses data yang bersifat rahasia (misalnya gaji pegawai), sedangkan *user* lain tidak boleh. *MySQL* juga mendukung konektivitas ke berbagai *software*. Sebagai contoh, dengan menggunakan *ODBC (Open Database Connectivity)*, *database* yang ditangani *MySQL* dapat diakses melalui program yang dibuat dengan *Visual Basic*. *MySQL* juga mendukung program klien melalui *JDBC (Java Database Connectivity)*. *MySQL* juga bisa diakses melalui aplikasi berbasis *Web* : misalnya dengan menggunakan *PHP*.

d. Dukungan *SQL*.

Seperti tersirat dalam namanya, *MySQL* mendukung perintah *SQL (Structured Query Language)*. Sebagai mana diketahui, *SQL* merupakan standart dalam pengaksesan *database relasional*. Pengetahuan akan *SQL* akan memudahkan siapa pun untuk menggunakan *MySQL*. (Kadir; 2008; 2-3).

## F. Basis Data

### 1. Pengertian Basis Data

Basis dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data merupakan representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Basis data (*database*) merupakan kumpulan data yang saling berhubungan atau mempunyai relasi. (Yakub, 2008). Menurut Janer (2007:2), Basis Data adalah koleksi data yang bisa mencari secara menyeluruh dan secara sistematis memelihara dan *me-retrieve* informasi.

### 2. Manfaat Basis Data

- a. Kecepatan dan kemudahan (*Speed*), pemanfaatan basis data memungkinkan untuk dapat, menyimpan, merubah, dan menampilkan kembali data tersebut dengan lebih cepat dan mudah.
- b. Efisiensi ruang penyimpanan (*space*), dengan basis data efisiensi atau optimalisasi penggunaan ruang penyimpanan dapat dilakukan, karena penekanan jumlah redundansi data, baik dengan sejumlah pengkodean atau dengan membuat tabel-tabel yang saling berhubungan.
- c. Keakuratan (*accuracy*), pembentukan relasi antar data bersama dengan penerapan aturan atau batasan (*constraint*) tipe, domain dan keunikan data dapat diterapkan dalam sebuah basis data.



- d. Ketersediaan (*availability*), dapat memilah data utama atau master, transaksi, data histori hingga data kadaluwarsa. Data yang jarang atau tidak digunakan lagi dapat diatur dari sistem basis data yang aktif.
- e. Keamanan (*security*), untuk menentukan siapa-siapa yang berhak menggunakan basis data beserta objek-objek di dalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukan.

Kebersamaan pemakai (*sharebility*), basis data dapat digunakan oleh beberapa pemakai dan beberapa lokasi. Basis Data yang dikelola oleh sistem (aplikasi) yang mendukung *multiuser* dapat memenuhi kebutuhan, akan tetapi harus menghindari inkonsistensi data. (Yakub, 2008:34)

### 3. Operasi Basis Data

Pada sebuah *disk (hard disk)*, basis dapat diciptakan dapat pula ditiadakan. Pada sebuah *disk* juga dapat menempatkan beberapa basis data, misalnya basis data kepegawaian, akademik, penjualan, perpustakaan dan lain-lain. Sementara dalam sebuah basis data dapat ditempatkan pada satu file atau tabel barang, faktur, pelanggan dan transaksi barang. Operasi-operasi dasar yang dapat dilakukan basis data adalah:

- a. Pembuatan basis data baru (*CREATE DATABASE*)
- b. Penghapusan basis data (*DROP DATABASE*)
- c. Pembuatan *file* atau tabel baru ke suatu basis data (*CREATE TABLE*)
- d. Penghapusan *file* atau tabel dari suatu basis data (*DROP TABLE*)

- e. Penambahan atau pengisian data baru di sebuah basis data (*INSERT*)
- f. Pengambilan data dari sebuah *file* atau tabel (*RETRIEVE* atau *SEARCH*)
- g. Pengubahan data dalam sebuah *file* atau tabel (*UPDATE*)
- h. Penghapusan data dari sebuah *file* atau tabel (*DELETE*)

Operasi pembuatan basis data dan tabel merupakan operasi awal yang hanya dilakukan sekali dan berlaku seterusnya. Sedangkan untuk operasi pengisian data merupakan operasi rutin yang dilakukan berulang-ulang. (Yakub, 2008:36).

#### 4. Persyaratan Basis Data

Ketentuan yang harus diperhatikan pada pembuatan file basis data agar dapat memenuhi kriteria sebagai basis data, yaitu: redundansi data, inkonsistensi data, pengaksesan data, data terisolasi untuk *standarisasi*, masalah keamanan, masalah *integritas* data, data *multiuser*.

##### a. *Redudansi dan Inkonsistensi Data*

Penyimpanan data yang sama di beberapa tempat disebut *redudansi*, hal ini akan menyebabkan pemborosan dan menimbulkan *inkonsistensi* data (data tidak konsisten) karena bila terjadi maka data harus dirubah pada beberapa tempat, hal ini tentunya tidak efisien.

##### b. *Pengaksesan Data*

Data di dalam basis data harus siap diakses oleh siapa saja yang membutuhkan dan mempunyai hak untuk mengaksesnya. Oleh karena itu perlu dibuat suatu program pengelolaan atau suatu aplikasi

untuk mengakses data yang dikenal sebagai *Database Management System* (DBMS).

**c. Data Terisolasi untuk Standarisasi**

Jika data tersebar dalam beberapa file dalam bentuk format yang tidak sama, maka akan menyulitkan dalam menulis program aplikasi, baik untuk mengambil dan menyimpan data. Oleh karena itu ada dalam satu database harus dibuat satu format yang sama, sehingga mudah dibuat program aplikasinya.

**d. Masalah Keamanan atau *Security***

Setiap pemakai sistem basis data tidak semua bagian diperbolehkan untuk mengakses semua data, misalnya data mengenai gaji pegawai hanya boleh dibuka oleh bagian keuangan, sedang bagian gudang dan bagian lain tidak diperkenankan untuk membukanya. Keamanan dapat diatur dan disesuaikan baik ditingkat basis data atau aplikasinya.

**e. *Multiple User***

Salah satu alasan basis data dibangun karena nantinya data tersebut akan digunakan oleh banyak orang, baik dalam waktu berbeda maupun bersamaan. Oleh karena itu diperlukan basis data yang handal dan dapat mendukung banyak pemakai atau *multiuser*. (Yakub, 2008:36).

## 5. Sistem Basis Data

### a. Pengertian Sistem Basis Data

Sistem basis data (*database*) merupakan sistem yang terdiri dari kumpulan file atau tabel yang saling berhubungan dan memungkinkan beberapa mengakses dan memanipulasinya. Sistem basis data juga merupakan suatu sistem yang menyusun dan mengelola data organisasi perusahaan, sehingga mampu menyediakan informasi yang diperlukan oleh pemakai. Istilah sistem basis data tentu saja berbeda dengan istilah basis data, sistem basis data merupakan lingkup yang lebih luas daripada basis data. (Yakub, 2008:37)

### b. Komponen Sistem Basis Data

Sistem basis data terdapat komponen-komponen utama yaitu; perangkat keras (*hardware*), sistem operasi (*operating system*), basis data (*database*), program aplikasi (*application program*), *Database Management System* (DBMS), dan pemakai (*user*).

### c. *Database Management System* (DBMS)

*Database Management System* (DBMS) merupakan kumpulan program aplikasi yang digunakan untuk membuat dan mengelola basis data. DBMS berisi suatu koleksi data dan satu set program untuk mengakses data. DBMS merupakan perangkat lunak (*software*) yang menentukan bagaimana data tersebut diorganisasi, disimpan, diubah dan diambil kebal. Perangkat ini juga

menerapkan mekanisme pengamanan data, pemakaian data bersama, dan konsistensi data.

**d. Pemakai (*Users*)**

Pemakai atau *users* adalah beberapa jenis atau tipe pemakai pada sistem basis data, berdasarkan cara mereka berinteraksi pada basis data, diantaranya program aplikasi, pemakai mahir, pemakai umum dan pemakai khusus.

1) *Programmer* Aplikasi

*Programmer* aplikasi adalah pemakai yang berinteraksi dengan basis data melalui DML (*Data Manipulation Language*), yang disertakan dalam program yang ditulis dalam bahasa pemrograman induk (seperti *pascal*, *cobol*, *clipper*, *foxpro*, dan lain – lain).

2) *User* Mahir (*Casual User*)

*Casual user* adalah pemakai yang berinteraksi dengan sistem tanpa menulis modul program. Pemakai menggunakan *query* (untuk akses data), dengan bahasa *query* yang telah disediakan oleh suatu DBMS.

3) *User* Umum (*Naive User*)

*Naive user* adalah pemakai yang berinteraksi dengan sistem basis data melalui pemanggilan satu program aplikasi permanen, yang telah ditulis atau disediakan sebelumnya.

#### 4) *User Khusus (Specialized User)*

*Specialized User* adalah pemakai yang menulis aplikasi basis data non konvensional untuk keperluan khusus, seperti untuk aplikasi sistem pakar, pengolahan citra, dan lain-lain.

### 6. **Administrator Basis Data**

Sebuah lingkup basis data seharusnya mempunyai satu orang atau sekelompok orang pada bagian struktur basis data untuk menangani administrasi basis data yang biasa disebut administrator basis data atau *Database Administrator (DBA)*. *Administrator* basis data adalah orang yang bertanggung jawab dan bekerjasama dengan analisis sistem dan *user-user* lain guna melengkapi berbagai tugas seperti; mendefinisikan data, pemodelan data, desain basis data, serta menjamin keserasian integritas data. (Yakub, 2008:39).

### G. **Flowchart (Diagram Alur)**

#### 1. Pengertian *flowchart* (Diagram Alur).

Karena komputer membutuhkan hal-hal yang rinci, maka bahasa pemrograman bukanlah alat baik untuk merancang sebuah *algoritma* awal. Alat yang banyak dipakai untuk membuat *algoritma* adalah diagram alur (*flowchart*).

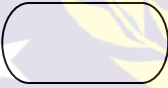

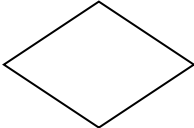
Diagram alur dapat menunjukkan secara jelas arus pengendalian suatu *algoritma*, yakni melaksanakan suatu rangkaian kegiatan secara *logis* dan *sistematis*. Suatu diagram alur dapat memberi gambaran dua *dimensi* berupa simbol-simbol grafis. Masing-masing simbol telah

ditetapkan lebih dahulu fungsi dan artinya. Simbol-simbol tersebut dipakai untuk menunjukkan berbagai kegiatan operasi dan jalur pengendalian. Arti khusus dari sebuah *flowchart* adalah simbol-simbol yang digunakan untuk menggambarkan urutan proses yang terjadi di dalam suatu program komputer secara *sistematis* dan *logis*. (Sutabri; 2004; 21).

2. Simbol-simbol *flowchart*.

Sudah dikemukakan di atas bahwa diagram alur atau *flowchart* memiliki beberapa simbol yang biasa digunakan untuk menggambarkan rangkaian proses yang harus dilaksanakan. Simbol-simbol tersebut dijelaskan di bawah ini: (Sutabri; 2004; 21-22)

Tabel 2.1. Simbol *Flowchart*

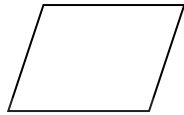
Simbol <i>Flowchart</i>	Fungsi
<i>TERMINAL</i> 	Simbol ini digunakan untuk mengawali atau mengakhiri suatu proses/kegiatan.
<i>PREPARATION</i> 	Simbol ini digunakan untuk mempersiapkan harga awal/nilai awal suatu variabel yang akan diproses.
<i>DECISION</i> 	Simbol ini digunakan untuk pengujian suatu kondisi yang sedang diproses.

## PROSES



Menggambarkan suatu proses yang sedang dieksekusi.

## INPUT/OUTPUT



Menggambarkan proses input (*read*) maupun proses output (*print*).

## SUBROUTINE



Menggambarkan proses pemanggilan subprogram dari main program.

## FLOW LINE



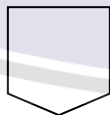
Menggambarkan arus proses dari suatu kegiatan ke kegiatan lain.

## CONNECTOR



Simbol ini digunakan sebagai penghubung antara suatu proses dengan proses lainnya yang ada di dalam suatu lembar halaman.

## PAGE CONNECTOR



Simbol ini digunakan sebagai penghubung antara suatu proses dengan proses lainnya, tetapi berpindah halaman.

## MANUAL OPERATION



Menggambarkan suatu kegiatan atau proses yang bersifat manualisasi.



#### PRINTER



Menggambarkan suatu kegiatan mencetak suatu informasi dengan mesin printer.

#### CONSOLE



Menggambarkan suatu kegiatan menampilkan data atau informasi melalui monitor atau CRT (*Cathode Ray Tube*).

#### DISK



Menggambarkan suatu kegiatan membaca atau menulis data menggunakan media *magnetic disk*.

#### MANUAL INPUT



Simbol ini digunakan untuk menggambarkan proses pemasukan data melalui media keyboard.

#### TAPE



Simbol ini digunakan untuk menggambarkan suatu kegiatan membaca atau menulis data menggunakan media *magnetic tape*.

---

Sumber: Analisis Sistem Informasi (Sutabri , 2004)

### 3. Jenis *flowchart*.

Bentuk diagram alur (*flowchart*) yang sering digunakan dalam proses pembuatan suatu program komputer adalah sebagai berikut:

a. Program *flowchart*.

Simbol-simbol yang menggambarkan proses secara rinci dan detail antara intruksi yang satu dengan intruksi yang lainnya dalam suatu program komputer yang bersifat *logic*.

b. Sistem *flowchart*.

Simbol-simbol yang menggambarkan urutan prosedur secara detail dalam suatu sistem komputerisasi. Bersifat fisik.

c. Teknik pembuatan *flowchart*.


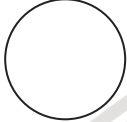

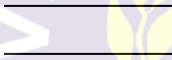
Sebelum kita membuat sebuah program komputer, yang harus kita lakukan sebelumnya adalah membuat *flowchart*. Jenis *flowchart* yang sering digunakan adalah program *flowchart*.

## H. **Data Flow Diagram (DFD)**

*Data Flow Diagram (DFD)* adalah representasi grafik dari sebuah sistem. *DFD* menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana komponen-komponen tersebut, dan asal, tujuan, dan penyimpanan dari data tersebut. Kita dapat menggunakan *DFD* untuk dua hal utama, yaitu untuk membuat dokumentasi dari sistem informasi yang ada, atau untuk menyusun dokumentasi untuk sistem informasi yang baru

Empat simbol yang digunakan:

Tabel 2.2. Simbol *DFD*

Simbol	Fungsi
	Simbol Entitas eksternal atau <i>terminator</i> menggambarkan asal atau tujuan data di luar sistem
	Simbol lingkaran menggambarkan entitas atau proses dimana aliran data masuk ditransformasikan ke aliran data keluar
	Simbol aliran data menggambarkan aliran data
	Simbol <i>file</i> menggambarkan tempat data disimpan

Sumber: Analisis Sistem Informasi (Sutabri , 2004)

Ada 3 (tiga) jenis *DFD*, yaitu :

1. *Diagram contex.*

Jenis pertama *Context Diagram*, adalah *data flow diagram* tingkat atas (*DFD Top Level*), yaitu diagram yang paling tidak detail, dari sebuah sistem informasi yang menggambarkan aliran-aliran data ke dalam dan ke luar sistem dan ke dalam dan ke luar *entitas-entitas eksternal*. (CD menggambarkan sistem dalam satu lingkaran dan hubungan dengan *entitas* luar. Lingkaran tersebut menggambarkan keseluruhan proses dalam sistem).

Beberapa hal yang harus diperhatikan dalam menggambar CD :

a. Terminologi sistem:

- 1) Batas Sistem adalah batas antara “daerah kepentingan sistem”.
- 2) Lingkungan Sistem adalah segala sesuatu yang berhubungan atau mempengaruhi sistem tersebut.
- 3) *Interface* adalah aliran yang menghubungkan sebuah sistem dengan lingkungan sistem tersebut.

b. Menggunakan satu simbol proses.

Yang masuk didalam lingkaran konteks (simbol proses) adalah kegiatan pemrosesan informasi (Batas Sistem). Kegiatan informasi adalah mengambil data dari *file*, *mentransformasikan* data, atau melakukan *filig* data, misalnya mempersiapkan dokumen, memasukkan, memeriksa, mengklasifikasi, mengatur, menyortir, menghitung, meringkas data, dan melakukan *filig* data (baik yang melakukan secara manual maupun yang dilakukan secara *terotomasi*).

c. Nama/keterangan di simbol proses tersebut sesuai dengan fungsi sistem tersebut.

d. Antara *Entitas Eksternal/Terminator* tidak diperbolehkan komunikasi langsung.

e. Jika terdapat *termintor* yang mempunyai banyak masukan dan keluaran, diperbolehkan untuk digambarkan lebih dari satu sehingga mencegah penggambaran yang terlalu rumit, dengan memberikan tanda *asterik* ( \* ) atau garis silang ( # ).

f. Jika Terminator mewakili individu (personil) sebaiknya diwakili oleh peran yang dipertainkan *personil* tersebut.

g. Aliran data ke proses dan keluar sebagai output keterangan aliran data berbeda.

## 2. *DFD fisik.*

*DFD fisik* adalah *representasi* grafik dari sebuah sistem yang menunjukkan *entitas-entitas internal* dan *eksternal* dari sistem tersebut, dan aliran-aliran data ke dalam dan keluar dari *entitas-entitas* tersebut. *Entitas-entitas internal* adalah *personal*, tempat (sebuah bagian), atau mesin (misalnya, sebuah komputer) dalam sistem tersebut yang *mentransformasikan* data. Maka *DFD fisik* tidak menunjukkan apa yang dilakukan, tetapi menunjukkan dimana, bagaimana, dan oleh siapa proses-proses dalam sebuah sistem dilakukan.

Perlu diperhatikan didalam memberikan keterangan di lingkaran-lingkaran (simbol proses) dan aliran-aliran data (simbol aliran data) dalam *DFD fisik* menggunakan label/keterangan dari kata benda untuk menunjukkan bagaimana sistem *mentransmisikan* data antara lingkaran-lingkaran tersebut.

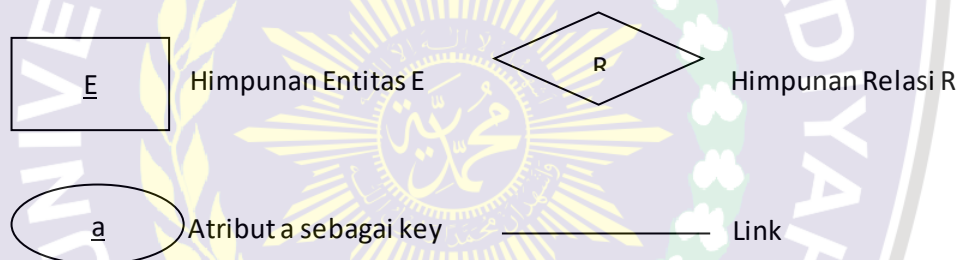
## 3. *DFD logis.*

*DFD Logis* adalah *representasi* grafik dari sebuah sistem yang menunjukkan proses-proses dalam sistem tersebut dan aliran-aliran data ke dalam dan ke luar dari proses-proses tersebut. Kita menggunakan *DFD logis* untuk membuat dokumentasi sebuah sistem informasi karena *DFD logis* dapat mewakili logika tersebut, yaitu apa yang dilakukan oleh sistem tersebut, tanpa perlu menspesifikasi dimana, bagaimana, dan oleh siapa proses-proses dalam sistem tersebut dilakukan. Keuntungan dari

*DFD logis* dibandingkan dengan *DFD fisik* adalah dapat memusatkan perhatian pada fungsi - fungsi yang dilakukan sistem.

## I. ERD (*Entity Relationship Diagram*)

Model *Entity-Relationship* yang berisi komponen-komponen Himpunan Entitas dan Himpunan Relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta dari 'dunia nyata' yang kita tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan Diagram *Entity-Relationship* (Diagram E-R). Notasi-notasi simbolik didalam Diagram E-R yang dapat kita gunakan adalah: (Sutabri , 2004)



Gambar 2.1 Kardinalitas relasi

1. Persegi panjang, menyatakan Himpunan Entitas.
2. Lingkaran/*Ellips*, menyatakan Atribut (Atribut yang berfungsi sebagai *key* digarisbawahi).
3. Belah ketupat, menyatakan Himpunan Relasi.
4. Garis, sebagai penghubung antara Himpunan Relasi dengan Himpunan Entitas dengan Atributnya.
5. Kardinalitas relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk relasi satu-ke-satu, 1 dan N

untuk relasi satu-ke-banyak atau  $N$  dan  $N$  untuk relasi banyak-ke-banyak).

