

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Sistem Informasi**

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial dari kegiatan strategi dari suatu organisasi, sistem informasi ini terdiri dari komponen-komponen dalam organisasi yang digunakan untuk mencapai suatu tujuan (Ladjamudin, 2013:14).

#### **2.2 Elemen Sistem**

Beberapa elemen yang membentuk sebuah sistem menurut Kadir (2003:54) diantaranya adalah sebagai berikut:

1. Tujuan

Setiap sistem memiliki tujuan (*goal*), dimana tujuan antara satu sistem dengan sistem lain berbeda-beda. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Begitu pula yang berlaku pada Sistem Informasi.

Walaupun begitu tujuan yang umum ada tiga macam yaitu:

1. Untuk mendukung fungsi kepengurusan manajemen.
2. Untuk mendukung pengambilan keputusan manajemen.

3. Untuk mendukung operasi perusahaan.

## 2. Masukan (*Input*)

Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud (tampak secara fisik) maupun yang tidak tampak. Contoh: masukan yang berwujud adalah informasi.

## 3. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna, misalnya berupa informasi dan produk.

## 4. Keluaran (*Output*)

Keluaran merupakan hasil dari pemrosesan. Pada Sistem Informasi keluaran bisa berupa informasi, saran, cetakan laporan, dan sebagainya.

## 5. Mekanisme Pengendalian dan Umpan Balik

Mekanisme pengendalian (kontrol mekanisme) diwujudkan dengan menggunakan umpan balik (*feedback*), yang menciptakan keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai.

## 6. Batas

Batasan (*boundary*) sistem adalah pemisah antara sistem dan daerah luar sistem (lingkungan). Batasan sistem menentukan konfigurasi, ruang lingkup atau kemampuan sistem.

## 7. Lingkungan

Lingkungan adalah segala sesuatu yang berada diluar sistem. Lingkungan bisa berpengaruh terhadap operasi sistem dalam arti bisa merugikan atau menguntungkan sistem itu sendiri.

### 2.3 Analisis sistem

Analisa sistem adalah teori sistem umum yang sebagai sebuah landasan konseptual yang mempunyai tujuan untuk memperbaiki berbagai fungsi didalam sistem yang sedang berjalan agar menjadi lebih efisien, mengubah sasaran sistem yang sedang berjalan, merancang/mengganti *output* yang sedang digunakan, untuk mencapai tujuan yang sama dengan seperangkat *input* yang lain (biasa jadi lebih sederhana dan lebih interatif) atau melakukan beberapa perbaikan serupa, Menurut Mulyato (2009:125).

#### 2.3.1 Tahap-tahap Analisis Sistem

Tahapan analisis sistem di mulai karena adanya permintaan terhadap sistem baru. Tujuan utama analisis sistem adalah untuk menentukan hal-hal detail tentang yang akan dikerjakan oleh sistem yang diusulkan (Kadir, 2003).

Menurut Mulyanto (2009:129), Di dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh seorang analis sistem, diantaranya adalah:

1. Identify, yaitu proses yang dilakukan untuk mengidentifikasi masalah.
2. Understand, yaitu memahami kerja dari sistem yang ada.
3. Analysis, yaitu melakukan analisa terhadap sistem.

4. Report, yaitu membuat laporan dari hasil analisis yang telah dilakukan dalam kurun waktu tertentu.

### **2.3.2 Fungsi Analisis Sistem**

Adapun fungsi analisa sistem adalah sebagai berikut:

1. Mengidentifikasi masalah–masalah kebutuhan pemakai (*user*).
2. Menyatakan secara spesifik sasaran yang harus dicapai untuk memenuhi kebutuhan pemakai.
3. Memilih alternatif–alternatif metode pemecahan masalah paling tepat.
4. Merencanakan dan menerapkan rancangan sistemnya. Pada tugas atau fungsi terakhir dari analisa sistem menerapkan rencana rancangan sistemnya yang telah disetujui oleh pemakai.

## **2.4 Desain sistem**

Desain sistem adalah Proses dari penggambaran, Pengaturan, dan susunan dari komponen sebuah sistem pada kedua tingkat keterkaitan dan tingkat detail dengan melihat perancangan sistem yang akan diajukan (Satzinger, dkk 2009).

Tahap desain sistem mempunyai dua tujuan utama, yaitu sebagai berikut :

1. Untuk memenuhi kebutuhan kepada pemakai sistem.
2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknik lainnya yang terlibat. Tujuan kedua ini lebih condong pada desain sistem yang terinci, yaitu pembuatan rancang bangun yang jelas dan lengkap untuk nantinya digunakan untuk pembuatan progam komputernya.

Untuk mencapai tujuan ini, analisis sistem harus dapat mencapai sasaran-sasaran sebagai berikut :

1. Desain sistem harus berguna, mudah dipahami dan nantinya mudah digunakan. Ini berarti bahwa data harus mudah ditangkap, metode-metode harus mudah diterapkan dan informasi harus mudah dihasilkan serta mudah dipahami dan digunakan.
2. Desain sistem harus dapat mendukung tujuan utama perusahaan sesuai dengan yang didefinisikan pada tahap perencanaan sistem yang dilanjutkan pada tahap analisis sistem.
3. Desain sistem harus efisien dan efektif untuk dapat mendukung pengolahan transaksi, pelaporan manajemen dan mendukung keputusan yang akan dilakukan oleh manajemen, termasuk tugas-tugas yang lainnya yang tidak dilakukan oleh komputer.
4. Desain sistem harus dapat mempersiapkan rancang bangun yang terinci untuk masing-masing komponen dari Sistem Informasi meliputi data, informasi, simpanan data, metode-metode, prosedur-prosedur, orang-orang, perangkat keras, perangkat lunak dan pengendalian *internal*.

## **2.5 Beasiswa PPA**

Beasiswa adalah berupa bantuan keuangan yang diberikan kepada perorangan yang bertujuan untuk digunakan demi keberlangsungan pendidikan yang ditempuh. Beasiswa dapat diberikan oleh lembaga pemerintah, perusahaan ataupun yayasan (Lahinta, 2009).

Beasiswa harus diberikan kepada penerima yang layak. Beasiswa adalah tunjangan uang yang diberikan kepada pelajar atau mahasiswa sebagai bantuan biaya belajar (KBBI, 1994). Pemberian beasiswa dapat dikategorikan pada pemberian cuma-cuma ataupun pemberian dengan ikatan kerja (biasa disebut ikatan dinas) setelah selesainya pendidikan. Lama ikatan dinas ini berbeda-beda, tergantung pada lembaga yang memberikan beasiswa tersebut. Dalam menentukan siapa yang berhak menerima beasiswa, harus dilakukan seleksi-seleksi khusus sesuai dengan kriteria yang ditetapkan.

Pada penerimaan beasiswa Peningkatan Prestasi Akademi (PPA) di Jurusan Ilmu Komputer FMIPA UNILA ini memiliki kriteria atau faktor bobot penilaian antara lain:

1. Mahasiswa harus memiliki kriteria Indeks Prestasi Kumulatif (IPK) setidaknya 3,0, karena pada beasiswa PPA ini IPK merupakan kriteria utama yang memiliki bobot penilaian terbesar.
2. Pada jenjang S1, tingkatan mahasiswa duduk pada semester II dan paling tinggi duduk pada semester VII.
3. Pada jenjang D3, tingkatan mahasiswa duduk pada semester II dan paling tinggi duduk pada semester V.
4. Penghasilan orang tua, tanggungan orang tua, dan jumlah saudara kandung, prestasi-prestasi di kegiatan akademik maupun prestasi ekstra kurikuler (olahraga, teknologi, seni/budaya tingkat internasional/dunia, Regional/Asia/Asean dan Nasional), merupakan kriteria utama yang akan menjadi pertimbangan apakah mahasiswa tersebut layak menerima atau tidak layak menerima beasiswa.

## 2.6 Pengertian Algoritma

Algoritma merupakan pola pikir yang terstruktur yang berisi tahap-tahap penyelesaian suatu masalah, yang nantinya akan diimplementasikan ke dalam suatu bahasa pemrograman (Kristanto, 2009: 9).

Biasanya dalam program, algoritma digunakan untuk tiga struktur program yaitu struktur urut (*sequence structure*), struktur keputusan (*decision structure*), dan struktur perulangan (*looping structure*).

## 2.7 Sorting

Metode sorting adalah suatu metode yang digunakan untuk proses pengurutan data yang sebelumnya disusun secara acak atau tidak teratur sehingga menjadi urut dan tersusun secara teratur menurut aturan tertentu (Kristanto, 2009 : 213).

### 2.7.1 *Quick Sort*

Metode sorting yang penulis gunakan pada penelitian ini adalah *quick sort* dan *selection sort*. Metode pengurutan *quicksort* adalah dengan membandingkan suatu elemen (*pivot*) dengan elemen yang lain dan menyusunnya sedemikian rupa sehingga elemen-elemen lain yang lebih kecil dari *pivot* tersebut terletak di sebelah kirinya dan elemen-elemen lain yang lebih besar daripada *pivot* tersebut terletak di sebelah kanan. Sehingga dengan demikian terbentuk dua sublist yaitu yang terletak disebelah kiri *pivot* dan sebelah kanan *pivot* (Kristanto, 2009: 239).

Algoritma pada proses *sorting* dengan menggunakan metode *quick sort* adalah sebagai berikut:

1. Tentukan data-data yang akan diurutkan dan disimpan dalam array.
2. Tentukan pivot dari data tersebut dengan memilih data terkecil.
3. Data-data yang ada dibagi menjadi 2 bagian dimana bagian yang satu dengan yang lain dilakukan proses pengecekan.
4. Pada bagian yang pertama merupakan data pivot, lakukan perbandingan pada bagian ke dua antara data yang satu dengan data yang lain, setelah menemukan data terkecil lalu bandingkan data tersebut dengan pivot, jika lebih data tersebut lebih kecil dari data pivot maka tukar data dengan pivot, namun jika lebih besar maka letakkan data pada indeks 2. Hal ini berlaku sampai data seterusnya.
5. Tampilkan data hasil perbandingan.
6. Ulangi langkah 4, sampai semua data dibandingkan.
7. Selesai.

### **2.7.2 Selection Sort**

Metode *sorting selection sort* adalah dengan membandingkan elemen yang sekarang dengan elemen yang berikutnya sampai dengan elemen yang terakhir. Jika ditemukan elemen lain yang lebih kecil dari elemen sekarang maka dicatat posisinya dan kemudian ditukar dan begitu seterusnya (Kristanto, 2009:236).

Menurut (Kristanto, 2009:239) algoritma pada proses *sorting* dengan menggunakan metode *selection sort* adalah sebagai berikut:

1. Tentukan data-data yang akan diurutkan dan disimpan dalam array.
2. Lakukan pengulangan dari data-data tersebut.
3. Lakukan perbandingan antara data yang satu dengan data yang lain, dimana jika data yang satu lebih kecil dari data yang lain, maka posisinya ditukar. Kalau tidak, posisinya tetap.
4. Tampilkan data hasil perbandingan.
5. Ulangi langkah ke 3 sampai semua data dibandingkan.
6. Selesai.

## 2.8 XAMPP

*Xampp* digunakan sebagai aplikasi pembuat DBMS ketika di *localhost*. Di dalam *Xampp* terdapat *MySQL* sebagai fungsi-fungsi dalam penggunaan *database* dan *Apache* sebagai Web Server. Sehingga dengan menginstal *Xampp* maka tidak perlu lagi melakukan instalasi dan konfigurasi web server *apache*, *PHP* dan *MySQL* secara manual karena *Xampp* akan menginstalasi dan mengkonfigurasi secara otomatis.

Menurut Kadir (2005) *XAMMP* merupakan suatu paket *software* yang terdiri dari *Apache*, *MySQL*, dan *PHP*. *Apache* adalah sebuah aplikasi yang memungkinkan suatu komputer menjadi *web server*. *MySQL* adalah DBMS (*Database Management System*), yaitu suatu sistem yang berfungsi untuk mengolah data

dalam *database*. Sedangkan *PHP* adalah bahasa pemrograman *server side coding* yang sering digunakan untuk menciptakan halaman *web*.



Gambar 2.1 Logo Xampp

## 2.9 MySQL

*MySql* merupakan salah satu jenis *database server* yang sangat terkenal, kepopulerannya disebabkan karena *MySql* menggunakan *SQL* sebagai bahasa dasar untuk mengakses *database* nya, selain itu *MySql* juga bersifat *opensource* (gratis) pada berbagai platform kecuali untuk jenis enterprise, yang bersifat komersial (Kadir, 2008).

Ulf Micheal Widenius adalah penemu awal versi pertama *MySql* yang kemudian pengembangan selanjutnya dilakukan oleh perusahaan *MySql AB*. *MySql AB* yang merupakan sebuah perusahaan komersial yang didirikan oleh para pengembang *MySql*.

Sukamto dan Shalahuddin (2011) di dalam bukunya mengatakan bahwa *MySql* menggunakan bahasa basis data *SQL (structured Query Language)* yang merupakan bahasa yang digunakan untuk mengelola data pada sistem manajemen basisdata relasional (RDBMS). *MySql* juga sering disebut sebagai sebuah perangkat lunak system manajemen basis data *SQL (DBMS)* yang *multithread*, dan *multi-user*, hal ini dikarenakan hampir semua *DBMS* mengadopsi *SQL* sebagai bahasa pengolahan data pada *DBMS*. *SQL (Structured Query Language)* adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau

seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Anhar (2010) menyatakan beberapa kelebihan MySQL:

1. MySQL dapat berjalan dengan stabil pada berbagai sistem operasi, seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris, dll.
2. Bersifat *open source*, MySQL didistribusikan secara *open source* (gratis), di bawah lisensi GNU *General Public Licence (GPL)*.
3. Bersifat *multiuser*, MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah.
4. MySQL memiliki kecepatan yang baik dalam menangani *query* (perintah SQL). Sehingga dapat memproses lebih banyak SQL per satuan waktu.
5. Dari segi *security* atau keamanan data, MySQL memiliki beberapa lapisan *security*, seperti level *subnet mask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* terenskripsi.
6. MySQL juga memiliki *interface* (antarmuka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).



Gambar 2.2 Logo My SQL

## 2.10 Web Browser

*Web browser* adalah *software* atau perangkat lunak yang berguna untuk mengakses informasi *web* ataupun untuk melakukan transaksi via web dari *server* web (Kustiyahningsih dan Anamisa, 2011:8).

Pada mulanya *web browser* hanya berorientasi pada teks saja, namun kini *web browser* sudah dapat berorientasi gambar dan bahkan dapat memutar *file* multimedia seperti video dan suara. Selain itu *web browser* juga dapat digunakan untuk mengirim dan menerima email, mengelolah HTML sebagai *input* dan halaman web sebagai hasil *output* yang informatif.

Menurut (Kustiyahningsih, 2011:8) fungsi dari *browser* antara lain adalah untuk melakukan atau menampilkan halaman web atau interaksi dengan dokumen yang disediakan oleh server. Setiap jenis *browser* memiliki perbedaan, kelebihan serta kekurangan masing masing hal ini disebabkan perbedaan produsen atau perusahaan yang menyediakan *browser* tersebut.

### **2.11 *Php Hypertext Preprocessor***

PHP adalah singkatan dari *PHP Hypertext Preprocessor* yang dikenal sebagai bahasa pemrograman yang kodenya dijalankan di sisi server sehingga source code program tidak akan terlihat pada sisi klien atau *browser* (Kadir, 2011).

PHP menggunakan bahasa pemrograman yang berbentuk script yang diletakkan di dalam web server. PHP ini merupakan bahasa yang hanya dapat berjalan pada *server* yang hasilnya dapat ditampilkan pada klien. Interpreter PHP dalam mengeksekusi kode PHP pada sisi server disebut *serverside*, berbeda dengan mesin maya Java yang mengeksekusi program pada sisi klien (client-server). (Kasiman Peranginangin, 2009).

Purwanto (2001) PHP dibuat pada tahun 1994 oleh Rasmus Lerdorf, seorang pemrograman C yang handal, semula PHP hanya digunakan untuk mencatat jumlah pengunjung pada *Homepage*-nya. PHP merupakan sebuah produk yang berbentuk open source, sehingga source code dari PHP dapat digunakan, diganti, diubah tanpa harus membayar atau dikenakan biaya.

Kriteria yang harus diperhatikan dalam penulisan *skrip* PHP, menurut Nugroho (2004) yaitu sebagai berikut:

1. Setiap halaman yang mengandung *skrip* PHP harus disimpan dengan bentuk ekstensi PHP sesuai dengan program PHP yang mendukungnya: *namafilename.php*.
2. Setiap *skrip* PHP harus didahului dengan pembuka PHP (<?) dan kemudian diakhiri dengan penutup (?>).
3. Setiap baris pada *skrip* isi didahului pernyataan cetak yang dibedakan menjadi dua yaitu *print* dan *echo*.
4. Setiap akhir baris perintah harus diakhiri dengan titik koma (;).
5. Semua bentuk variabel harus diawali dengan tanda dolar.
6. Penulisan *comment* atau keterangan didahului dengan tanda pembuka (\*) dan diakhiri dengan tanda (\*) atau menggunakan tanda *slash* ganda (//) yang hanya digunakan untuk pesan yang informasinya satu baris saja.
7. Semua *listing program* HTML yang akan digabungkan dalam skrip PHP harus dihilangkan tanda petik ganda dan dapat digantikan dengan tanda petik tunggal atau menghilangkannya.

**Kelebihan PHP antara lain sebagai berikut:**

- a. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- b. Web Server yang mendukung PHP dapat ditemukan dimana-mana dari mulai apache, IIS, hingga Xitami dengan konfigurasi yang relatif mudah.
- c. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.
- d. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
- e. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui console serta juga dapat menjalankan perintah-perintah *system*.
- f. PHP merupakan perangkat lunak yang *open source*.
- g. Merupakan bahasa *server-side scripting* yang dinamis.
- h. Seluruh aplikasi berbasis *website* dapat diubah dengan PHP.
- i. Merupakan bahasa pemrograman yang dapat dikembangkan sendiri seperti untuk menambah fungsi-fungsi baru.
- j. Akses ke sistem *database* yang lebih fleksibel, seperti MySQL.
- k. Memiliki banyak konektivitas dengan sistem *database*. Sistem *database* yang didukung PHP antara lain Oracle, MySQL, Sybase, PostgreSQL dll.
- l. PHP dapat berjalan pada berbagai jenis *web server* seperti PWS (*Personal Web Server*), Apache, IIS, AOLServer, fhttpd, phttpd, dan sebagainya.
- m. Mendukung komunikasi dengan layanan seperti protocol IMAP, SNMP, NNTP, POP3, dan bahkan HTTP.

- n. PHP dapat diinstal sebagai bagian atau modul dari *Apache web server* atau sebagai *CGI script* yang mandiri.
- o. Waktu eksekusi yang lebih cepat dibandingkan dengan bahasa pemrograman *web* lainnya yang berorientasi pada *server-side scripting*. (Kadir, 2007).

**Adapun Kelemahan PHP adalah:**

- a. Tidak detail untuk pengembangan skala besar.
- b. Tidak bisa memisahkan antara tampilan dengan logik dengan baik (walau penggunaan template dapat memperbaikinya).
- c. PHP memiliki kelemahan *security* tertentu apabila programmer tidak jeli dalam melakukan pemrograman dan kurang memperhatikan isu dan konfigurasi PHP.



Gambar 2.3 Logo PHP

## **2.12 Database**

Basis data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi (Kadir, 2003).

Basis data dimaksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas. Berbeda dengan sistem *file* yang menyimpan data secara terpisah, pada basis data sebuah data tersimpan secara terintegrasi.

Untuk mengelola basis data diperlukan perangkat lunak yang disebut DBMS. DBMS adalah sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data (Sukamto dan Shalahuddin, 2011).

Komponen penyusun utama dari sebuah basis data:

1. *Hardware*

Biasanya berupa perangkat komputer standar, media penyimpanan sekunder, dan media komunikasi untuk sistem jaringan.

2. *Operating System*

Merupakan perangkat lunak yang berfungsi mengendalikan seluruh sumber daya dan melakukan operasi dasar dalam sistem komputer.

3. *Database*

*Database* merupakan basis data yang mewakili sistem tertentu untuk dikelola. Sebuah sistem basis data bisa terdiri dari lebih dari satu basis data.

4. DBMS (*Database Management System*)

Perangkat lunak yang digunakan untuk mengelola basis data. Contoh kelas sederhana: *dBase, Foxbas, Rbase, Microsoft Access, Microsoft Foxpro, Borland Paradox*. Contoh kelas kompleks: *Borland-Interbase, MS. SQL Server, Oracle, Informix, Sybase*.

5. *User* (pengguna sistem basis data)

Orang yang berinteraksi dengan sistem basis data, mulai dari yang merancang sampai yang menggunakan di tingkat akhir.

## 6. Optional *Software*

Perangkat lunak pelengkap yang mendukung. Bersifat Abstraksi data pada sebuah basis data yang merupakan level dalam bagaimana melihat data dalam sebuah sistem basis data. Biasanya pengguna hanya tau bagaimana data itu terlihat tanpa tahu bagaimana data tersebut disimpan dan dipelihara. Abstraksi data pada basis data terdiri dari 3 level yaitu level eksternal, level konseptual, dan level *internal*.

### 2.13 Diagram Konteks (*Context Diagram*)

Diagram konteks adalah sebuah diagram sederhana yang menggambarkan hubungan antara entity luar, masukan, dan keluaran dari sistem. Diagram konteks direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem (Kristanto, 2003).

Al-Bahra bin Ladjamudin (2013) Mengatakan bahwa diagram konteks ini merupakan level tertinggi dari DFD yang menggambarkan seluruh *input* ke sistem atau *output* dari sistem. Diagram konteks ini akan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus) dan pada diagram konteks ini hanya ada satu proses serta tidak boleh ada data *store* dalam diagram konteks. Untuk menggambarkan diagram konteks deskripsikan data apa saja yang dibutuhkan oleh sistem dan dari mana sumbernya, serta informasi apa saja yang akan dihasilkan sistem dan kemana informasi tersebut akan diberikan.

## 2.14 ERD

ERD (*Entity Relationship Diagram*) merupakan diagram sederhana yang menggambarkan desain *database* dari suatu sistem. ERD pertama kali diperkenalkan oleh Chen pada tahun 1976 dan menjadi teknik permodelan data yang dominan sejak 15 tahun yang lalu (Lane, 2004).

Dalam rekayasa perangkat lunak *Entity-Relationship* adalah salah satu metode pemodelan basis data yang digunakan untuk menghasilkan skema konseptual untuk jenis/model data semantik sistem. ERD juga menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Diagram untuk menggambarkan model *Entity-Relationship* ini disebut *Entity-Relationship diagram*, *ER diagram*, atau *ERD*. Model ini juga membantu perancangan basis data, karena model ini dapat menunjukkan berbagai macam data yang dibutuhkan dan keserasian antara data di dalamnya. Bagi pemakai, model ini sangat membantu dalam hal pemahaman model sistem dan rancangan basis data yang akan dikembangkan oleh perancang basis data.

Tabel 2.1 Simbol-simbol yang digunakan dalam Entity Relationship Diagram sebagai berikut:

 Menunjukkan <i>Entity</i>	 Menunjukkan Atribut
 Menunjukkan Hubungan	 Menunjukkan Garis

### 1. *Entity*

Suatu obyek yang dapat didefinisikan dalam lingkungan pemakai dalam konteks sistem yang telah dibuat. Entity digunakan atau digambarkan persegi empat.

### 2. Atribut

Atribut merupakan pendeskripsian karakteristik dari entitas. Atribut digambarkan dalam bentuk lingkaran atau elips. Atribut yang menjadi kunci entitas atau *key* diberi garis bawah.

### 3. Relasi atau Hubungan

Hubungan ini dinamakan relationship atau relasi. Hubungan harus dibedakan antara hubungan bentuk dengan entity dari isi hubungan ini sendiri. Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Hubungan digambarkan dengan simbol ketupat.

### 4. Garis

Digunakan untuk menghubungkan entity dengan entity maupun entity dengan atribut, atribut dinyatakan dalam bentuk garis.

## **2.15 Derajat Relasi atau Kardinalitas**

Kardinalitas merupakan jumlah maksimum tupel yang dapat berelasi dengan sejumlah entitas yang ada ditabel lain (Al-Bahra Bin Ladjamudin, 2013:147).

Menurut Al-Bahra Bin Ladjamudin Macam-macam kardinalitas adalah sebagai berikut:

1. **Satu ke satu** (*one to one*)

Setiap tupel entitas A berhubungan dengan paling banyak satu tupel entitas B, dan begitu pula sebaliknya.

2. **Satu ke banyak** (*one to many*)

Setiap tupel entitas A dapat berhubungan dengan lebih dari satu tupel entitas B tetapi tidak sebaliknya.

3. **Banyak ke satu** (*many to one*)

Relasi ini menunjukkan bahwa banyak tupel pada entitas A hanya boleh berhubungan dengan satu tupel pada entitas B.

4. **Banyak ke banyak** (*many to many*)

Setiap tupel entitas A dapat berhubungan dengan banyak tupel pada entitas B dan demikian pula sebaliknya.

## 2.16 Data Flow Diagram (DFD)

DFD adalah gambaran alur data atau informasi tanpa mengaitkan bentuk fisik media penyimpanan data atau DFD merupakan alat bantu yang berfungsi untuk menggambarkan secara rinci mengenai sistem sebagai jaringan kerja antar fungsi yang berhubungan satu sama lain dengan menunjukkan dari dan kemana data mengalir serta penyimpanannya (Kendall dan Kendall, 2003).

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 atau lingkungan fisik dimana data tersebut akan disimpan. DFD merupakan suatu grafik yang menggambarkan alur data berupa perpindahan data dari *input* menuju *output*. DFD juga digunakan

untuk menggambarkan perpindahan data dari suatu perangkat lunak (*software*) atau sistem pada tiap tingkatan (*level*).

Tabel 2.2 Simbol Data Flow Diagram (Kendall dan Kendall, 2003)

Simbol	Arti
	Menunjukkan entitas
	Menunjukkan aliran (arus data)
	Menunjukkan proses
	Menunjukkan penyimpanan data

Empat komponen utama yang menyusun suatu DFD, yaitu:

1. Kesatuan Luar (*External Entity*)

Setiap sistem pasti mempunyai batas sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan di lingkungan luar sistem dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.

2. Arus Data (*Data Flow*)

Arus data 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 dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

3. Proses (*Process*)

Proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk

dihasilkan 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.

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

Simpanan data merupakan simpanan dari data yang dapat berupa :

1. Suatu *file* atau *database* di sistem komputer
2. Suatu arsip atau catatan manual.
3. Suatu kotak tempat data di meja seseorang.
4. Suatu tabel acuan manual.
5. Suatu agenda atau buku.

Simpanan data di DFD dapat disimbolkan dengan sepasang garis horizontal paralel yang kedua ujungnya terbuka atau salah satu ujungnya tertutup.

### 2.17 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah Prosedur eksekusi sebuah program atau sistem dengan tujuan untuk menemukan kesalahan serta pengujian perangkat lunak ini juga digunakan untuk memastikan apakah sistem telah bekerja sesuai dengan spesifikasinya (Nidhra dan Dondethi, 2012).

Menurut Zohrahayati (2007) Sejumlah aturan yang berfungsi sebagai sasaran pengujian pada perangkat lunak adalah sebagai berikut:

1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan masalah.
2. *Test Case* yang baik adalah yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya.

3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya.

### **2.18 Black Box Testing**

*Black Box Testing* adalah suatu metode yang digunakan untuk menguji perangkat lunak dari segi spesifikasi fungsionalitas tanpa menguji desain dan kode program, Pengujian *black box* ini dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sudah sesuai dengan yang dibutuhkan (Shalahuddin dan Rosa, 2011).

Gries (2005) mengatakan bahwa tujuan dari pengujian *black box* ini yaitu:

1. Menemukan fungsi yang hilang atau tidak benar
2. Kesalahan *interface*
3. *Error* pada struktur data atau akses *eksternal database*,
4. *Error* pada kinerja
5. Dan batasan dari suatu data.

Keuntungan dari pengujian dengan menggunakan metode *black box* adalah:

1. Penguji tidak harus menguasai pemrograman
2. Kesalahan dari perangkat lunak ataupun *bug* sering kali ditemukan oleh kelompok penguji yang berasal dari pengguna.
3. Hasil dari pengujian menggunakan metode *black box* dapat memperjelas kerancuan yang mungkin timbul dari proses eksekusi sebuah perangkat lunak.
4. Proses pengujian lebih cepat dibandingkan dengan pengujian *white box*.