

BAB II

LANDASAN TEORI

2.1. Sistem Informasi

Sistem informasi memiliki banyak pengertian, tapi semuanya itu mengarah pada maksud yang sama. Untuk lebih lebih jelas, maka pengertian dapat dilihat dari arti masing-masing unsur pembentuknya, yang dapat dilihat dibawah ini.

a. Sistem

Sistem adalah sekumpulan hal atau kegiatan atau elemen atau subsistem yang saling bekerjasama atau yang dihubungkan dengan cara-cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai tujuan.

b. Informasi

Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimaannya dan bermanfaat dalam mengambil keputusan saat ini atau mendatang.

Dari pernyataan diatas dapat diambil kesimpulan bahwa sistem informasi (SI) dapat didefinisikan sebagai kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi (Jogiyanto, 2011).

Dengan kata lain , sistem informasi merupakan kesatuan elemn-elemen yang saling berinteraksi secara sistematis dan teratur untuk menciptakan dan

membentuk aliran informasi yang akan mendukung pembuatan keputusan dan melakukan control terhadap jalannya perusahaan.

2.2. Desain Sistem

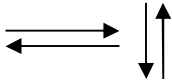
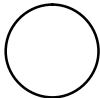
Desain sistem adalah suatu fase dimana diperlukan suatu keahlian perencanaan untuk elemen-elemen komputer yang akan menggunakan sistem baru (Kristanto, 2012). Ada 2 hal yang perlu diperhatikan dalam desain sistem yaitu pemilihan peralatan dan program komputer untuk sistem yang baru.

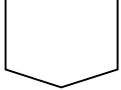

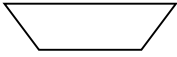
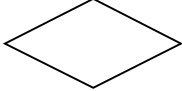
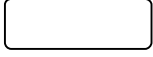
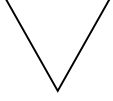

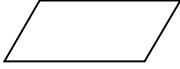
Ada beberapa alat bantu yang digunakan dalam desain sistem yaitu *Flowchart*, *Context Diagram*(CD), *Data Flow Diagram*(DFD), *Entity Relationship* (E-R), Normalisasi. Adapun desain sistem yang digunakan adalah:

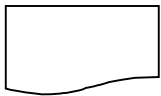
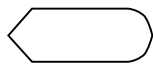
2.2.1 Diagram Alur (*Flowchart*)

Diagram Alur berfungsi untuk memodelkan masuk, keluaran, proses maupun transaksi dengan menggunakan simbol tertentu. Pembuatan diagram alur harus memudahkan bagi pemakai dalam memahami dari alur sistem atau transaksi. Adapun simbol- simbol yang sering digunakan dalam diagram alur dapat dilihat pada tabel 2.1 dan tabel 2.2 yang terlampir.

Tabel 2.1 Simbol-Simbol Bagan Alur Sistem (*Flowchart System*)

Simbol	Nama Simbol	Fungsi
	Simbol arus atau <i>flow</i>	Menyatakan jalannya arus suatu proses
	Simbol <i>connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam



		halaman yang sama.
	Simbol <i>offline connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
	Simbol <i>process</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
	Simbol <i>manual</i>	Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.
	Simbol <i>decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua jawaban : ya atau tidak.
	Simbol <i>terminal</i>	Menyatakan permulaan atau akhir suatu program.
	Simbol <i>offline-storage</i>	Menunjukkan bahwa data pada simbol ini akan disimpan ke suatu media tertentu.
	Simbol <i>manual input</i>	Memasukkan data secara manual dengan menggunakan keyboard.
	Simbol <i>input</i> atau <i>output</i>	Menyatakan proses input atau output tanpa tergantung jenis


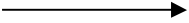
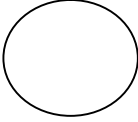
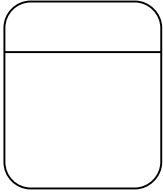
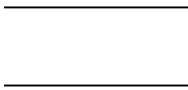
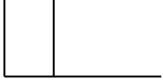
		peralatannya.
	Simbol <i>document</i>	Mencetak keluaran dalam bentuk dokumen.
	Simbol display	Mencetak keluaran dalam layar monitor.

2.2.2 Context Diagram (CD)

Diagram konteks adalah sebuah diagram sederhana yang menggambarkan hubungan antar entity luar, masuk dan keluar dari sistem. Simbol-simbol yang digunakan dalam gambaran *Context Diagram* (CD) dan *Data Flow Diagram* (DFD) dapat digambarkan sebagai berikut :

Tabel 2.2 Simbol-Simbol Gambaran *Context Diagram* dan DataDFD

Menurut DeMarco dan Yourdan	Menurut Gene dan Searson	Arti	Keterangan
		Entitas	Entitas eksternal merupakan sumber atau tujuan dari aliran data dari atau system

		Aliran data	Aliran data menggambarkan aliran data dari satu proses ke proses lainnya
		Proses	Proses atau fungsi yang mentransformasikan data secara umum
		Penyimpanan Data	Berkas atau tempat penyimpanan fungsi untuk menyimpan data atau file

2.2.3 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses apa yang dikenal pada data tersebut. Simbol-simbol yang digunakan dalam Context Diagram dan DFD dapat digambarkan pada tabel diatas.

2.2.4 Entity Relationship Diagram (ERD)

Model ERD adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas, atribut dan hubungan antar

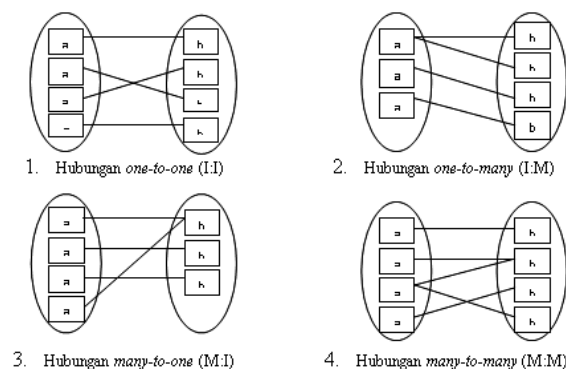
entitas. Huruf E sendiri menyatakan entitas dan R menyatakan hubungan (dari kata relationship). Model ini dinyatakan dalam bentuk diagram. Itulah sebabnya model ERD acap kali juga disebut sebagai diagram ERD. Perlu diketahui bahwa model seperti ini tidak mencerminkan bentuk fisik yang nantinya akan disimpan dalam database, melainkan hanya bersifat konseptual. Itulah sebabnya model ERD tidak bergantung pada produk DBMS yang akan digunakan.

Simbol-simbol yang digunakan dalam ERD dapat dilihat dalam gambar 2.2 yang terlampir. Jenis hubungan antar dua tipe entitas dinyatakan dengan istilah hubungan *one-to-one*, *one-to-many*, *many-to-one* dan *many-to-many*. Dengan mengasumsikan bahwa terdapat dua buah tipe entitas bernama A dan B. Penjelasan masing-masing jenis hubungan tersebut adalah seperti berikut:

1. Hubungan *one-to-one* (I:I) menyatakan bahwa setiap entitas pada tipe entitas A paling banyak berpasangan dengan satu entitas pada tipe entitas B, begitu pula sebaliknya.
2. Hubungan *one-to-many* (I:M) menyatakan bahwa setiap entitas pada tipe entitas A bisa banyak berpasangan dengan banyak entitas pada tipe entitas B, sedangkan entitas pada tipe entitas B bisa berpasangan dengan satu entitas pada tipe entitas A.
3. Hubungan *many-to-one* (M:I) menyatakan bahwa setiap entitas pada tipe entitas A paling banyak berpasangan dengan satu

entitas pada tipe entitas B, dan setiap entitas pada tipe entitas B bisa berpasangan dengan banyak entitas pada tipe entitas A.

Hubungan *many-to-many* (M:M) menyatakan bahwa setiap entitas pada tipe entitas A bisa berpasangan dengan banyak entitas pada tipe entitas B, dan begitu pula sebaliknya. Ilustrasi jenis hubungan yang digunakan dapat digambarkan sebagai berikut:



Gambar 2.1 Ilustrasi Jenis Hubungan

2.3. Data

Data dapat didefinisikan sebagai bahan keterangan tentang kejadian-kejadian nyata atau fakta yang dirumuskan dalam sekelompok lambing tertentu yang tidak acak, yang menunjukkan jumlah, tindakan atau hal. Data dapat berupa catatan-catatan dalam kertas, buku atau tersimpan sebagai file dalam basis data.

Data menjadi bahan dalam suatu proses pengolahan data. Oleh karena itu, suatu data belum dapat berbicara banyak sebelum diolah lebih lanjut. Contoh data adalah catatan identitas pegawai, catatan transaksi pembelian, catatan transaksi penjualan dan lain-lain. (Edhy Sutanta, 2011)

Dalam sebuah aplikasi yang menggunakan basis data terdapat beberapa macam jenis data, yaitu :

1. Data operasional : data dari suatu organisasi berupa data yang di simpan dalam basis data.
2. Data masukan (input data) : data dari luar Sistem yang dimasukan melalui peralatan input (misalnya: keyboard) yang dapat mengubah data operasional.
3. Data keluaran : data berupa laporan melauai peralatan output(misalnya : screen, printer dan lain-lain) sebagai hasil proses dari dalam suatu Sistem yang mengakses data operasional.

2.4. Database

Database secara bahasa memiliki arti tempat atau wadah dari sebuah data, secara istilah *database* memiliki arti yaitu kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Untuk memahaminya pertama-tama kita lihat sejumlah definisi *database* seperti yang terlihat pada berikut :

Table 2.3 Definisi *Database*

Definisi Database	Sumber
Kumpulan data yang terkait	Elmazri 1994
Kumpulan data yang terorganisasi dari data-data yang secara nalar terkait	Hoffer, 2005

Kumpulan data yang umumnya menjabarkan aktifitas-aktifitas dari satu atau lebih dari satu organisasi yang terkait	Ramakrishnan dan Gerke, 2000
---	------------------------------

Butir-butir yang bisa diambil dari ketiga definisi diatas (Kristanto, 2012)

1. Sebuah *database* menghimpun data yang terkait atau data yang saling berhubungan.
2. Kumpulan data tersebut terorganisasi.
3. Bisa melibatkan lebih dari satu organisasi.

2.5 PHP

PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memprogram situs *web* dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain.

Contoh terkenal dari aplikasi PHP adalah phpBB dan MediaWiki (*software* di belakang Wikipedia). PHP juga dapat dilihat sebagai pilihan lain dari ASP.NET/C#/VB.NET Microsoft, ColdFusion Macromedia, JSP/Java Sun Microsystems, dan CGI/Perl. Contoh aplikasi lain yang lebih kompleks berupa CMS yang dibangun menggunakan PHP adalah Mambo, Joomla!, Postnuke, Xaraya, dan lain-lain. Kelebihan PHP dari bahasa pemrograman lain :

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana – mana dari mulai IIS sampai dengan apache, dengan konfigurasi yang relatif mudah.

3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis – milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (linux, unix, windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem. (Kadir , 2013)

2.6 MySQL

MySQL adalah suatu perangkat lunak database relasi (*Relational Database Management System* atau RDBMS), seperti halnya ORACLE, Postgresql, MS SQL, dan sebagainya. MySQL AB menyebut produknya sebagai *database open source* terpopuler di dunia. Berdasarkan riset dinyatakan bahwa bahwa di *platform Web*, dan baik untuk kategori *open source* maupun umum, MySQL adalah *database* yang paling banyak dipakai. Menurut perusahaan pengembangnya, MySQL telah terpasang di sekitar 3 juta komputer. Puluhan hingga ratusan ribu situs mengandalkan MySQL bekerja siang malam memompa data bagi para pengunjungnya. Penyebab utama MySQL begitu populer di kalangan *Web* adalah karena ia memang cocok bekerja di lingkungan tersebut.

Pertama, MySQL tersedia di berbagai *platform* Linux dan berbagai varian Unix. Sesuatu yang tidak dimiliki *Access*, misalnya : padahal *Access* amat *populer* di *platform Windows*. Banyak *server Web* berbasiskan Unix, ini menjadikan

Access otomatis tidak dapat dipakai karena ia pun tidak memiliki kemampuan *client-server/networking*.

Kedua, fitur-fitur yang dimiliki MySQL memang yang biasanya banyak dibutuhkan dalam aplikasi *Web*. Misalnya, klausa LIMIT SQL-nya, praktis untuk melakukan paging. Atau jenis indeks *field* FULLTEXT, untuk *full text searching*. Atau sebutlah kekayaan fungsi-fungsi built-innya, mulai dari memformat dan memanipulasi tanggal, mengolah *string*, *regex*, *enkripsi* dan *hashing*. Yang terakhir misalnya, praktis untuk melakukan penyimpanan *password* anggota situs.

Ketiga, MySQL memiliki *overhead* koneksi yang rendah. Soal kecepatan melakukan transaksi atau kinerja di kondisi load tinggi mungkin bisa diperdebatkan dengan berbagai benchmark berbeda, tapi kalau soal yang satu ini MySQL-lah juaranya. Karakteristik ini membuat MySQL cocok bekerja dengan aplikasi CGI, di mana di setiap *request* skrip akan melakukan koneksi, mengirimkan satu atau lebih perintah SQL, lalu memutuskan koneksi lagi. Cobalah melakukan hal ini dengan Interbase atau bahkan Oracle. Maka dengan load beberapa *request* per detik saja *server Web/database* Anda mungkin akan segera menyerah karena tidak bisa mengimbangi beban ini. (Kadir, 2013)

2.7 APACHE

Apache merupakan *web server* yang digunakan untuk menampilkan *website* di *internet* seperti menggunakan Mozilla fire fox, Google Chrome, IE, Safari, dll berdasarkan kode-kode yang di tulis di dalam *website* tersebut baik menggunakan bahasa pemrograman HTML maupun PHP yang mengambil suatu *database* yang

dibangun di MySQL, sehingga terbentuklah sebuah *website* yang dapat di lihat di Mozilla fire fox dan kawan-kawannya. Apache sendiri bersifat *open source* sehingga dapat digunakan oleh siapa saja dan dikembangkan oleh siapa saja tentunya bagi yang mampu mengembagkannya. (Kadir , 2013)

2.8 Cuti

Cuti secara umum dapat diartikan tidak masuk kerja yang diizinkan dalam jangka waktu tertentu dan mematuhi semua peraturan yang berlaku di perusahaan tersebut. (Arif Setiyanto, 2015)