

BAB II

LANDASAN TEORI



2.1. Pengertian Sistem

Terdapat dua kelompok pendekatan di dalam pendefinisian sistem, yaitu yang menekankan pada prosedur dan menekankan pada komponen atau elemen. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut :

"Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu". (Jogiyanto H.M., Analisis dan Desain Sistem Informasi, 1998:1)

Pendekatan sistem yang menekankan pada komponen atau elemen-elemen mendefinisikan sebagai berikut :

"Sistem adalah elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu". (Jogiyanto H.M., Analisis dan Desain Sistem Informasi, 1998:1)

Sasaran atau tujuan dalam sistem adalah untuk menentukan operasi yang akan dilaksanakan. Berkembangnya suatu sistem dipengaruhi oleh informasi yang terdapat didalamnya. Suatu sistem jika kekurangan informasi maka lambat laun tidak dapat dipergunakan lagi.

2.1.1. Karakteristik Sistem

Menurut Jogiyanto H.M, dalam buku Analisis dan Desain Sistem Informasi, sistem mempunyai karakteristik atau sifat-sifat yang tertentu, antara lain :

1. Komponen sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk satu kesatuan.

2. Batas sistem

Batasan sistem merupakan daerah yang membatasi antara satu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Lingkungan luar sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

4. Penghubung sistem

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya.

5. Masukan sistem

Masukan (*input*) adalah energi yang dimasukkan ke dalam suatu sistem.

6. Keluaran Sistem

Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

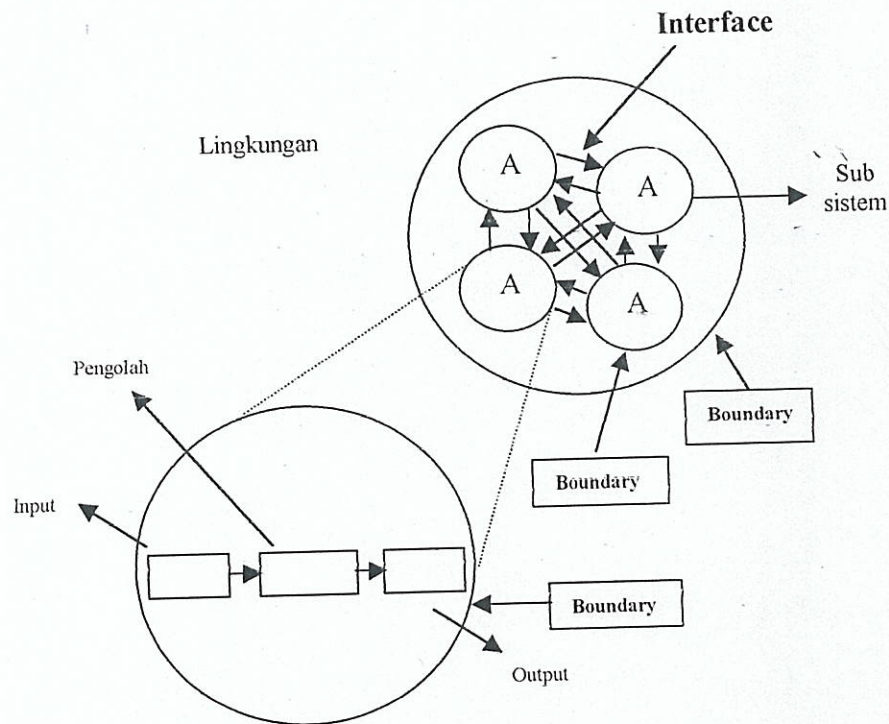
7. Pengolahan sistem

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran.

8. Sasaran sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada

gunanya. Sasaran dari sistem sangat menentukan sekali atas masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan (perbedaan tujuan dan sasaran lihat gambar 2.11)



Gambar 2.1.1. Sasaran sistem

2.2. Pengertian Informasi

Informasi sangat penting bagi perusahaan. Selain itu informasi juga penting bagi sistem, karena apabila suatu sistem tidak mempunyai informasi maka akan menjadi suatu sistem yang susah berkembang.

Informasi itu sendiri mempunyai arti "*data yang diolah menjadi bentuk yang lebih yang lebih berguna dan lebih berarti bagi yang menerimanya.*" (Jogiyanto H.M, *Analisis dan Desain Sisyem Informasi*, 1998:8).

Menurut John Burch dan Gary Grudnitski dalam buku *Analisis dan Desain Sistem Informasi* mengatakan bahwa kualitas sistem terdiri dari tiga hal, yaitu :

a. Akurat

Informasi harus bebas dari kesalahan dan tidak menyesatkan

b. Tepat pada waktunya

Informasi yang datang pada penerima tidak boleh terlambat.

c. Relevan

Informasi tersebut mempunyai manfaat untuk pemakainya.

2.3. Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat managerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan yang diperoleh. Sistem informasi adalah hasil dari sistem pengolahan data.

Data itu sendiri mempunyai arti suatu kumpulan kejadian yang diangkat berdasarkan suatu kejadian nyata, dan dapat berupa angka-angka, huruf-huruf atau simbol-simbol khusus suatu gabungan darinya.

Pengolahan data adalah manipulasi dari data kedalam bentuk yang lebih berguna dan lebih berarti. Jadi pengolahan data adalah kumpulan dari interaksi

elemen-elemen sistem yang bertanggung jawab mengumpulkan dan mengolah data dari dokumen-dokumen dasar untuk menghasilkan suatu yang lebih berguna berupa laporan, sehingga menunjang pengambilan keputusan didalam kegiatan manajemen.

2.4. Pengertian Basis Data

2.4.1. Konsep Basis Data

Menurut Kronke, dalam buku Konsep dan Tuntunan Praktis Basis Data menyebutkan basis data adalah kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan dalam perangkat keras dan menggunakan perangkat lunak untuk memanipulasinya.

Basis data merupakan dasar dalam membuat sistem informasi, karena basis data adalah salah satu komponen yang penting dalam sistem informasi. Interaksi antar basis data dalam mengadakan informasi disebut dengan sistem basis data. Sistem basis data adalah suatu sistem informasi yang mengintegrasikan komputer dari data yang saling berhubungan dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam dalam suatu organisasi.

Informasi apa saja yang akan dimasukkan dalam suatu basis data tergantung dari faktor, apakah informasi tersebut penting untuk disimpan dan dikelola.

Tujuan utama dari mempelajari basis data adalah agar kita dapat mengelola data secara optimal, baik dalam penyimpanan (*data storage*) maupun pada saat pengambilan data (*data retrieval*).

2.4.2. Pendekatan Data Base

Sebelum kita membuat basis data, terlebih dahulu dilakukan perancangan. Proses perancangan ini bersifat konseptual. Kita belum menentukan DBMS apa yang akan kita gunakan untuk mengimplementasikan rancangan basis data yang dibuat.

Tujuan perancangan basis data adalah mendapatkan skema basis data yang meminimasi terjadinya redundansi dan duplikasi data serta menjaga integrasi data. Kebanyakan metode perancangan yang ada berbasis pada model basis data relasional, struktur data diatur melalui pembuatan tabel-tabel dan terdapat keterkaitan antar tabel satu dengan lainnya (relasi)

Salah satu metode yang sering digunakan untuk merancang basis data relasional adalah Entity Relationship Diagram. Dua elemen fundamental pada ER adalah entitas dan relationship (keterhubungan).

Pendekatan basis data memperbaiki kelemahan yang terjadi pada pengolahan data secara tradisional, yaitu :

1. Duplikasi data (*data redundancy*) dapat dikurangi.

Karena data base merupakan kumpulan dari semua data secara umum, maka dapat digunakan secara bersama-sama untuk semua aplikasi, sehingga duplikasi data dapat dikurangi.

2. Hubungan data (*data relability*) dapat dikurangi.

Karena data dikumpulkan secara bersama-sama, maka hubungan data dapat ditingkatkan yang berarti data di file tertentu dapat dihubungkan dengan data di file lainnya.

2.4.3. Himpunan entitas

Adalah suatu objek (baik yang nyata maupun abstrak) didunia nyata yang dapat dibedakan dari objek lain berdasarkan karakteristik yang dimikinya, misalnya :

Pegawai suatu perusahaan, barang yang dijual disuatu toko, mahasiswa dan sebagainya.

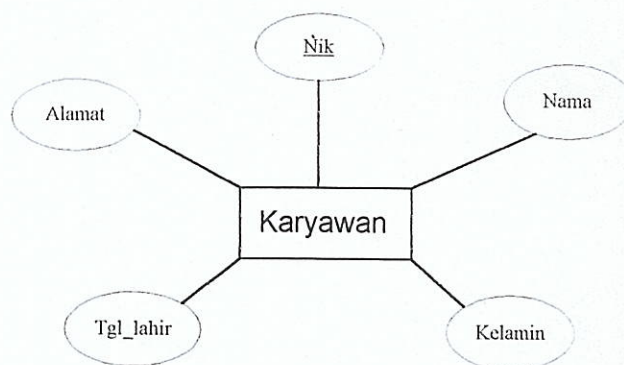
Informasi yang disimpan dari suatu entitas disebut dengan **properti/atribut**. Misalnya pada entitas Karyawan, beberapa atribut yang mungkin adalah NIK, nama mahasiswa, tanggal lahir, alamat, kelamin, dan sebagainya. Pada ER Diagram, entitas memiliki simbol persegi panjang, sedangkan atribut dilambangkan dengan elips.

Untuk membedakan antara data pada satu entitas, biasanya salah satu atribut digunakan sebagai identifier yang dapat membedakan antara data satu dengan lainnya. Atribut ini disebut dengan primary key.

Pertimbangan pemilihan atribut mana yang dijadikan sebagai primary key tergantung dari beberapa hal, antara lain :

1. Atribut harus unik
2. Nilai atribut tersebut jarang berubah.
3. Mudah diingat, dan biasanya memiliki format konsisten.

Contoh Penggambaran entitas (belum memiliki relasi) dengan atributnya. Atribut primary key memiliki garis bawah.



Gambar 2.4.3. Entitas Karyawan

Dari skema di atas, dapat dibuat tabel data :

Nik	Nama	Tgl_lahir	Kelamin	Alamat
1080999	Payrol	09/08/1979	L	Bubat 50
.....				

Untuk tiap atribut, nilai atribut memiliki batasan yang disebut dengan *domain constraint*. Misalnya, atribut NIK merupakan string dengan panjang tetap yaitu 8. Atribut nama merupakan string dengan panjang maksimal 20. Kelamin hanya boleh diisi dengan 'L' atau 'P'.

Jika suatu atribut nilainya tidak / belum terdefinisi, maka nilai atribut tersebut adalah null. Misalnya alamat seorang karyawan belum diketahui, maka diberikan nilai null pada kolom alamat.

Pada kasus lain, suatu atribut dapat mengacu/merefer ke atribut pada tabel lain (atau atribut lain pada tabel itu sendiri). Atribut yang diacu oleh atribut lain disebut dengan *foreign key*. Foreign key adalah salah satu cara untuk menjaga integritas data.

2.4.4. Entity Relationship

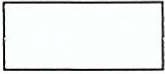
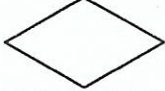
Adalah suatu cara untuk mengorganisasikan data yang memperlihatkan hubungan antar entity.

Derajat keterhubungan antar entitas pada suatu relasi disebut dengan kardinalitas. Terdapat tiga jenis kardinalitas :

1. 1 – 1, menunjukkan hubungan satu ke satu.
2. 1 – N, menunjukkan hubungan satu ke banyak.
3. N – N, menunjukkan hubungan banyak ke banyak.



Beberapa simbol yang digunakan dalam pembuatan entity relationship adalah:

Simbol	Arti
	Himpunan entitas
	Atribut sebagai key
	Himpunan relasi
	Link

Tabel 2.2.4

Basis data sangat erat kaitannya dengan normalisasi. Suatu basis data dapat dikatakan baik jika setiap tabel yang menjadi unsur pembentuk basis data tersebut telah berada dalam keadaan baik atau normal. Sesuai dengan yang dikemukakan oleh Kroenke, mendefinisikan normalisasi sebagai proses untuk mengubah suatu

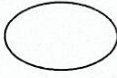
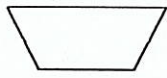
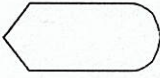
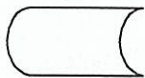
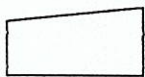
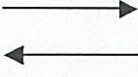
relasi yang memiliki masalah tertentu di dalam dua buah relasi atau lebih yang tidak memiliki masalah. Normalisasi sangat membantu bagi pengembangan untuk mencegah penciptaan struktur tabel yang kurang fleksibel atau mengurangi keefisienan.

Aturan-aturan normalisasi dinyatakan dalam istilah bentuk normal. Bentuk normal adalah suatu aturan yang dikenakan pada relasi-relasi dalam basis data dan harus dipenuhi oleh relasi-relasi tersebut pada level-level normalisasi.

2.4.5. Flow Map

Adalah merupakan bagian alir yang menunjukkan arus data dari laporan dan form termasuk tembusan-tembusannya pada sistem.

Beberapa simbol yang digunakan dalam pembuatan flow map adalah :

Simbol	Arti	Simbol	Arti
	Dokumen		Penghubung
	Kegiatan manual		Input/Output
	Proses komputer		Display
	Disket		Arsip
	Keyboard		Hubungan

Tabel 2.2.5

2.4.6. Diagram Konteks

Diagram konteks adalah merupakan alat-alat untuk struktur analisis. Dan pendekatan struktur ini mencoba untuk menggambarkan sistem secara garis besar atau secara keseluruhan.

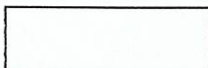
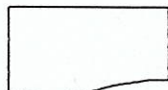
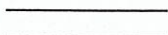
2.4.7. Data Flow Diagram (DFD)

Adalah suatu jaringan kerja dari proses dengan tempat penyimpanan data serta dihubungkan satu dengan yang lainnya, atau kumpulan simbol-simbol yang menggambarkan jalannya aliran data dari sistem.

Keuntungan menggunakan data flow diagram adalah supaya lebih memudahkan pemakai (*user*) yang kurang menguasai bidang komputer. Untuk lebih mengerti sistem yang akan dikembangkan atau dikerjakan. Proses data pada flow diagram merupakan kumpulan program data dan juga merupakan informasi data secara manual.

Dari pengertian dan definisi serta keuntungan-keuntungan yang didapat dari pengguna data flow diagram maka dapat ditarik kesimpulan bahwa data flow diagram merupakan suatu diagram yang mudah dimengerti dan merupakan suatu gambaran mengenai tata letak lokasi dan semua kegiatan-kegiatan serta aktivitas yang terjadi didalam proses aktivitas tersebut.

Beberapa simbol yang digunakan dalam pembuatan data flow diagram dan konteks diagram adalah :

Simbol	Arti
	Aliran Data
	Terninator (tujuan dan sumber aliran data)
	Proses
	Keluaran atau kumpulan file lama maupun baru
	File

Tabel 2.4.7

2.4.8. Struktur File

Adalah spesifikasi atau keterangan suatu file yang terdiri dari file-file dimana keterangan tersebut meliputi nama file, type file, lebar file, dan keterangan file. Pemberian keterangan pada suatu file sangat mempengaruhi terhadap program yang dibuat. Kesalahan pada saat pemberian nilai keterangan untuk suatu atribut menyebabkan tingkat akurasi yang dihasilkan menjadi kurang efektif.

2.5. ALAT BANTU PEMBUATAN APLIKASI (*TOOLS*)

2.5.1. Sekilas Tentang Visual Basic

Objek Oriented adalah beberapa konsep yang berbeda dengan pendekatan pemrograman yang konvensional. Konsep-konsep ini saling berhubungan erat antara yang satu dengan yang lainnya, sehingga pemahaman akan salah satu konsep menuntut pemahaman akan konsep yang lainnya (*Danie Wirajaya, Dasar-Dasar Pemrograman Berorientasi Objek, 1995:1*).

Visual Basic merupakan *tool* pembuatan aplikasi *GUI-based* (berbasis grafis), yang dibuat oleh *Microsoft Corp.* Bahasa yang digunakan oleh VB diadaptasi dari bahasa BASIC (Beginners All-Purpose Symbolic Instruction Code) dengan penambahan fungsi-fungsi baru, *control object*, prosedur dan *keyword* yang akan memudahkan pembuatan beragam aplikasi, serta kemampuan berkomunikasi dengan berbagai macam file basis data (teknologi *ActiveX Data Object*).

Salah satu program aplikasi dalam windows yang berorientasi objek (*Object Oriented*). Visual basic digunakan untuk mengembangkan dan memanfaatkan keistimewaan konsep-konsep grafis dalam windows, dan mengimplementasikan objek penghubung dengan pemakai (*User Interface Object*). Konsep dasar dari Visual basic antara lain :

1. Objek

Adalah benda yang dikenakan pekerjaan.

2. Properti

Adalah Objek yang dikerjakan.

3. Event

Adalah peristiwa yang diterima oleh objek.

Ada 3 langkah yang harus dilakukan dalam membuat aplikasi dalam VB :

1. Membuat antar muka, yaitu menentukan control *object* mana saja yang akan dipakai.
2. Mengatur properti tiap objek dan memilih event.
3. Menulis kode program untuk tiap objek. Bagian ini biasanya merupakan yang paling sulit (untuk membuat kode program, anda harus memahami sintaks-sintaks pemrograman).

Pemrograman dengan cara visual basic (*Visual Programing*) dikatakan mendesain tampilan program setengah dari pemrograman itu sendiri, maka cara ini dapat mempermudah dan mempersingkat waktu yang dibutuhkan dalam proses pemrograman. Dengan demikian programmer tinggal mencurahkan pada cara kerja program, karena soal tampilan sudah disesuaikan oleh visual basic.

Visual basic memiliki beberapa kelebihan antara lain :

1. Mempercepat waktu yang dibutuhkan dalam proses pembuatan program.
2. Menampilkan sekumpulan objek supaya tampilan program tampak menarik.
3. Tidak perlu menulis berpuluh-puluh kode untuk menampilkan sebuah tombol.
4. Mampu membuat aplikasi data base dengan meminimkan sistem pengkodean.

2.5.2. Sekilas Microsoft Access

Microsoft Access adalah perangkat lunak DBMS (*Database Management System*) relational dalam platform windows. MsAccess dilengkapi dengan fasilitas penggunaan SQL (*Sequential Query Language*).



SQL menjadi standar bahasa pengolahan database. Data MsAccess dapat dihubungkan dengan database lain seperti pada sumber data eksternal (misalnya dBase atau MsExcel) atau pada sumber data ODBC (misalnya Ms SQL Server).

Model data merupakan hasil analisis data yang dirancang dan diimplementasikan kedalam database pada saat database dibuat. Setiap unit data dimasukkan ke database dengan aturan yang ditentukan dalam model data.

Pada MsAccess, tabel berisi objek-objek kolom yang masing-masing kolomnya juga memiliki properti sendiri. Database hanya dapat berisi tabel-tabel yang di-*attach* dari database lain dan query-query yang dihubungkan untuk melakukan operasi ke tabel-tabel tersebut. Pendefinisian field/kolom dilakukan melalui Table Design View. Properti tiap field berguna untuk mengatur format pengisian data, validasi, dan pengorganisasian data.

2.5.3. Crystal Reports

Crystal report merupakan alat bantu dalam pembuatan laporan setiap transaksi. Dengan menggunakan strukture Sql maupun Ado sangat memudahkan dalam mendesain konsep laporan.

2.5.4. Structure Query Language

SQL (*Structure Query Language*) adalah bahasa standar yang digunakan untuk mengakses server database.

Semenjak tahun 1970, bahasa ini telah dikembangkan oleh IBM, yang kemudian diikuti dengan adanya Oracle, Infomix dan Sybase. Dengan menggunakan SQL, proses akses database menjadi lebih *user-friendly*, dibandingkan dengan menggunakan dbase maupun clipper yang masih menggunakan perintah-perintah pemrograman murni.

SQL tidak memiliki fasilitas pemrograman yang lengkap, tidak ada looping ataupun percabangan. Sehingga untuk menutupi kelemahan ini perlu digabung dengan bahasa pemrograman basic.

SQL dikelompokkan menjadi dua bagian yaitu :

1. DDL (Data Definition Language)

Merupakan bahasa yang digunakan untuk membuat, menghapus dan memodifikasi tabel.

Perintahnya meliputi : Create, Drop, Alter.

2. DML (Data Manipulation Language)

Merupakan bahasa yang digunakan untuk memanipulasi data seperti menambah, menghapus, mengedit dan menyeleksi data.

Perintahnya meliputi Insert, Delete, Update dan Select.

SQL Statis

Contoh Perintah sql yang menggunakan SQL Statis :

Untuk menyeleksi data : "Select * From Tpegawai"

SQL Dinamis

Yang dimaksud dengan SQL Dinamis adalah perintah SQL yang dijalankan melalui kode program dan biasanya memiliki parameter sebagai kondisi yang bisa diubah-ubah sesuai keinginan.

Cara pengaksesan menggunakan dua buah objek bertipe 'ADODB.connection' dan 'ADODB.recordset'. Dengan bantuan kedua objek tersebut, kita tidak perlu mengatur konfigurasi pada Adodc. Adodc tetap digunakan sebagai penghubung antara kedua objek tersebut dan komponen visualisasi, namun pengaturan koneksi dilakukan terhadap 'ADODB.connection' dan 'ADODB.recordset'.

- ❑ ADODB.connection , digunakan untuk membangun koneksi
 - ❑ ADODB.recordset, digunakan untuk menyimpan record/tuple hasil query.
- Nilai Adodb.recordset inilah yang akan di *passing* ke adodc.recordset untuk kemudian ditampilkan ke komponen visualisasi (misalnya DataGrid).

Ada dua cara untuk membangun koneksi dengan ADODB.

1. OLEDB Provider
2. MSDASQL.

Kode program berikut akan membuat koneksi terhadap SQL server dengan tipe koneksi menggunakan OLEDB Provider, Data source bernama gaji.mdb

Koneksi = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path & "
"gaji.mdb".

Pada penulisan kode program untuk koneksi ini, ditempatkan pada modules, dimana pada module itu sendiri dapat terdiri berbagai macam fungsi. Sehingga penulisan baik untuk koneksi ke database tidak perlu diinisialisasikan lagi.

