

2. DASAR TEORI

2.1 DESAIN SISTEM

Desain sistem dapat didefinisikan sebagai berikut.

Menurut John Burch & Gary Grudnitski :

Systems design can be defined as the drawing, planning, sketching or arranging of many separate elements into a viable, reunified a whole.

(desain sistem dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi).

2.2 DATABASE MANAGEMENT SYSTEM

Database Management System (DBMS) adalah suatu kumpulan basis data yang saling berelasi satu sama lain dan suatu bentuk program untuk mengakses basis data tersebut. DBMS terdiri dari beberapa basis data dan satu atau lebih program pengelola yang berguna pengolahan data yang ada seperti untuk menambah data, menghapus data, mengambil data dan memperbarui data-data yang ada.

Basis data sendiri merupakan sekelompok kumpulan data yang saling berelasi, yang mana dapat disimpan, diproses, dimanipulasi dan digunakan oleh pihak-pihak yang berkepentingan. Relasi tersebut biasanya ditujukan dengan bagian-bagian penting dari tiap-tiap *file* (kumpulan data) yang ada. Untuk mengenal *Database*, secara lebih men-*detail*, perlu diketahui komponen-komponen dasar pembentuk *Database*, yaitu :

1. *Entity*.

Entity adalah suatu bentuk dari orang,tempat,kejadian atau konsep dari suatu informasi yang dijadikan suatu data bagi basis data yang akan digunakan.

2. *Atribut*

Atribut adalah sebutan untuk mewakili suatu *Entity*.

3. *Data value*

Data value adalah *data actual* atau informasi yang disimpan dalam tiap-tiap data elemen atau *atribut*.

4. *Record/tuple*

Adalah kumpulan elemen-elemen yang saling berkaitan yang berguna untuk menginformasikan tentang suatu *Entity* secara lengkap.

5. *File*

Adalah sekumpulan *record-record* sejenis yang mempunyai panjang elemen sama, *atribut* yang sama namun berbeda *data value*-nya.

2.3 *Entity Relationship Diagram (ERD)*

Metode ini dipakai pada jumlah *atribut* yang banyak. Metode ini menggambarkan hubungan antar *table* yang ada dalam suatu *Database*. Pertama kali yang dilakukan adalah membagi *Database* menjadi beberapa *table* yang disebut object atau *Entity*. Masing-masing *Entity* ini mempunyai beberapa *atribut*, yang merupakan bagian dari *Entity* tersebut. Antar beberapa *Entity* tersebut dapat digambarkan suatu hubungan (*relationship*) yang biasanya dipakai dalam ERD. ERD (*Entity relationship diagram*) digunakan untuk menggambarkan hubungan antar *Entity*. Untuk menghubungkan satu *Entity* dengan *Entity* yang lain digunakan *Entity key*, yaitu suatu *atribut* tertentu atau sekelompok *atribut* tertentu yang bersifat unik, sehingga dapat digunakan untuk membedakan suatu anggota *Entity* dengan anggota yang lain pada *Entity* yang sama. Selain itu harus ada pula yang disebut *relationship key*, yaitu suatu *relationship* yang digunakan untuk menyatakan hubungan antar *Entity key*.

Macam-macam hubungan antar *Entity* :

☞ *One to one relationship*

Merupakan suatu hubungan dimana satu anggota *Entity* mempunyai hubungan dengan satu anggota *Entity* pada *Entity* yang berbeda. Ada 2 macam hubungan, yaitu *obligatory* dan *non-obligatory*. *Obligatory* adalah semua anggota dari suatu *Entity* harus berpartisipasi atau mempunyai hubungan dengan *Entity* yang lain. *Non-obligatory* adalah tidak semua anggota harus mempunyai *Entity* yang lain. Contoh satu orang mahasiswa hanya mempunyai satu nomor induk mahasiswa (nrp) dan satu nomor induk mahasiswa (nrp) hanya dimiliki oleh satu orang mahasiswa.

☞ *One to many relationship*

Merupakan suatu hubungan antara suatu anggota *Entity* yang satu dengan beberapa anggota *Entity* pada *Entity* yang berbeda. Hubungan ini juga bisa dua macam yaitu *obligatory* dan *non-obligatory*. Contoh seorang mahasiswa dapat mempunyai beberapa no telepon tetapi setiap no telp hanya dapat dipunyai oleh seorang mahasiswa.

☞ *Many to many relationship*

Merupakan hubungan antara beberapa anggota *Entity* yang satu dengan beberapa anggota *Entity* pada *Entity* yang lain. Jadi kedua belah pihak bisa mempunyai hubungan lebih dari satu dengan beberapa anggota *Entity*. Hubungan *Entity* juga bisa 2 macam yaitu *obligatory* dan *non obligatory*. Contoh satu orang mahasiswa mempunyai mata kuliah yang lebih dari satu dan mata kuliah dapat diikuti oleh lebih dari satu orang mahasiswa.

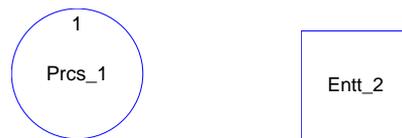
2.4 Data Flow Diagram (DFD)

Data Flow Diagram atau yang dapat disingkat dengan DFD adalah representasi dari sebuah sistem secara grafis yang digambarkan dengan sejumlah *symbol* tertentu untuk menunjukkan perpindahan data dalam proses-proses suatu sistem.

Dalam hal ini DFD menunjukkan perpindahan dan perubahan data dalam suatu sistem. Meskipun diberi nama data DFD, namun penekanan pada DFD lebih pada prosesnya, bahkan DFD merupakan salah satu alat pemodelan proses dari sistem yang paling sering digunakan.

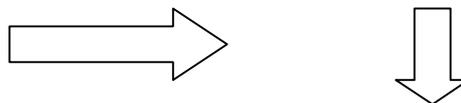
Simbol-simbol yang digunakan dalam DFD hanya terdiri dari 4 macam yaitu proses, arus data, simpanan data, dan kesatuan luar (*eksternal Entity*). Berikut uraian singkat mengenai 4 macam simbol yang ada :

- ✎ Proses adalah simbol yang mengubah suatu data dari suatu bentuk menjadi bentuk yang lain. Atau dengan kata lain, proses menerima *input* data dan mengeluarkan *output* data lain yang telah diproses. Simbol dari proses dalam DFD ada 2 macam yaitu :



Gambar 2.4.1 Gambar Proses

- ✎ Arus data atau data *flow* adalah aliran yang menunjukkan perpindahan data dari satu bagian ke bagian yang lain dalam sebuah sistem. Data *flow* dalam DFD disimbolkan dengan tanda panah dan diberi nama atau keterangan di sampingnya yang menunjukkan data apa yang mengalir. Contoh :



Gambar 2.4.2 Gambar Arus Data

- ✎ Simpanan data adalah tempat penyimpanan data dalam suatu sistem, baik secara manual maupun secara elektronik. Simpanan data digunakan jika suatu proses perlu menggunakan data tersebut lagi kemudian. Simbol dari simpanan data dalam DFD juga ada 2 macam :



Gambar 2.4.3 Gambar Simpanan Data

☞ Kesatuan luar (*eksternal Entity*) adalah seseorang, sekelompok orang, sebuah departemen di dalam maupun diluar organisasi, atau sebuah sistem yang lain yang memberikan *input* untuk sistem yang ada atau menerima *output* dari sistem yang ada. *Ekstrenal Entity* juga disebut *terminator*, karena merupakan batas dari sebuah sistem. Dalam DFD kesatuan luar disimbolkan dengan sebuah kotak persegi panjang berikut :



Gambar 2.4.4 Gambar *Eksternal Entity*

Contex diagram adalah DFD yang menunjukkan batas-batas dari sebuah sistem informasi, yaitu *top level view* dari sebuah sistem. Untuk menggambar *Contex diagram*, hanya dibuat satu proses saja yang menggambarkan keseluruhan proses dari sistem tersebut dan beberapa kesatuan luar disekelilingnya yang berhubungan. Contoh *Contex diagram* untuk DFD diatas adalah sebagai berikut :

- ? Dari DFD *level 0*, kadang-kadang ada proses yang dapat dijabarkan lebih *detail* lagi
- ? Dari DFD *level 0* dapat dibuat sejumlah DFD *level 1*, demikian pula dari setiap *level 1* tersebut jika masih dapat dijabarkan lagi menjadi DFD *level 2, 3* dan seterusnya. Sebuah DFD minimal digambarkan sampai *level 0* saja, namun dalam kenyataan sebuah DFD umumnya pasti terdiri dari beberapa *level*.

Hal hal yang perlu diperhatikan dalam penyusunan DFD :

1. Dalam DFD tidak dibedakan antara data dan informasi, semua dianggap data.
2. Nama proses dalam *Contex diagram* harus sama dengan nama

sistem tersebut.

3. Setiap DFD harus muat dalam satu halaman.
4. Setiap simbol harus diberi nama yang unik, namun harus konstan atau tidak boleh berubah jika berpindah ke *level* berikutnya.
5. Hindari garis yang berpotongan jika mungkin.
6. DFD pasti selalu mempunyai *output*.
7. Setiap proses dalam DFD harus mempunyai *input* dan *output*.
8. Suatu kesatuan luar hanya boleh mempunyai *input* atau *output*, tapi tidak boleh kedua-duanya.
9. Setiap simpanan data hanya boleh menerima *input* dari proses dan juga memberikan *output* ke proses saja.

2.5 DATABASE

Manajemen data adalah bagian dari IRM (*Information Resource Management*) dan menjamin sumber data dari perusahaan mencerminkan sistem fisik yang direpresentasikan.

Sumber data disimpan di *secondary storage* :

- *sequential access* : *magnetic tape*
- *direct access* : *magnetic disk, optical disk*

Konsep *Database* dimulai dari *index* dan *link* untuk memperoleh hubungan *logical* antara *file*.

2.6 KONSEP DATABASE

Database adalah kumpulan dari komputer data yang diintegrasikan, diorganisir dan disimpan dengan cara yang mudah untuk diakses. Untuk ini digunakan *direct access storage*. Integrasi secara *logical* dari *record* di beberapa *file* disebut konsep *Database*.

Dua tujuan utama konsep *Database* :

- meminimalkan *redundancy*
- diperoleh data *independen*

Kemampuan untuk mengubah struktur data tanpa mengubah program untuk memproses data. Hal ini dapat dicapai dengan menempatkan *data specification* pada suatu tabel yang secara fisik terpisah dari programnya. Program akan mereferensi tabel untuk mengakses data, perubahan struktur data hanya dilakukan sekali pada tabel.

2.7 SISTEM INFORMASI MANAJEMEN (SIM)

Informasi adalah salah satu jenis utama sumber daya yang tersedia bagi manajer. Informasi dapat dikelola seperti halnya sumber daya yang lain. Pengaruh dari informasi bersumber pada bisnis yang berkembang semakin kompleks, dan komputer telah mencapai kemampuan yang semakin baik.

Output informasi dari komputer digunakan oleh para manajer, nonmanajer, serta orang-orang dan organisasi-organisasi dalam lingkungan perusahaan. Manajer berada pada semua tingkat organisasional perusahaan, dan dalam semua area fungsional. Manajer melaksanakan berbagai fungsi dan peran, dan untuk berhasil, manajer perlu mengerti komputer (*computer literate*), tetapi yang lebih penting, mereka perlu mengerti informasi (*information literate*)

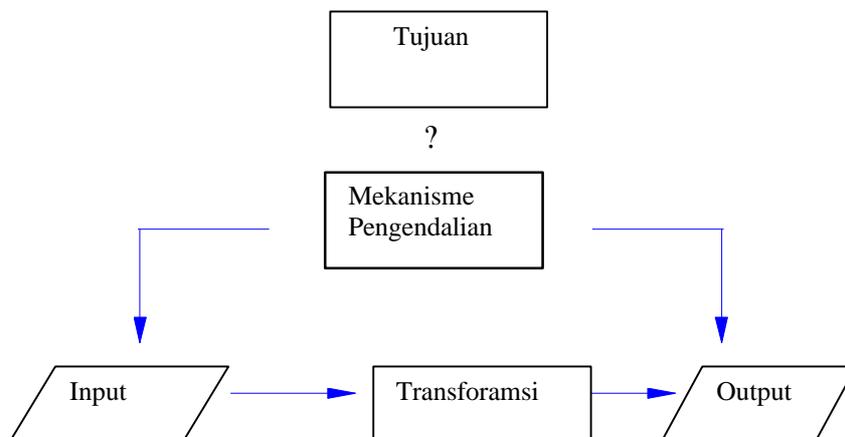
Sangat bermanfaat jika manajer mampu melihat unitnya sebagai suatu sistem yang terdiri dari beberapa subsistem dan berada dalam supersistem yang lebih besar. Perusahaan adalah suatu sistem yang bersifat fisik, namun dikelola dengan menggunakan suatu sistem konseptual. Sistem konseptual itu terdiri dari suatu pengolah informasi yang mengubah data menjadi informasi dan menggambarkan sumber daya fisik.

Awalnya, aplikasi komputer utama adalah pengolahan data akuntansi. Aplikasi tersebut lalu diikuti oleh empat aplikasi lain : sistem informasi manajemen, (*management information system*), sistem pendukung keputusan (*decision support system*), otomasi kantor (*office automation*) dan sistem pakar (*expert system*). Kelima aplikasi ini membentuk sistem informasi berbasis komputer (*computer-based information system*) atau CBIS.

Sangat sulit untuk membuktikan nilai ekonomis dari suatu aplikasi komputer, tetapi banyak analisis yang dilakukan untuk menjustifikasi tiap

proyek potensial. Setelah berjalan, proyek tersebut berkembang melalui suatu siklus kehidupan sistem (*system life cycle*). Para spesialis informasi dapat berperan serta dalam tingkat yang beragam ,tetapi keseluruhan siklus, termasuk pengembangan dan pemakaian, harus dikelola oleh seorang manajer.

Suatu sistem mempunyai susunan dasar, antara lain : *input,output*, transformasi, mekanisme pengendalian, dan tujuan. Berikut adalah gambar dari elemen-elemen sistem :



Gambar 2.7.1 Gambar Elemen Sistem

Sumber daya *input* diubah menjadi sumber daya *output*. Sumber daya mengalir dari elemen *input*, melalui elemen transformasi, kepada elemen *output*.suatu mekanisme kontrol memantau proses transformasi untuk meyakinkan bahwa sistem tersebut memenuhi tujuannya. Mekanisme kontrol ini dihubungkan pada arus sumber daya dengan memakai suatu lingkaran umpan balik (*feedback loop*) yang mendapatkan informasi dari *output* sistem dan menyediakan informasi bagi mekanisme kontrol. Mekanisme kontrol membandingkan sinyal-sinyal umpan balik dengan tujuan, dan mengarahkan sinyal pada elemen *input* jika sistem operasi memang perlu diubah.

Perusahaan berada dalam suatu lingkungan yang terdiri dari elemen-elemen yang berupa individu-individu atau organisasi-organisasi. Sumber daya mengalir antara perusahaan dan tiap elemen tersebut.

Dalam upaya mencapai keberhasilan di pasaran, para manajer sangat menyadari pengaruh dari para pelanggan dan pesaing perusahaan. Mereka berusaha memperoleh keunggulan kompetitif dengan mengelola arus informasi. Sejumlah usaha awal berpusatkan pada arus informasi ke dan dari pelanggan-pelanggan perusahaan. Pandangan yang lebih luas mencakup elemen-elemen lingkungan lain, seperti para pemasok. Hubungan elektronik antara komputer perusahaan dengan komputer milik elemen-elemen lingkungan memungkinkan semua organisasi tersebut berfungsi sebagai suatu sistem informasi antar organisasi.

Sumber daya informasi perusahaan mencakup lebih dari sekedar informasi. Sumber daya tersebut mencakup pula perangkat keras, fasilitas, perangkat lunak, data, para spesialis informasi dan para pemakai informasi.

2.8 TEKNIK PERANCANGAN STRUKTUR DATA

setelah basis data selesai didesain, baik memakai metode dekomposisi dan metode hubungan *Entity*, maka langkah selanjutnya adalah menentukan cara penyampaian data pada *file-file* tersebut dapat dilakukan. Pada setiap *file* akan terdapat beberapa *atribut* atau *field*. Data-data yang akan disimpan pada masing-masing *field* ini akan berbeda satu sama lainnya sesuai dengan struktur dan jenis data yang akan disimpan. Perancangan struktur data yang harus dilakukan adalah dengan menentukan jenis atau karakteristik data yang akan disimpan dan digunakan di *field* tersebut, juga berapa panjang data tersebut, misalnya apakah berupa data karakter, data numeric atau berupa data logika. Setelah ditentukan jenis data kemudian ditentukan panjang jenis data tersebut yang diperlukan untuk tiap-tiap *field* tersebut atau dengan kata lain berapa karakter yang akan disimpan di *field* tersebut. Penentuan tempat penyimpanan ini harus berdasarkan jumlah maksimum yang akan dicapai untuk pengisian data pada *field* tersebut, sehingga tidak akan ditemukan kesalahan pada system karena tempat penyimpanan yang kurang besar.

Dalam menentukan panjang karakter suatu *field* tidak boleh dilakukan secara sembarangan dengan memberikan besaran yang maksimum

karena jika dilakukan maka suatu *field* yang mestinya hanya terisi sebanyak 5 karakter tetapi didefinisikan dengan 10 karakter maka untuk jangka pendek hal tersebut akan kecil pengaruhnya, tetapi untuk jangka panjang hal tersebut akan menimbulkan masalah serius dengan dibutuhkannya tempat penyimpanan yang besar. Karena itu harus ditentukan jumlah yang paling efisien dan paling baik untuk tiap-tiap *field*.

Proses perancangan struktur data ini harus dilakukan untuk seluruh *atribut* atau *field* dari *file* yang ada, sehingga setelah itu *file-file* tersebut akan siap digunakan untuk system, karena sudah mengandung informasi yang benar.

2.9 TEKNIK DAN DESAIN MENU *INPUT-OUTPUT*

Setelah basis data selesai dirancang, lengkap dengan struktur datanya, maka tugas selanjutnya adalah merancang system *input/output* dari basis data tersebut. System *input/output* diperlukan untuk membentuk suatu system yang lengkap, dimana suatu program aplikasi akan menggunakannya guna proses pemasukan data (*input data*), pengolahan data dan menampilkan hasil (*output data*) yang diinginkan. Rancangan basis data, struktur data dan system *input/output* tersebut akan menjadi dasar untuk pembuatan program aplikasi, dimana maksud dan tujuan awal dari pembuatan *system* telah ditetapkan dari awal.

Perancangan system menu ini berkaitan dengan kemudahan pemakaian program bagi operator. Menu yang dibuat sebaiknya dirancang sedemikian rupa sehingga mudah untuk dipelajari dan mudah digunakan oleh operator yang mempunyai berbagai macam tingkat pendidikan, sehingga hanya dengan waktu yang singkat, pengoperasian program dapat dilakukan sepenuhnya. Selain itu menu-menu yang dibuat harus dapat mencakup seluruh bagian program yang akan dibuat.

Dalam pembuatan menu ini, program akan dibagi bagi ke dalam kelompok menu-menu tertentu sesuai dengan kegunaannya, sehingga nantinya akan mempermudah pemakai dalam mencari fungsi-fungsi dari

program yang diinginkan. Menu-menu ini dapat dibagi kedalam beberapa tingkatan, misalnya mulai dari menu utama, kemudian dari menu utama dapat dipilih beberapa menu pilihan lagi dan seterusnya.

Perancangan selanjutnya adalah merancang system *input/output*. Untuk merancang system *input/output* ini selain berkaitan dengan perangkat lunak atau *software*, maka akan berkaitan juga dengan perangkat keras atau *hardware* yang tersedia, yaitu peralatan *input/output* yang akan dipakai. Pada tahap ini harus direncanakan bentuk masukan apa yang diinginkan untuk diolah oleh program dan disimpan di dalam basis data dan bagaimana cara data masukan tersebut akan dimasukkan ke dalam sistem. Selain itu juga perlu direncanakan dan dirancang juga mengenai bentuk laporan atau *output* dari program, baik bentuk keluaran tersebut misalnya jenis printer serta bentuk laporan yang akan ditampilkan.