BAB II

LANDASAN TEORI

2.1 Pengertian Dasar Sistem

Mempelajari suatu sistem akan lebih mengenal bila kita mengetahui terlebih dahulu mengenai sistem. Terdapat beberapa definisi sistem antara lain :

“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”. [4]

“Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”. [4]

“Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi/tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses/pekerjaan tertentu”. [2]

“Sistem adalah kumpulan/*group* dari sub sistem/bagian/komponen apapun baik phisik ataupun non phisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu”. [1] . Berdasarkan definisi di atas, maksud dari suatu sistem adalah untuk mencapai suatu tujuan *(goal)* dan untuk mencapai suatu sasaran *(objectives).*

2.2 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu :

a. Komponen Sistem *(components)*

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian dari sistem. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut *supra system*.

b. Batas Sistem *(Boundary)*

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan Luar Sistem *(environment)*

Lingkungan luar dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem yang bersifat menguntungkan merupakan energi dari sistem sehingga harus tetap dijaga dan dipelihara, sedangkan lingkungan luar sistem yang bersifat merugikan harus ditahan dan dikendalikan, agar tidak mengganggu kelangsungan hidup dari sistem.

d. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan media penghubung yang memungkinkan sumber - sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*) untuk subsistem yang lainnya dengan melalui penghubung. Dengan penghubung satu

subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

e. Masukan Sistem (*Input*)

Masukan (*Input*) merupakan energi yang dimasukkan ke dalam sistem, dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

f. Keluaran Sistem (*Output*)

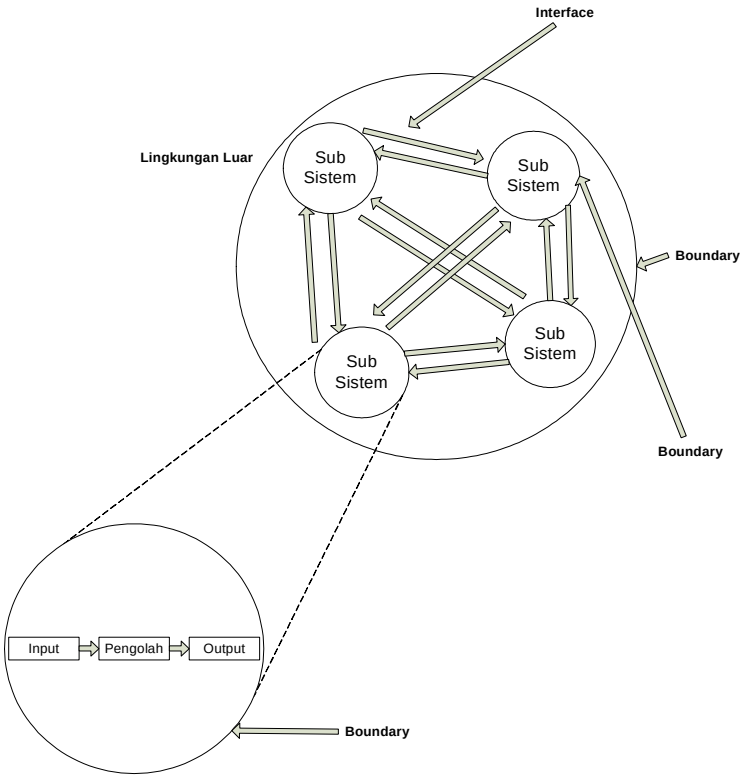
Keluaran (*Output*) merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau pada supra sistem.

g. Pengolah Sistem (*process*)

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

h. Sasaran Sistem (*Objectives*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan *system* dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.



Gambar 2.1 Karakteristik Sistem

Jogianto HM, (2005 : 6)

2.3 Klasifikasi Sistem

Sistem dapat diklasifikasikan dalam beberapa sudut pandangan, diantaranya sebagai berikut:

1. Sistem diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*).
2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*).
3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem yang tak tentu (*propabilistic system*).

4. sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*).

2.4 Kedudukan SPK dalam Sistem Informasi

Decision Support System (DSS) adalah kelas dari sistem informasi terkomputerisasi yang mendukung aktivitas pengambilan keputusan. DSS adalah interaktif berbasis komputer sistem dan subsistem dimaksudkan untuk membantu pengambil keputusan menggunakan teknologi komunikasi, data, dokumen, pengetahuan dan / atau model proses keputusan untuk menyelesaikan tugas. *Decision Support System* telah berevolusi selama 25 tahun terakhir dari sistem mainframe tidak fleksibel, untuk PC terisolasi alat, untuk *client / server* data dippers, dan sekarang untuk kinerja tinggi dan perusahaan *extensible* dukungan keputusan-aplikasi, sering melibatkan organisasi intranet. Pada saat yang sama, hubungan antara departemen TI dan pengguna telah berkembang dari badai ke koperasi.

Payung besar sistem pendukung keputusan (DSS) telah lama menyediakan tempat pengumpulan selamat datang bagi anda yang ingin membangun aplikasi perangkat lunak yang didasarkan pada model campuran, analisis data, dan antarmuka kuat. DSS menarik praktisi, akademisi dan mahasiswa dari berbagai bidang termasuk sistem informasi, riset operasi / ilmu manajemen, ilmu komputer, psikologi dan disiplin bisnis lainnya.

2.5 Konsep Dasar Sistem Pendukung Keputusan

Sitem Pendukung Keputusan/*Decision Support Systems (DSS)* adalah sebuah sistem yang dimaksudkan untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan semiterstruktur. DSS dimaksudkan untuk menjadi alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas mereka, namun tidak untuk menggantikan penilaian mereka. DSS ditujukan untuk keputusan-keputusan yang memerlukan penilaian atau pada keputusan-keputusan yang sama sekali tidak dapat didukung oleh algoritma. Aplikasi DSS dapat terdiri dari beberapa subsistem, yaitu: subsistem manajemen data, subsistem manajemen model dan subsistem antarmuka pengguna. Selain itu DSS juga bisa memiliki subsistem manajemen basis pengetahuan yang mendukung subsistem-subsistem lainnya.

2.6 *Profile Matching*

Profile matching merupakan suatu proses yang sangat penting dalam manajemen SDM dimana terlebih dahulu ditentukan kompetensi (kemampuan) yang diperlukan oleh suatu jabatan. Kompetensi/kemampuan tersebut haruslah dapat dipenuhi oleh pemegang/calon pemegang jabatan.

Dalam proses *profile matching* secara garis besar merupakan proses membandingkan antara kompetensi individu kedalam kompetensi jabatan sehingga dapat diketahui perbedaan kompetensinya (disebut juga *gap*), semakin kecil *gap* yang dihasilkan maka bobot nilainya semakin besar yang berarti memiliki peluang lebih besar untuk karyawan menempati posisi tersebut.

2.6.1 Analisis Penyelesaian Kenaikan Jabatan dengan Metode *Profile Matching*

Untuk menganalisis karyawan yang sesuai dengan jabatan tertentu dilakukan dengan metode *profile matching,* dimana dalam proses ini terlebih dahulu kita menentukan kompetensi (kemampuan) yang diperlukan oleh suatu jabatan. Dalam proses *profile matching* secara garis besar merupakan proses membandingkan antara kompetensi individu ke dalam kompetensi jabatan sehingga dapat diketahui perbedaan kompetensinya (disebut juga *gap*).

Contoh Kasus :

Sampai bulan November 2009 ini, jabatan *Manager* divisi sales pada PT. Sysmex unit Cabang Bandung masih kosong, dikarenakan karyawan yang menempati jabatan tersebut telah mengundurkan diri (*resign*). Untuk mengisi jabatan yang kosong maka dilakukan pemilihan karyawan yang sesuai dengan jabatan tersebut.

Persyaratan wajib karyawan untuk menjadi kandidat yang akan diajukan jabatannya untuk menjadi *Manager* pada PT. Sysmex (*Divisi Sales*) adalah dimana hasil *point* kerja harus memenuhi syarat wajib, dan syarat tersebut adalah hasil point harus di atas 3, point tersebut dapat dilihat melalui hasil penjualan para *sales* terhadap para *customer*, dan untuk lebih jelasnya kriteria *point* dapat dijelaskan sebagai berikut :

1 : hasil penjualan sangat tidak memuaskan
2 : hasil penjualan tidak memuaskan
3 : hasil penjualan cukup memuaskan
4 : hasil penjualan memuaskan
5 : hasil penjualan sangat memuaskan

1. Perhitungan Pemetaan GAP Kompetensi

Setelah proses pemilihan kandidat, proses berikutnya adalah menentukan kandidat mana yang paling cocok menduduki jabatan yang diajukan oleh perusahaan. Dalam kasus ini penulis menggunakan perhitungan pemetaan gap kompetensi dimana yang dimaksud dengan *gap* disini adalah beda antara profil jabatan dengan profil karyawan atau dapat ditunjukkan pada rumus di bawah ini:

1) *Gap = Profil Karyawan – Profile Jabatan*

2. Perhitungan Pemetaan GAP Kompetensi Berdasarkan Aspek-Aspek

Untuk perhitungan pemilihan karyawan pengumpulan *gap-gap* yang terjadi itu sendiri pada tiap aspeknya mempunyai perhitungan yang berbeda-beda. Untuk keterangannya bisa dilihat pada tabel 2.1 :

Tabel 2.1 Keterangan Sub Aspek Kriteria

Kriteria	Keterangan Sub Kriteria
Aspek Kapasitas Intelektual	CS : *Common Sense* VI : *Verbalisasi Ide* SB : *Sistematika Berfikir* PSR : *Penalaran dan Solusi Real* KN : *Konsentrasi* LP : *Logika Praktis* FB : *Fleksibilitas Berfikir* IK : *Imajinasi Kreatif* ANT : *Antisipasi* IQ : *Potensi Kecerdasan*

Aspek Sikap Kerja	EP : *Energi Psikis* KTJ : *Ketelitian dan Tanggung Jawab* KH : *Kehati-hatian* PP : *Pengendalian Perasaan* DB : *Dorongan Berprestasi* VP : *Vitalitas dan Perencana*
Aspek Perilaku	D : *Dominance (Kekuasaan)* I : *Influences (Pengaruh)* S : *Steadiness (Keteguhan Hati)* C : *Compliance (Pemenuhan)*

Dimana nilai aspek sub kriterianya adalah sebagai berikut :

Nilai Sub Kreteria	1 : Tidak Memenui Syarat 2 : Kurang 3 : Cukup 4 : Baik 5 : Sangat Baik

Setelah didapatkan tiap *gap* masing-masing karyawan maka tiap profil karyawan diberi bobot nilai dengan patokan tabel bobot nilai *gap* seperti yang dapat dilihat pada tabel 2.2 :

Tabel 2.2 Keterangan Bobot nilai *gap*

No	Selisih	Bobot Nilai	Keterangan
1	0	5	Tidak ada selisih (kompetensi sesuai dengan yang dibutuhkan
2	1	4,5	Kompetensi individu kelebihan 1 tingkat/level
3	-1	4	Kompetensi individu kekurangan 1 tingkat/level
4	2	3,5	Kompetensi individu kelebihan 2 tingkat/level
5	-2	3	Kompetensi individu kekurangan 2 tingkat/level
6	3	2,5	Kompetensi individu kelebihan 3 tingkat/level
7	-3	2	Kompetensi individu kekurangan 3 tingkat/level
8	4	1,5	Kompetensi individu kelebihan 4 tingkat/level
9	-4	1	Kompetensi individu kekurangan 4 tingkat / level

Tiap karyawan akan memiliki tabel bobot nilai seperti contoh-contoh tabel yang ada di bawah ini. Dengan contoh tabel bobot nilai karyawan dan dengan acuan pada tabel bobot nilai *gap*.

3. Perhitungan dan Pengelompokan *Core* dan *Secondary Factor*

Setelah menentukan bobot nilai *gap* untuk ketiga aspek yaitu aspek kapasitas intelektual, sikap kerja dan perilaku dengan cara yang sama. Kemudian tiap aspek dikelompokkan menjadi 2 (dua) kelompok yaitu kelompok *Core Factor* dan *Secondary Factor*. Untuk perhitungan core factor dapat ditunjukkan pada rumus di bawah ini:

2) NCF = $\frac{\Sigma \text{ NC (I, s, p)}}{\Sigma \text{ IC}}$

Keterangan:

NCF : Nilai rata-rata *core factor*

NC(*i, s, p*) : Jumlah total nilai *core factor (Intelektual, Sikap kerja, Perilaku)*

IC : Jumlah *item core factor*

Sedangkan untuk perhitungan *secondary factor* dapat ditunjukkan pada rumus di bawah ini:

3) NCS = $\frac{\Sigma \text{ NS (I, s, p)}}{\Sigma \text{ IS}}$

Keterangan:

NSF : Nilai rata-rata *secondary factor*

NS(*i, s, p*) : Jumlah total nilai *secondary factor (Intelektual, Sikap kerja, Perilaku)*

IS : Jumlah *item secondary factor*

Untuk lebih jelasnya pengelompokan bobot nilai *gap* dapat dilihat pada contoh perhitungan aspek kapasitas intelektual, sikap kerja dan perilaku sebagai berikut

4. Perhitungan Nilai Total

Dari hasil perhitungan dari tiap aspek di atas kemudian dihitung nilai total berdasar presentasi dari *core* dan *secondary* yang diperkirakan berpengaruh terhadap kinerja tiap-tiap profil. Contoh perhitungan dapat dilihat pada rumus di bawah ini:

4) *N* (*i, s, p*) = (*x*)%*NCF*(*i, s, p*) (*x*)%*NSF*(*i, s, p*)

Keterangan:

(i,s,p) : *(Intelektual, Sikap Kerja, Perilaku)*

N(i,s,p) : Nilai total dari aspek

NCF(i,s,p) : Nilai rata-rata *core factor*

NSF(i,s,p) : Nilai rata-rata *secondary factor*

(x)% : Nilai persen yang di*input*kan

Untuk lebih jelasnya penghitungan nilai total terlebih dahulu menentukan nilai persen yang di*input*kan yaitu *core factor* 60% dan *secondary factor* 40%. Kemudian nilai *core factor* dan *secondary factor* ini dijumlahkan sesuai rumus dan hasilnya dapat dilihat pada contoh perhitungan aspek kapasitas intelektual, aspek kerja dan aspek perilaku.

5. Perhitungan Penentuan Hasil Akhir/Ranking

Hasil akhir dari proses ini adalah ranking dari kandidat yang diajukan untuk mengisi suatu jabatan tertentu. Penentuan ranking mengacu pada hasil perhitungan tertentu. Perhitungan tersebut dapat ditunjukkan pada rumus di bawah ini:

5) $\boldsymbol{Ha = (x)\%Ni + (x)\%Ns + (x)\%Np}$

Keterangan:

Ha : Hasil Akhir

Ni : Nilai Kapasitas Intelektual

Ns : Nilai Sikap Kerja

Np : Nilai Perilaku

(x)% : Nilai Persen yang di*input*kan

2.7 Konsep Basis Data

Basis data terdiri atas dua kata , yaitu basis dan data. Basis dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul, sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Basis data merupakan kumpulan data yang terorganisasi untuk melayani berbagai aplikasi pada saat bersamaan dengan melakukan penyimpanan dan pengelolaan data sehingga data tersebut terlihat di satu lokasi. Prinsip utama basis data adalah pengaturan data atau arsip dengan tujuan utama untuk kemudahan dan

kecepatan dalam pengambilan kembali data atau arsip. Dapat disimpulkan bahwa basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya.

Basis data (*database*) merupakan salah satu komponen yang penting di sistem informasi, karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Sistem basis data (*database system*) ini adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi.

2.7.1 DBMS (*Database Management System*)

Sistem manajemen *database* atau *database management system* (DBMS) adalah merupakan suatu sistem *software* yang memungkinkan seorang *user* dapat mendefinisikan, membuat, dan memelihara serta menyediakan akses terkontrol terhadap data. *Database* sendiri adalah sekumpulan data yang berhubungan dengan secara logika dan memiliki beberapa arti yang saling berpautan. DBMS yang utuh biasanya terdiri dari :

1. *Hardware*

   Hardware merupakan sistem komputer aktual yang digunakan untuk menyimpan dan mengakses *database*. Dalam sebuah organisasi berskala besar, *hardware* terdiri dari jaringan dengan sebuah *server* pusat dan beberapa program *client* yang berjalan di komputer desktop.

2. *Software* beserta *utility software* adalah DBMS yang aktual. DBMS memungkinkan para *user* untuk berkomunikasi dengan *database*. Dengan kata lain DBMS merupakan mediator antara *database* dengan *user*. Sebuah *database* harus memuat seluruh data yang diperlukan oleh sebuah organisasi.
3. Prosedur

   Bagian integral dari setiap sistem adalah sekumpulan prosedur yang mengontrol jalannya sistem, yaitu praktik-praktik nyata yang harus diikuti *user* untuk mendapatkan, memasukkan, menjaga, dan mengambil data.
4. Data

   Data adalah jantung dari DBMS. Ada dua jenis data. Pertama, adalah kumpulan informasi yang diperlukan oleh suatu organisasi. Jenis data kedua adalah metadata, yaitu informasi mengenai *database*.
5. *User* (Pengguna)

   Ada sejumlah *user* yang dapat mengakses atau mengambil data sesuai dengan kebutuhan penggunaan aplikasi-aplikasi dan *interface* yang disediakan oleh DBMS, antara lain adalah

   a. *Database* administrator adalah orang atau group yang bertanggungjawab mengimplementasikan sistem *database* di dalam suatu organisasi
   b. *Enduser* adalah orang yang berada di depan *workstation* dan berinteraksi secara langsung dengan sistem.

c. *Programmer* aplikasi, orang yang berinteraksi dengan *database* melalui cara yang berbeda.

2.7.2 ERD (*Entity Relationship Diagram*)

Setelah semua memory esensial ditransformasikan menjadi *object data store,* maka dapat disusun relasi antara satu *object data store* dengan yang lain. *Enttity Relationship* (E-R) *Diagram* dibuat untuk memperlihatkan relasi itu. *Enttity Relationship Diagram* adalah ilustrasi grafis objek-objek (atau *entities*) dan atribut, serta relasi antara keduanya. Diagram E-R diperkenalkan oleh Peter Chen dan sifatnya independen terhadap teknologi *database* yang digunakan.

Salah satu metode pembuatan diagram E-R, yaitu dengan menggambarkan objek-objek sebagai segiempat, relasi sebagai belah ketupat kecil, dan atribut sebagai lingkaran. Notasi yang lain adalah notasi.

E-R Diagram merupakan suatu bahasa pemodelan yang dimana posisinya dapat dianalogikan dengan *story board* dalam industri film, *blu print* arsitektur suatu bangunan, miniatur, dan lain-lain. Dalam praktiknya, membangun suatu sistem terlebih dahulu dilakukannya suatu perencaan. Pemodelan merupakan suatu sub bagian dari perencanaan secara keseluruhan sebagai salah satu upaya *feedback* evaluasi perampungan suatu perencanaan. E-R Diagram sebagai suatu pemodelan setidaknya memiliki beberapa karakteristik dan manfaat sebagai berikut:

a. Memudahkan untuk dilakukannya analisis dan perubahan sistem sejak dini, bersifat murah dan cepat

b. Memberikan gambaran umum akan sistem yang akan di buat sehingga memudahkan developer.
c. Menghasilkan dokumentasi yang baik untuk *client* sebagai bahan diskusi dengan bentuk E-R Diagram itu sendiri, dan
d. Kamus data bagi bagi para pengembang *database*.

Struktur dari E-R Diagram secara umum ialah terdiri dari:

a. Entitas merupakan objek utama yang informasi akan disimpan, biasanya berupa kata benda. Contoh: mahasiswa, dosen, nasabah, mata kuliah, ruangan, dan lain-lain. Objek dapat berupa benda nyata maupun abstrak.
b. Atribut merupakan deskripsi dari objek yang bersangkutan
c. *Relationship* merupakan suatu hubungan yang terjalin antara dua entitas yang ada.

Kardinalitas Relasi ERD yang mempersentasikan suatu basisdata relasional senantiasa memiliki relasi-relasi dari sejumlah entitas yang dapat ditentukan banyaknya. Banyaknya suatu relasi yang dimiliki oleh suatu relasi entitas disebut derajat relasi. Derjat relasi maksimum disebut dengan kardinalitas sedangkan derajad minimum disebut dengan modalitas. Kardinalitas yang terjadi diantara dua himpunan entitas (misal A dan B) dapat berupa :

a. *One to One* (1-1), relasi yang terjadi jika sebuah *entry* dalam sebuah *object data sore* dihubungkan dengan hanya sebuah *entry* dalam *object data store* yang lain.

b. *One to Many* (1-M), relasi yang terjadi jika sebuah *entry* dalam sebuah *object data store* dihubungkan dengan satu atau lebih *entry* dalam *object data store* yang lain.

c. *Many to Many* (M-M), relasi yang terjadi jika satu atau lebih *entry* dalam sebuah *object data store* dihubungkan dengan satu atau lebih *entry* dalam *object data store.*

Dalam semua definisi relasi di atas, "1" menyatakan hanya satu, dan "M" menyatakan satu atau lebih. Jika suatu saat dibutuhkan pernyataan relasi paling sedikit sebuah dan juga lebih dari satu, maka relasi itu dapat dinyatakan dengan "M*" (* = *minimum of one*).

2.7.3 DFD (*Data Flow Diagram*)

Data Flow Diagram (DFD) adalah teknik grafis yang menggambarkan aliran informasi dan perubahan yang digunakan sebagai perpindahan data dari masukan ke keluaran. Elemen dasar dari *data flow diagram* adalah :

a. Entitas Luar (*External Entity*)

Sesuatu yang berada diluar sistem, tetapi ia memberikan data kedalam sistem atau memberikan data dari sistem, disimbolkan dengan suatu kotak notasi. *External Entity* tidak termasuk bagian dari sistem. Bila sistem informasi dirancang untuk satu bagian maka bagian lain yang masih terkait menjadi *external entity*.

b. Arus Data (*Data Flow*)

Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data

ditunjukan dengan arah panah dan garis diberi nama atas arus data yang mengalir. Arus data ini mengalir diantara proses, *data store* dan menunjukan arus data dari data yang berupa masukan untuk sistem atau hasil proses sistem.

1. Proses (*Process*)

   Proses merupakan apa yang dikerjakan oleh sistem. Proses dapat mengolah data atau aliran data masuk menjadi aliran data keluar. Proses berfungsi mentransformasikan satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Setiap proses memiliki satu atau beberapa masukan serta menghasilkan satu atau beberapa data keluaran. Proses sering juga disebut *bubble*.

2. Simpanan Data (*Data Store*)

   Simpanan data merupakan tempat penyimpanan data yang ada dalam sistem. Data store dapat disimbolkan dengan dua garis sejajar atau dua garis dengan salah satu sisi samping terbuka. Proses dapat mengambil data dari atau memberikan data ke simpanan data (*database*).

3. Kamus Data

   Kamus data berfungsi untuk membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

2.7.4 Kamus Data

Kamus data merupakan sebuah daftar yang terorganisasi dari elemen data yang berhubungan dengan sistem, dengan definisi yang teliti sehingga pemakai dan analisis sistem akan memiliki pemahaman yang umum mengenai *input*, *output*, komponen penyimpanan serta kalkulasi *intermediate*.

Pendefinisian data tersebut dilakukan dengan menggunakan notasi yang umum digunakan dalam menganalisis sistem yaitu dengan menggunakan sejumlah simbol. Kamus data biasanya dipelihara secara otomatis oleh sistem manajemen *database*. Cara mendefinisikan kamus data yaitu :

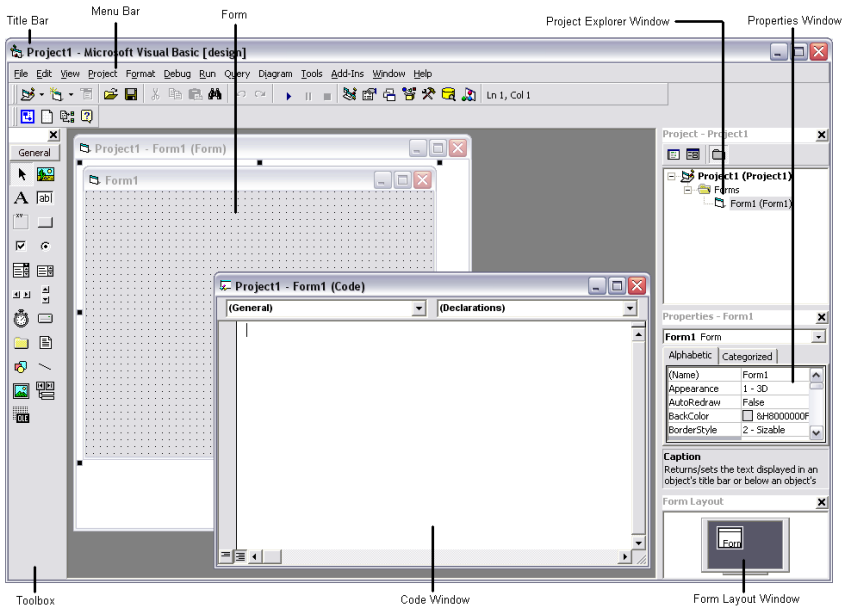
1. Menggambarkan arti aliran data atau penyimpanan yang ditunjukan dalam DFD.
2. Menggabungkan komponen dari kumpulan data yang mengalir yaitu kumpulan komponen yang mungkin bisa dipecah lagi menjadi data elementer.
3. Menggambarkan data yang tersimpan.
4. Menentukan nilai dibagian elementer dari informasi yang relevan di DFD dan *data store*-nya.

2.8 Tinjauan Perangkat Lunak

Dalam pembuatan program aplikasi ini, penulis menggunakan beberapa perangkat lunak.

2.8.1 Visual Basic 6.0

Visual Basic 6.0 merupakan salah satu aplikasi pemrograman visual yang dikeluarkan oleh Microsoft. Visual Basic 6.0 menggunakan pendekatan Graphical User Interface (GUI) yang lebih nyaman dan lebih mudah digunakan oleh programmer untuk membuat aplikasi, baik aplikasi kecil untuk keperluan sendiri ataupun sistem enterprise yang besar serta aplikasi terdistribusi yang dijalankan melalui Internet. Visual Basic 6.0 memiliki antarmuka yang dapat diatur sesuai penggunanya.



Gambar 2.2 Tampilan Visual Basic 6.0

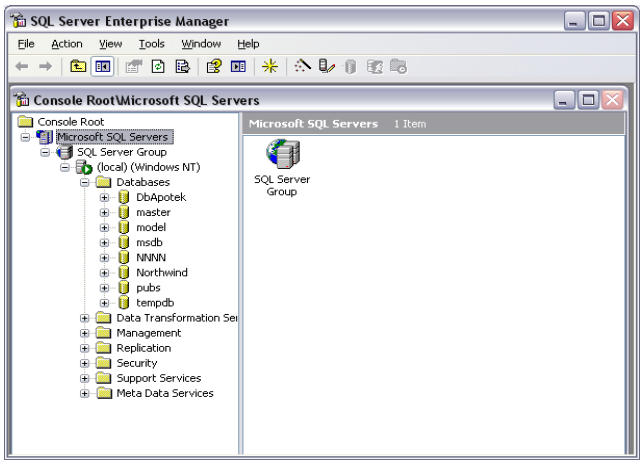
a. Menu Bar berisi menu-menu utama yang dimiliki Visual Basic 6.0, misalnya menu File, Edit, View, Project, Run, dan lain-lain. Pada masing-masing menu terdapat beberapa submenu yang berbeda.
b. Main ToolBar adalah toolbar utama yang yang berisi icon-icon yang dapat diklik oleh pengguna untuk melakukan suatu perintah khusus secara cepat.
c. Form adalah tempat untuk merancang tampilan program aplikasi yang sedang dibuat.
d. Project Explorer Window adalah jendela untuk menampilkan project-project, form-form, atau modul-modul yang terlibat dalam proses pembuatan program aplikasi.
e. Properties Window adalah jendela untuk melihat dan mengubah properti-properti yang dimiliki oleh sebuah objek.
f. Toolbox terdiri atas beberapa class objek yang dapat digunakan untuk menanamkan objek ke dalam proses pembuatan aplikasi.
g. Code Window adalah jendela yang digunakan untuk melihat atau untuk mengetikkan kode program. Setiap form memiliki satu buah Code Window sendiri.
h. Form Layout Window adalah jendela yang menunjukkan posisi relatif form terhadap layar monitor saat form tersebut ditampilkan di layar monitor.

2.8.2 SQL Server 2000

SQL Server 2000 merupakan salah satu produk RDBMS (*Relational Database Management System*) yang paling tangguh dan paling dominan yang digunakan oleh para *programmer* di lingkungan kerja berbasis windows untuk menangani data-data yang besar. SQL Server 2000 menawarkan beberapa fitur di dalam mengelola database, ada 2 fitur yang biasa digunakan untuk mengelola database di dalam SQL Server 2000, yaitu :

1. Menggunakan Enterprise Manager

   Fitur ini relatif mudah digunakan karena mode pengelolaannya berbasis GUI (Graphical User Interface) cukup dengan metode click dan drag.

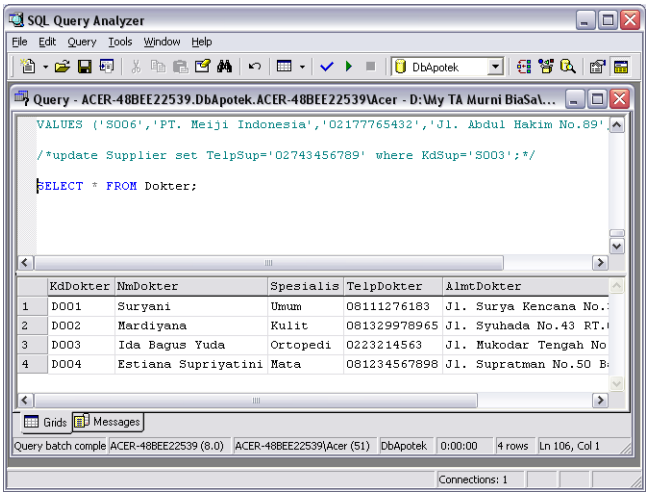


Gambar 2.3 Tampilan SQL Server Enterprise Manager

2. Mengunakan SQL Query Analyzer

   Fitur ini menggunakan Transact SQL (perintah-perintah SQL) untuk mengelola database di dalam SQL Server 2000. Perintah-perintah Transact SQL merupakan pengembangan dari perintah-perintah SQL standar yang disesuaikan dengan manajemen database pada SQL

Server 2000. Transact SQL dapat digunakan untuk dapat membuat database, membuat tabel, mengubah struktur tabel, menghapus database, menghapus tabel, menyisipkan data, mengubah data dan lain-lain.



Gambar 2.4 Tampilan SQL Query Analizer

2.8.3 Windows

Windows adalah salah satu dari sekian banyak sistem operasi yang ada. Windows berada di bawah naungan Microsoft dan tidak bersifat *free* (gratis). Windows sendiri memiliki beberapa keunggulan diantaranya adalah *user friendly* yaitu memudahkan pemakainya dalam penggunaan aplikasi yang terdapat pada windows walaupun pemula sekalipun.

2.9 Deskripsi jabatan

Untuk jabatan yang terdapat pada divisi sales PT. Sysmex secara umum mencakup 3 bagian, yaitu:

1. Sales Manager
2. Supervisor
3. Sales Enginering

2.9.1 Sales Manager

Dalam suatu perusahaan harus terdapat suatu jabatan yang berfungsi untuk mengatur atau me-*manage* suatu kegiatan yang berjalan pada perusahaan tersebut. Pada PT. Sysmex khususnya pada divisi sales terdapat suatu posisi jabatan yang bertugas untuk mengatur seluruh kegiatan seluruh personilnya untuk mencapai tujuan perusahaan meraih penjualan dengan pendapatan yang memuaskan.

Untuk lebih jelasnya sales manager mempunyai tugas-tugas seperti berikut:

1. Mengatur suatu strategi dalam penjualan
2. Memotivasi dan memimpin tim penjualan dalam berbagai sektor
3. Bertanggung jawab atas keuangan pada divisinya
4. Bertanggung jawab atas kinerja tim
5. Mengatur skema insentif yang tujuannya memotivasi tim untuk mencapai atau melebihi target penjualan.
6. Menentukan target penjualan produk perusahaan.

2.9.2 Supervisor

Untuk menunjang suatu kinerja tim yang hebat dalam suatu strategi penjualan, tim harus mempunyai ketua atau *leader* yang fungsinya untuk mengatur segala bentuk kerja tim. Jabatan yang sesuai pada deskripsi tersebut adalah *supervisor*.

Supervisor dalam cara kerjanya hampir sama dengan sales manager dan hanya sedikit perbedaannya, untuk lebih jelasnya tugas seorang *supervisor* adalah sebagai berikut :

1. Memberi jadwal kerja pada tim
2. Mengimplementasikan perencanaan yang telah ditetapkan oleh manager.
3. Mempekerjakan tim sesuai dengan kebutuhan
4. Bertanggung jawab dalam meningkatkan sumber daya manusia pada perusahaan, khususnya pada timnya.
5. Dapat melakukan *coaching* atau pelatihan terhadap sumber daya manusia yang ada.

2.9.3 Sales Engineering

Suksesnya suatu perusahaan tidak mungkin terjadi bila tidak adanya suatu koordinasi tim yang baik. Dalam suatu perusahaan pasti harus memiliki tenaga-tenaga kerja yang baik. Pada divisi sales, posisi yang juga dapat menunjang kesuksesan dalam penjualan produk tiada lain adalah *sales engineering* atau tenaga kerja sales.

Sales engineering dalam tugasnya mencakup beberapa hal sebagai berikut:

1. Melakukan langkah awal dalam sebuah hubungan *customer* dan perusahaan.
2. Menawarkan produk perusahaan pada *customer*.
3. Berusaha mengambil hati para *customer* agar dapat melakukan hubungan kerjasama dengan perusahaan.