BAB II
TINJAUAN PUSTAKA

2.1 Profil Tempat Kerja Praktek

Balai Kesehatan Mata Masyarakat (BKMM) adalah suatu instansi pemerintah yang dibina oleh Kementrian Kesehatan Republik Indonesia berserta Direktorat Jendral Bina Upaya Kesehatan untuk menangani Kesehatan Mata dan juga Telinga Hidung Tenggorokan (THT).

2.1.1 Sejarah Singkat BKMM

Balai Kesehatan Mata Masyarakat yang selanjutnya dengan sebutan BKMM Didirikan pada tahun 1957 dan mulai dipergunakan pada tahun 1958 dengan nama Balai Pengobatan Mata (BP Mata) dalam upaya pemberantasan penyakit trakhoma, yang opersionalnya dilaksanakan oleh beberapa orang tenaga dan di kepalai oleh seorang paramedis terlatih di bidang kesehatan mata.

Pada tahun 1967 ditunjuk seorang dokter umum dari Dinas Kesehatan Karawang. Selanjutnya pada tahun 1973 masuk seorang dokter ahli mata sebagai dokter tamu yang ditunjuk oleh Dinas kesehatan Karawang.

Pada tahun 1982 ditunjuk dokter ahli mata yang menetap dari Kantor Wilayah Departemen kesehatan Republik Indonesia Provinsi Jawa Barat. Secara bertahap mulai mengembangkan ada tindakan operasi dan menerima rujukan dari beberapa puskesmas kecamatan, sehingga dari tahun ke tahun kunjungan semakin meningkat. Untuk mengembangkan pelayanan kesehatan mata, maka pemerintah melalui Surat Keputusan Menteri kesehatan Republik Indonesia No. 350a/Menkes/SK/VI/1991 tentang susunan organisasi, tugas pokok dan fungsi pada Balai Kesehatan Mata Masyarakat yang meliputi pelembagaan dari Balai Pengobatan Umum mata menjadi BKMM yaitu di Propinsi jawa Barat, Sulawesi Selatan dan Sumatera Selatan. Dengan adanya kegiatan UKM/PK dan PKKP yang merupakan program kegiatan penanggulangan kebutaan pelaksanaannya terintegrasi di Puskesmas dengan BKMM Cikampek adalah untuk melayani

kesehatan mata yang pada akhirnya mengurangi prevalensi kebutaan terutama katarak.

Dengan adanya perubahan struktur dan nomenklatur melalui Keputusan Menteri kesehatan Nomor. 1277/Menkes/SK/VI/2001 tentang organisasi dan tata kerja Departemen Kesehatan, maka terjadi perubahan kedudukan Balai Kesehatan Mata Masyarakat yang diperkuat dengan adanya Keputusan Menteri Kesehatan RI Nomor. 556/Menkes/SK/VI/2002 tentang perubahan rumusan kedudukan Unit Pelaksana Teknis di Lingkungan Departemen Kesehatan yang berada di bawah Direkrorat jenderal Bina Kesehatan Masyarakat dan dipimpin oleh seorang kepala yang bertanggungjawab kepada Direktur jenderal. Dalam melaksanakan tugas teknis, secara fungsional dibina oleh Sub Direktorat Indera.

2.1.2 Tujuan Didirikannya BKMM Cikampek

Tersusunnya landasan pemikiran sebagai dasar untuk penataan dan pembentukan Balai Kesehatan Indera Masyarakat Cikampek di Cikampek yang mempunyai tugas pokok untuk meningkatkan derajat kesehatan indera masyarakat Jawa Barat dan sebagai pusat rujukan masalah kesehatan indera di wilayah Indonesia bagian Barat, melalui penyediaan pelayanan kesehatan indera dalam bentuk pelayanan kesehatan spesialistik Kesehatan Mata dan THT.

2.1.3 Strategi BKMM Cikampek

Penjabaran visi, misi dan nilai Balai Kesehatan Masyarakat tersebut diwujudkan melalui strategi sebagai berikut:

a. Meningkatkan kerja sama dengan Dinas Kesehatan Provinsi dan intansi terkait dalam membentuk Forum Kordinasi Penanggulangan Gangguan Penglihatan dan Kebutaan.
b. Meningkatkan advokasi di tingkat Propinsi maupun Kabupaten/Kota.
c. Meningkatkan koordinasi dan kerjasama lintas program dan sektor dalam penanggulangan gangguan penglihatan dan kebutaan.
d. Menggalang kemitraan dalam penanggulangan gangguan penglihatan dan kebutaan dengan instansi terkait.
e. Menggalang kemitraan dengan LSM dalam dan luar negeri.

f. Peningkatan kualitas dan kuantitas sumber daya manusia yang terlibat dalam upaya penanggulangan gangguan penglihatan dan kebutaan.

2.1.4 Visi dan Misi BKMM

1. ` Visi BKMM

Sebagai pusat kegiatan upaya-upaya kesahatan mata masyarakat.

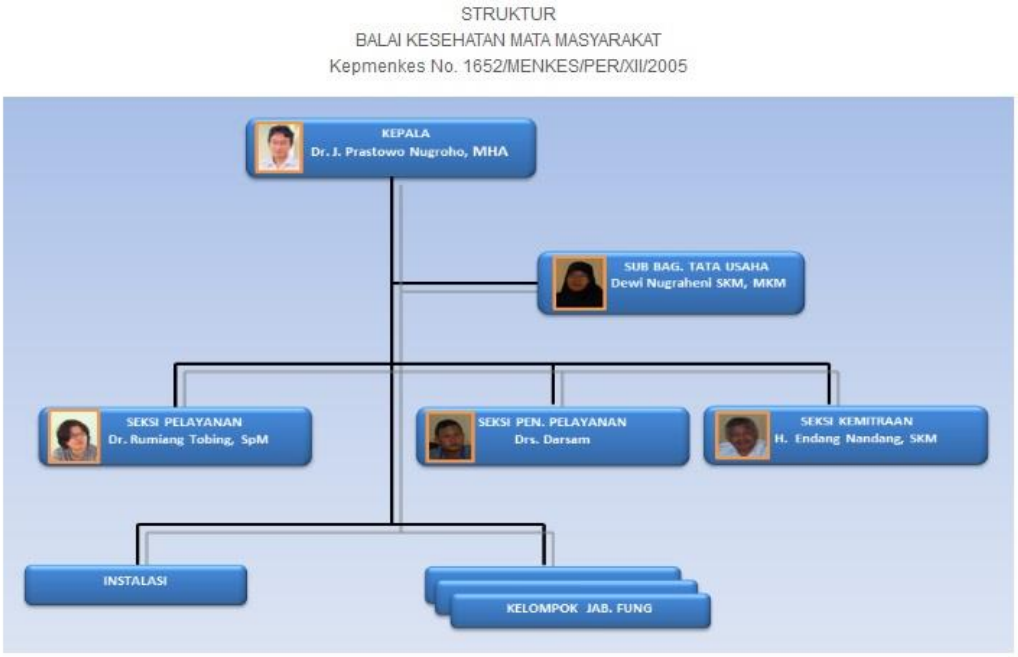
2. Misi BKMM

a. Meningkatkan mutu dan keterjangkauan pelayanan di bidang kesehatan mata masyarakat.
b. Meningkatkan promosi kesehatan mata untuk pemberdayaan masyarakat.
c. Mengembangkan dan meningkatkan Pendidikan dan Pelatihan.
d. Meningkatkan Penelitian dan Pengembangan Teknologi tepat guna.
e. Mengembangkan dan meningkatkan jejaring kemitraan.
f. Meningkatkan pelaksanaan urusan ketatausahaan dan kerumahtanggaan BKMM.
g. Mengembangkan Sumber Daya Manusia (SDM) Tenaga kesehatan dan non kesehatan.
h. Meningkatkan sarana dan prasarana pendukung Pelayanan Kesehatan Mata.

2.1.5 Struktur Organisasi dan *Job Description*

Suatu instansi dapat berjalan dengan baik sesuai dengan tujuannya jika ditunjang dengan suatu struktur organisasi dan anggaran yang tersusun dengan baik pula. Dengan adanya struktur organisasi yang teratur maka setiap fungsi organisasi dapat berjalan sesuai dengan tugas dan perannya masing-masing.

Struktur organisasi tersebut perlu dilengkapi dengan suatu gambaran uraian pekerjaan dan tanggung jawab yang harus dilaksanakan oleh setiap pemegang jabatan sehingga dapat menjadi efekif dan efesien. Tanpa dilengkapi dengan suatu uraian tugas, struktur organisasi tidak dapat melaksanakan tugas dan memenuhi tanggung jawabnya.



Gambar 2.1. Struktur Organisasi dan Tata Kerja

1. Uraian Bagian BKMM

a. Kepala
b. Sub. Bagian Tata Usaha
c. Seksi Pelayanan
d. Seksi Penanggung jawab Pelayanan
e. Seksi Kemitraan
f. Instalasi
g. Kelompok Jabatan Fungsional

2.2 Landasan Teori

Dalam melaksanakan kerja praktek ini digunakan dasar-dasar teori sistem informasi sebagai bahan acuan. Berisi teori-teori yang bisa dijadikan dasar dan acuan dalam perancangan sistem informasi ini.

2.2.1 Pengertian Sistem

Definisi sistem adalah suatu kesatuan yang terdiri atas komponen atau elemen yang saling berinteraksi, saling terkait, atau saling bergantung membentuk

keseluruhan yang kompleks. Dalam definisi yang paling umum, sebuah sistem adalah sekumpulan objek/benda yang memiliki hubungan diantara mereka.

Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (o*utput*), pengolahan (*process*) dan sasaran (*objective*) atau tujuan (*goal*).

a. Komponen Sistem (*components*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu sub sistem atau bagian-bagian dari sistem. Setiap sistem tidak perduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem.

b. Batasan Sistem (*boundary*)

   Merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas sustu sistem menunjukan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan Luar Sistem (*environments*)

   Lingkungan Luar Sistem (*environments*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

d. Penghubung Sistem (*interface*)

   Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya.

Keluaran (*output*) dari subsistem 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 adalah energi yang dimasukan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). Maintenanace input adalah energi yang dimasukan supaya sistem dapat beroperasi. Signal input adalah energi yang diproses untuk mendapatkan keluaran.

f. Pengolahan Sistem (*process*)

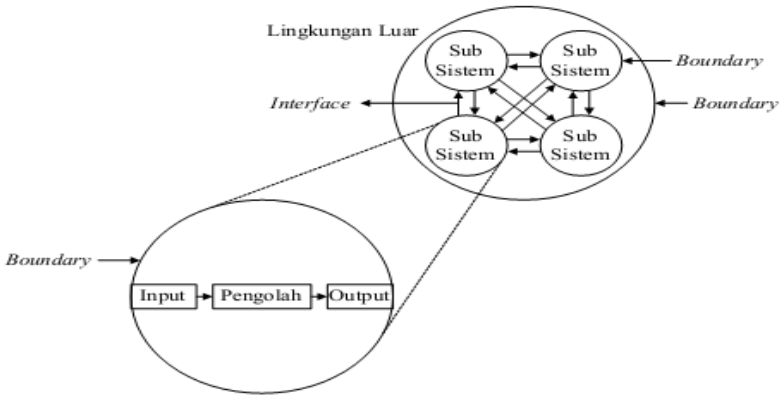
Suatu sistem dapat mempunyai bagian pengolah yang akan mengubah masukan menjadi keluaran.

g. Keluaran Sistem (*output*)

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

h. 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 masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.



Gambar 2.2. Karakteristik Sistem

2.2.2 Pengertian Informasi

Konsep dasar informasi, data adalah fakta atau yang dapat digunakan sebagai input dalam menghasilkan informasi sedangkan informasi adalah “hasil pengolahan data yang memberikan arti dan manfaat bagi orang yang akan menerimanya”.

1. Siklus Informasi

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi. Data yang diolah untuk menghasilkan informasi menggunakan suatu model proses tertentu.

Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lainnya yang akan membuat sejumlah data kembali. Data tersebut akan ditanggap sebagai input, diproses kembali lewat suatu model dan membentuk suatu siklus. Menurut John Burch disebut dengan siklus informasi (*information cycle*) atau siklus ini disebut juga dengan siklus pengolahan data (*data processing cycles*).

2. Kualitas Informasi

Kualitas dari suatu informasi tergantung dari tiga hal, yaitu informasi harus akurat, tepat pada waktunya dan relevan, di antaranya :

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan yang dapat merubah atau merusak informasi tersebut.
2. Tepat pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat.
3. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya.

3. Nilai Informasi

Nilai informasi ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.

2.2.3 Sistem Informasi

Computer Based Information System (CBIS) atau yang dalam bahasa Indonesia disebut juga Sistem Informasi Berbasis Komputer merupakan sistem pengolah data menjadi sebuah informasi yang berkualitas dan dipergunakan untuk suatu alat bantu pengambilan keputusan. Sistem Informasi yang akurat dan efektif, dalam kenyataannya selalu berhubungan dengan istilah "*computer-based*" atau pengolahan informasi yang berbasis pada komputer. Sistem Informasi "berbasis komputer" mengandung arti bahwa komputer memainkan peranan penting dalam sebuah sistem informasi.

Secara teori, penerapan sebuah Sistem Informasi memang tidak harus menggunakan komputer dalam kegiatannya. Tetapi pada prakteknya tidak mungkin sistem informasi yang sangat kompleks itu dapat berjalan dengan baik jika tanpa adanya komputer. Sistem Informasi merupakan sistem pembangkit informasi. Dengan integrasi yang dimiliki antar subsistemnya, sistem informasi akan mampu menyediakan informasi yang berkualitas, tepat, cepat dan akurat sesuai dengan manajemen yang membutuhkannya.

2.2.4. Basis Data

Basis data (*database*), atau sering pula disebut basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system, DBMS*). Sistem basis data dipelajari dalam ilmu informasi.

Istilah "basis data" berawal dari ilmu komputer. Meskipun kemudian artinya semakin luas, memasukkan hal-hal di luar bidang elektronika, artikel ini mengenai basis data komputer. Catatan yang mirip dengan basis data sebenarnya

sudah ada sebelum revolusi industri yaitu dalam bentuk buku besar, kuitansi dan kumpulan data yang berhubungan dengan bisnis.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan di antara obyek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel.

Istilah basis data mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai sistem manajemen basis data (*database management system/DBMS*). Jika konteksnya sudah jelas, banyak administrator dan programer menggunakan istilah basis data untuk kedua arti tersebut.

1. Data, Informasi dan Basis Data

Data merupakan fakta mengenai suatu objek seperti manusia, benda, peristiwa, konsep, keadaan dan sebagainya yang dapat dicatat dan mempunyai arti secara implisit. Data dapat dinyatakan dalam bentuk angka, karakter atau simbol, sehingga bila data dikumpulkan dan saling berhubungan maka dikenal dengan istilah basis data (database).

Sedangkan menurut George Tsu-der Chou basis data merupakan kumpulan informasi bermanfaat yang diorganisasikan ke dalam aturan yang khusus. Informasi ini adalah data yang telah diorganisasikan ke dalam bentuk yang sesuai dengan kebutuhan seseorang.

Menurut Encyclopedia of Computer Science and Engineer, para ilmuwan di bidang informasi menerima definisi standar informasi yaitu data yang digunakan dalam pengambilan keputusan.

Definisi lain dari basis data menurut Fabbri dan Schwab adalah sistem berkas terpadu yang dirancang terutama untuk meminimalkan duplikasi data.

Menurut Ramez Elmasri mendefinisikan basis data lebih dibatasi pada arti implisit yang khusus, yaitu:

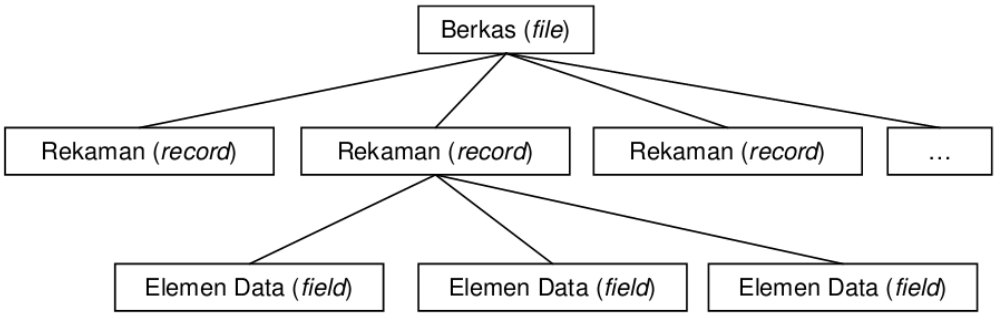
a. Basis data merupakan penyajian suatu aspek dari dunia nyata (*real world*).
b. Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit. Sehingga data yang terkumpul secara acak dan tanpa mempunyai arti, tidak dapat disebut basis data.
c. Basis data perlu dirancang, dibangun dan data dikumpulkan untuk suatu tujuan. Basis data dapat digunakan oleh beberapa user dan beberapa aplikasi yang sesuai dengan kepentingan user.

Dari beberapa definisi-definisi tersebut, dapat dikatakan bahwa basis data mempunyai berbagai sumber data dalam pengumpulan data, bervariasi derajat interaksi kejadian dari dunia nyata, dirancang dan dibangun agar dapat digunakan oleh beberapa user untuk berbagai kepentingan.

2. Hirarki Data

Data diorganisasikan kedalam bentuk elemen data (*field*), rekaman (*record*), dan berkas (*file*). Definisi dari ketiganya adalah sebagai berikut:

a. Elemen data adalah satuan data terkecil yang tidak dapat dipecah lagi menjadi unit lain yang bermakna. Misalnya data siswa terdiri dari NIS, Nama, Alamat, Telepon atau Jenis Kelamin.
b. Rekaman merupakan gabungan sejumlah elemen data yang saling terkait. Istilah lain dari rekaman adalah baris.
c. Berkas adalah himpunan seluruh rekaman yang bertipe sama.

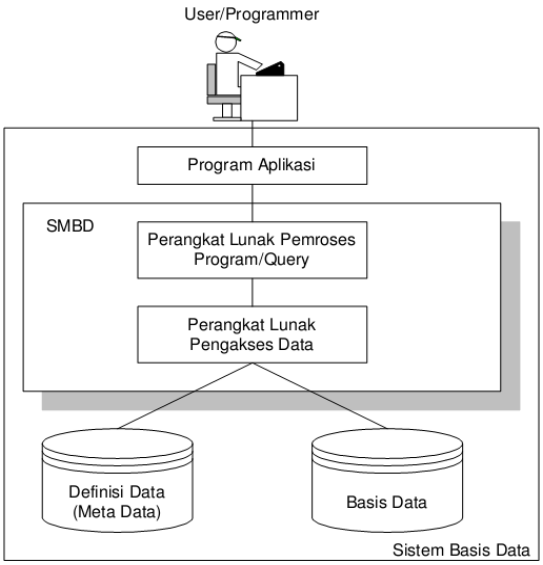


Gambar 2.3. Hirarki Data

3. Sistem Basis Data

Gabungan antara basis data dan perangkat lunak SMBD (Sistem Manajemen Basis Data) termasuk di dalamnya program aplikasi yang dibuat dan bekerja dalam satu sistem disebut dengan Sistem Basis Data.

Sistem basis data dapat dianggap sebagai tempat untuk sekumpulan berkas data yang terkomputerisasi dengan tujuan untuk memelihara informasi dan membuat informasi tersebut tersedia saat dibutuhkan.



Gambar 2.4. Konsep Sistem Basis Data

4. Ikhtisar Basis Data

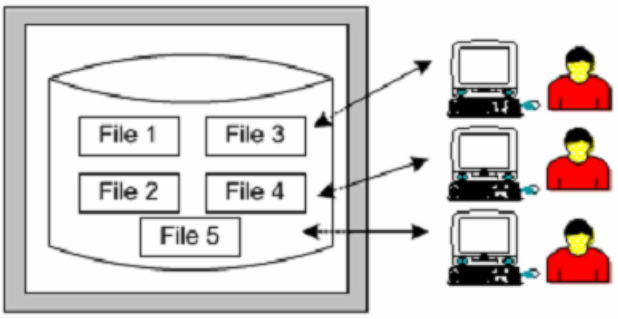
a. **Basis**: markas/gudang, tempat bersarang/berkumpul.
b. **Data**: representasi fakta dunia nyata yang mewakili suatu obyek (seperti, manusia: dosen, mahasiswa, pelanggan,dll; barang: buku, meja; peristiwa, konsep, dan sebagainya.), yang direkam baik dalam bentuk angka, huruf, teks, gambar atau suara.

c. **Definisi Basis Data**

1. Basis Data adalah sekumpulan data yang saling berelasi.
2. Himpunan kelompok data (arsip) yang saling berhubungan, yang diorganisasi sedemikian rupa, sehingga kelak dapat dimanfaatkan kembali dengan cepat.
3. Kumpulan data yang saling berhubungan yang disimpan secara bersama tanpa adanya pengulangan (redudansi) data.
4. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

d. **Sistem Basis Data**

Sistem yang terdiri atas sekumpulan tabel data yang saling berhubungan dan sekumpulan program (DBMS: *Database Management System*) yang memungkinkan berbagai user dan/atau program lain dapat mengakses dan memanipulasi tabel-tabel tersebut.



Gambar 2.5. Sistem Basis Data

e. **DBMS (Database Management System)**

Kumpulan program yang digunakan user untuk me-*management* database (*create, maintain*)

DBMS mencakup proses:

1. *Defining*: database mendefiniskan tipe data, struktur dan batasan (*constraint*) dari data yang disimpan dalam database.
2. *Manipulating*: database mencakup berbagai fungsi dan *query* untuk mendapatkan data yang dicari, termasuk operasi *insert*, *update* dan *delete* serta dalam generate report data.

3. *Sharing*: database dapat diatur untuk dapat *sharing multiple user* dan program untuk mengakses database secara bersama-sama.

Fungsi yang lebih penting dari DBMS adalah proteksi dan maintain database dalam jangka panjang.

1. Proteksi: mengandung *system protection* yang menangani kondisi malfunction (*crash*) baik pada hardware ataupun software, juga mengandung *security protection* yang menangani pengaksesan oleh user terlarang.
2. Maintain: mengandung sistem maintaining yang selalu meningkatkan kebutuhan perubahan tiap waktu.

f. **Sistem Basis Data**

Sistem Basis Data adalah gabungan antara DBMS dengan Basis data.

g. **Struktur File Database**

1. ***Data*** adalah satu satuan informasi yang akan diolah, dimana sebelum diolah dikumpulkan di dalam suatu file database. Pengumpulan data dilakukan secara sistematis menurut struktur file database tersebut.
2. ***RECORD*** adalah data yang isinya merupakan satu kesatuan seperti Nama, Alamat, Nomor Telepon. Setiap keterangan yang mencakup Nama, Alamat dan Nomor Telepon dinamakan satu record. Dan setiap record diberi nomor urut yang disebut nomor record (*Record Number*). Ukuran suatu file database ditentukan oleh jumlah record yang tersimpan di dalamnya.
3. ***FIELD*** adalah subbagian dari Record. Dari contoh isi record diatas maka terdiri dari 3 field, yaitu field Nama, field Alamat dan field Nomor Telepon.

h. **Keuntungan Sistem Basis Data**

1. **Terkontrolnya kerangkapan data**

   Dalam basis data hanya mencantumkan satu kali saja field yang sama yang dapat dipakai oleh semua aplikasi yang memerlukannya.
2. **Terpeliharanya keselarasan (kekonsistenan) data**

   Apabila ada perubahan data pada aplikasi yang berbeda maka secara otomatis perubahan itu berlaku untuk keseluruhan.
3. **Data dapat dipakai secara bersama (*shared*)**

Data dapat dipakai secara bersama-sama oleh beberapa program aplikasi (secara *batch* maupun *on-line*) pada saat bersamaan.

4. **Dapat diterapkan standarisasi**

   Dengan adanya pengontrolan yang terpusat maka DBA dapat menerapkan standarisasi data yang disimpan sehingga memudahkan pemakaian, pengiriman maupun pertukaran data.

5. **Keamanan data terjamin**

   DBA dapat memberikan batasan-batasan pengaksesan data, misalnya dengan memberikan password dan pemberian hak akses bagi user (contoh : *modify, delete, insert, retrieve*).

6. **Terpeliharanya integritas data**

   Jika kerangkapan data dikontrol dan kekonsistenan data dapat dijaga maka data menjadi akurat.

7. **Terpeliharanya keseimbangan (keselarasan) antara kebutuhan data yang berbeda dalam setiap aplikasi**

   Struktur basis data diatur sedemikian rupa sehingga dapat melayani pengaksesan data dengan cepat.

8. **Kemandirian data (*Data independence*)**

   Dapat digunakan untuk bermacam-macam program aplikasi tanpa harus merubah format data yang sudah ada.

i. Contoh DBMS:
   a. Dbase
   b. FoxPro
   c. Ingres
   d. Postgresql
   e. MySQL
   f. MS Access
   g. SQL Server
   h. Oracle
   i. DB2, dll.

j. **Tujuan Basis Data**

1. Kemudahan dan kecepatan dalam pengambilan data (*speed*).
2. Efisiensi ruang penyimpanan (*space*). Mengurangi / menghilangkan redudansi data.
3. Keakuratan (*accuracy*). Pembentukan kode & relasi antar data berdasar aturan / batasan (*constraint*) tipe data, domain data, keunikan data, untuk menekan ketidakakuratan saat entry / penyimpanan data.
4. Ketersediaan (*availability*). Pemilahan data yang sifatnya pasif dari database aktif.
5. Kelengkapan (*completeness*). Kompleksnya data menyebabkan perubahan struktur database.
6. Keamanan (*security*). Memberikan keamanan atas hak akses data.
7. Kebersamaan pemakaian (*sharability*). Bersifat multiuser.

k. **Manfaat penggunaan DBMS**

1. *Controlling Redundancy*

   Redundancy: duplikasi data, penyimpanan data secara berulang.

   Redudancy salah satu syarat larangan dalam database relasional, karena akan menimbulkan inconsistensi data.

   Dengan controlling redundancy, selain akan meningkatkan performance query juga menjaga konsistensi data.
2. *Restricting Unauthorized Access*

   Memberikan pengaturan hak akses / batasan akses user database
3. *Providing Persistent Storage for Program Objects*

   Menyediakan ruang penyimpanan khusus untuk obyek-obyek program (contoh : Object-Oriented Database system yang menyimpan obyek-obyek pemrograman berbasis obyek).
4. *Providing Storage Structures for Efficient Query Processing*

   Menyediakan struktur penyimpanan yang bagus untuk efisiensi proses query.
5. *Providing Backup and Recovery*
6. *Providing Multiple User Interface*
7. *Representing Complex Relationship among Data*

8. *Enforcing Integrity Constraints*
9. *Permitting Inferencing and Actions using Rules*

   Menyediakan actions khusus berdasarkan aturan (*rules*) yang telah ditetapkan dalam sistem database.
10. *Additional Implications of using the Database Approach*

    Flexible, up-to-date data, ekonomis, dan lain lain.

l. **Pengguna Basis Data**

Para pengguna database dapat dibagi menjadi Pengguna database ("*Actor on the scene*") dan Pekerja dibalik database ("*Worker behind the scene*").

1) "*Actor on the scene*" lebih cenderung menggunakan / ada keterkaitan dengan penggunaan database dan dapat dikelompokkan menjadi:

a. *Database Administrators (DBA)***.** Orang yang memiliki tanggung jawab penuh dalam manajemen database (pengaturan hak akses, koordinasi dan monitoring, kebutuhan hardware/software). Dalam pekerjaannya biasanya dibantu oleh staf Admin.

1) Tugas DBA:

   Mengontrol DBMS dan software-software

   Memonitor siapa yang mengakses basis data

   Mengatur pemakaian basis data

   Memeriksa *security*, *integrity*, *recovery* dan *concurency*

2) Program Utility yang digunakan oleh DBA:

   Loading Routines. Membangun versi utama dari database.

   Reorganization Routines. Mengatur / mengorganisasikan kembali database.

   Journaling Routines. Mencatat semua operasi pemakaian database.

   Recovery Routines. Menempatkan kembali data, sebelum terjadinya kerusakan.

   Statistical Analysis Routines. Membantu memonitor kehandalan sistem.

b. *Database Designers*. Orang bertanggung jawab dalam identifikasi data yang tersimpan dalam database, menentukan struktur data yang tepat

untuk disimpan dalam database. Perlu koordinasi akan kebutuhan user database.

c. *End Users*. Adalah orang-orang yang pekerjaannya membutuhkan akses ke database untuk melakukan *query*, *update* maupun *genereate report database.*

1) End user tak tetap (*Casual end users*): user yang tidak selalu mengakses database, tapi kadang memerlukan informasi terbaru.
2) *Naïve / parametric end users*: Pemakai yang berinteraksi dengan sistem basis data melalui pemanggilan satu program aplikasi permanen (*executable program*) yang telah ditulis/disediakan sebelumnya. Pekerjaan selalu konstan query dan update data. (seperti: bank teller, pegawai reservasi, dan lain lain.).
3) *Sophisticated end users*: Pemakai yang menulis aplikasi basis data non konvensional, tetapi untuk keperluan-keperluan khusus seperti aplikasi AI, Sistem Pakar, Pengolahan Citra, dan lain lain, yang bisa saja mengakses basis data dengan/tanpa DBMS yang bersangkutan. Melengkapi kebutuhan database user. (seperti: *engineer*, *scientist*, *business analyst*).
4) *Stand-alone users*: user yang memaintain personal database.

d. *System Analyst dan Application Programmers (Software Engineers)*

1) *System Analyst*: orang menentukan kebutuhan sistem end user.
2) *Application Programmers (Software Engineering)*: orang yang kerjaannya berhubungan dengan kebutuhan koneksi database.

2) *"Workers behind the scene"*. Orang-orang yang tidak tertarik pada database, akan tetapi lebih cenderung pekerjaannya men-*develop tool* untuk kebutuhan database. Pengguna dapat dikelompokkan menjadi:

a. *DBMS system designers dan implementer*. Orang-orang yang merancang dan mengimplementasikan modul-modul dan interface paket-paket software DBMS. (contoh Modul: catalog, *procs query lang*, *procs interface*, *access & buffering data, controlling cuncurrency, handling data recovery & security, interfacing: interface for integrated system*).

b. *Tool developers*. Orang-orang yang merancang dan mengimplementasikan tools untuk mendukung software DBMS (tool untuk meningkatkan performance database, tool untuk monitoring operasional database, dan lain lain).
c. *Operators dan maintenance personnel*. Para personel administrator yang bertanggung jawab akan jalannya operasional database termasuk *maintenance (hardware/software)* DBMS.

m. **Komponen Sistem Basis Data**

1. Perangkat Keras (*Hardware*). Komputer, memori, storage (Harddisk), peripheral, dan lain lain.
2. Sistem Operasi (*Operating System*). Program yang menjalankan sistem komputer, mengendalikan resource komputer dan melakukan berbagai operasi dasar sistem komputer.
3. Basis Data (*Database*). Menyimpan berbagai obyek database (struktur tabel, indeks, dan lain lain).
4. DBMS (*Database Management System*). Perangkat lunak yang memaintain data dalam jumlah besar.
5. Pemakai (*User*). Para pemakai database.
6. Aplikasi (perangkat lunak) lain. Program lain dalam DBMS.

n. **Abstraksi Data.** Dalam database, data disimpan dan diperlihara dengan baik dan terstruktur oleh DBMS. Sistem ini menyembunyikan detail tentang bagaimana data disimpan dan dipelihara. Sehingga seringkali data yang terlihat oleh user, berbeda dengan data yang tersimpansecara fisik. Abstraksi data merupakan tingkatan/level dalam melihat bagaimana menampilkan data dalam sebuah sistem database.

Terdapat 3 Level abstraksi data:

1) Level Fisik (*Physical Level*). Level terendah dalam abstraksi data, yang menunjukkan bagaimana sesungguhnya suatu data disimpan. User melihat data sebagai gabungan dari struktur dan datanya sendiri.

Tingkatan ini berurusan dengan:

a. Alokasi ruang penyimpanan untuk data dan indeks

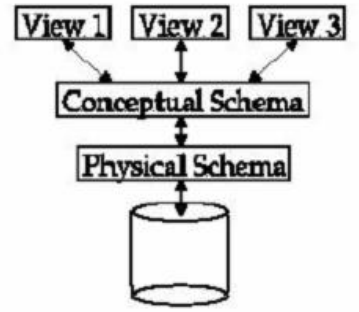
b. Deskripsi record untuk penyimpanan
c. Penempatan record data
d. Teknik kompresi dan enkripsi data

2) Level Konsepsual (*Conceptual Level*). Menggambarkan data apa yang sebenarnya disimpan dalam database, serta hubungannya (*relationship*) dengan data lainnya. Berisi struktur logika database yang hanya dapat dilihat oleh DBA.

Tingkat konsepsual ini menyatakan:

a. Entitas, atribut dan relasinya
b. Konstrain-konstrain terhadap data
c. Informasi semantiks data
d. Informasi keamanan dan integritas data

3) Level Pandangan (*View Level*). Level tertinggi dari abstraksi data, yang hanya menampilkan data hanya sebagian dari database. Tidak semua user membutuhkan semua data dalam database.



Gambar 2.6. Abstraksi Data

o. **Bahasa Basis Data**. DBMS merupakan perantara antara user dengan database. Cara komunikasi diatur dalam suatu bahasa khusus yang telah ditetapkan oleh DBMS (Contoh: SQL, dBase, QUEL, dan sebagainya.). Bahasa database, dibagi dalam 2 bentuk yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML).

1) Data Definition Language (DDL)

a) Digunakan dalam membuat tabel baru, indeks, mengubah tabel, menentukan struktur tabel, dan sebagainya.

b) Hasil dari kompilasi perintah DDL berupa kumpulan tabel yang disimpan dalam file khusus: Kamus Data (*Data Dictionary*).

c) *Data Dictionary* merupakan metadata (*superdata*), yaitu data yang mendeskripsikan data sesungguhnya. *Data dictionary* ini akan selalu diakses dalam suatu operasi database sebelum suatu file data yang sesungguhnya diakses.

2) Data Manipulation Language (DML)

a) Digunakan dalam memanipulasidan pengambilan data pada database.

b) Manipulasi data, dapat mencakup:

a. Pemanggilan data yang tersimpan dalam database (*query*)

b. Penyisipan/penambahan data baru ke database (*Insert*)

c. Pengubahan data pada database (*Update*)

d. Penghapusan data dari database (*Delete*)

c) Terdapat dua (2) jenis DML:

a. Prosedural. Menghendaki user untuk menspesifikasikan data apa yang diperlukan dan bagaimana cara mendapatkan data itu. (Contoh: bahasa C/C++, PL/SQL, dan sebagainya).

b. Nonprosedural. Menghendaki user untuk menspesifikasikan data apa yang dibutuhkan, tanpa harus menspesifikasikan bagaimana cara mendapatkan data tersebut. (Contoh: SQL).

p. **Istilah - Istilah Dasar Basis Data**

1) **Enterprise.** Suatu bentuk organisasi seperti: bank, universitas, rumah sakit, pabrik, dan sebagainya. Data yang disimpan dalai basis data merupakan data operasional dari suatu enterprise. (Contoh data operasional: data keuangan, data mahasiswa, data pasien).

2) **Entitas.** Suatu obyek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalai basis data. Contoh Entitas dalam lingkungan bank terdiri dari : Nasabah, Simpanan, Hipotik Contoh Entitas dalam lingkungan universitas terdiri dari : Mahasiswa, mata kuliah Kumpulan dari entitas disebut **Himpunan Entitas (**Contoh: semua nasabah, semua mahasiswa).

3) **Atribut (Elemen Data).** Karakteristik dari suatu entitas. (Contoh: Entitas Mahasiswa atributnya terdiri dari Npm, Nama, Alamat, Tanggal lahir).

4) **Nilai Data *(Data Value*).** Isi data / informasi yang tercakup dalai setiap elemen data. (Contoh: Atribut Nama Mahasiswa dapat berisi Nilai Data: Diana, Sulaeman, Lina).
5) **Kunci Elemen Data (*Key Data Element*).** Tanda pengenal yang secara unik mengidentifikasikan entitas dari suatu kumpulan entitas. (Contoh: Entitas Mahasiswa yang mempunyai atribut-atribut npm, nama, alamat, tanggal lahir menggunakan Kunci Elemen Data npm.)
6) **Record Data.** Kumpulan Isi Elemen data yang saling berhubungan. (Contoh: kumpulan atribut npm, nama, alamat, tanggal lahir dari Entitas Mahasiswa berisikan : "10200123", "Sulaeman", "Jl. Sirsak 28 Jakarta", "8 Maret 1983".)

2.2.5. Pemodelan Data

Pemodelan sistem memainkan peranan yang penting dalam pengembangan sistem. Pemodelan data kadang-kadang disebut pemodelan database karena model data kadang-kadang diimplementasikan sebagai sebuah database. Pemodelan data dapat di gambarkan dengan ERD (*Entity Relationship Diagram*).

Entity Relationship Diagram (ERD) merupakan suatu diagram yang digunakan untuk menghubungkan antar elemen (*Relational Condition*), dimana pada tahap selanjutnya dapat diimplementasikan kedalam bentuk tabel relasi.

ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data, pada dasarnya ada 3 macam simbol yang digunakan, yaitu:

1. Entity

   Adalah suatu objek yang dapat diidentifikasi dalam lingkaran pemakaian dan sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat.
2. Atribut

   Elemen dari sebuah entity yang berfungsi mendeskripsikan karakter entity.
3. Hubungan

   Sebagaimana halnya entity maka dalam hubungan pun harus dibedakan antara hubungan atau bentuk hubungan antara entity dengan isi dari hubungan itu sendiri.

Beberapa macam hubungan antar relasi, antara lain:

a. Satu Ke Satu (*One to One*)

   Bentuk relasi antara satu entitas dengan sejumlah satu ke entitas dengan jumlah yang sama.

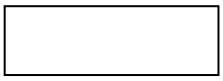
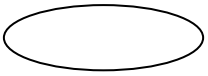
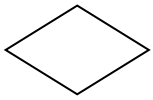
b. Satu Ke Banyak (*One to Many*)

   Bentuk relasi dari entitas dengan jumlah satu ke entitas lain yang berjumlah lebih dari satu (Entitas dengan banyak alternatif tujuan).

c. Banyak Ke Banyak (*Many to Many*)

   Bentuk relasi yang mendeskripsikan permasalahan yang komplek yaitu hubungan antara entitas yang berjumlah lebih dari satu dengan entitas yang sama.

Tabel 2.1. Tabel Notasi yang digunakan pada *Entity Relational Diagram* (ERD)

No	Nama simbol	Fungsi	Simbol
1	Persegi Panjang	Menyatakan suatu entitas	
2	Elips	Menyatakan atribut	
3	Belah Ketupat	Menyatakan relasi	
4	Garis	Menyatakan hubungan antar entitas dengan relasi atau hubungan antar entitas dengan relasi	

2.2.6. Analisis Sistem

Analisis sistem didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

Sedangkan menurut Abdul Kadir (2003: 4), analisis sistem merupakan tahapan yang dimulai karena adanya permintaan terhadap sistem baru. Dimana, permintaan dapat datang dari seorang manajer di luar departemen sistem

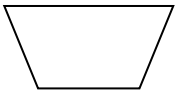
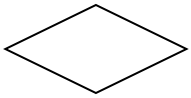
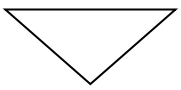
informasi atau dari pihak eksekutif yang melihat adanya masalah atau menemukan adanya peluang baru. Sehingga tujuan utama analisis sistem adalah untuk menetukan hal-hal detail yang akan dikerjakan oleh sistem yang akan diusulkan.

Pada model analisis terdapat perangkat lunak yang dapat digambarkan dalam bentuk sebagai berikut:

1. Flowmap

   Flowmap adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Flowmap berguna untuk membantu analis dan programer untuk memecahkan masalah kedalam segmen yang lebih kecil dan menolong dalam menganalisis alternatif pengoperasian. Biasanya flowmap mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

Tabel 2.2. Tabel Notasi yang digunakan pada Flowmap

No	Nama simbol	Fungsi	Simbol
1	Dokumen	Menyatakan suatu formulir/document	
2	Trapesium	Menyatakan proses yang dilakukan secara manual	
3	Persegi Panjang	Menyatakan proses yang dilakukan secara otomatis	
4	Belah Ketupat	Menyatakan proses pengambilan keputusan benar atau salah	
5	Segitiga	Menyatakan arsip	
6	Garis Penghubung	Menggambarkan aliran document	

2. Diagram Konteks

   Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem

atau output dari sistem.Ia akan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh boundary (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses.Tidak boleh ada store dalam diagram konteks.

Diagram konteks berisi gambaran umum (secara garis besar) sistem yang akan dibuat. Secara kalimat, dapat dikatakan bahwa diagram konteks ini berisi "siapa saja yang memberi data (dan data apa saja) ke sistem, serta kepada siapa saja informasi (dan informasi apa saja) yang harus dihasilkan sistem." Jadi dalam diagram ini yang dibutuhkan adalah:

a. Siapa saja pihak yang akan memberikan data ke sistem
b. Data apa saja yang diberikannya kesistem
c. Kepada siapa sistem harus memberikan informasi atau laporan
d. Apa saja isi atau jenis laporan yang harus dihasilkan sistem

3. *Data Flow Diagram* (DFD)

DFD atau diagram alir data adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana 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 yang dikenakan pada data tersebut.

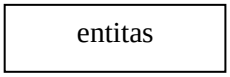
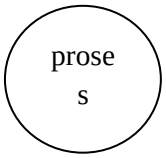
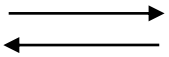
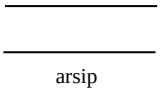
Terdapat 4 (empat) macam simbol yang digunakan dalam *Data Flow Diagram*, diantaranya:

a. Kesatuan luar (*external entity*) atau batas sistem (*Boundary*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain yang berada di lingkungan luarnya yang akan memberikan input atau menerima output dari sistem.
b. Arus Data (*Data Flow*) Arus data ini mengalir diantara proses, simpanan data dan kesatuan luar. Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem. Arus data ini ditunjukkan dengan simbol panah.
c. Proses (*process*) adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil arus data yang masuk ke

dalam proses untuk dihasilkan arus data yang akan keluar dari proses.

d. Simpanan data (*Data Store*) 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

Tabel 2.3. Tabel Notasi yang digunakan pada Data Flow Diagram (DFD)

No	Nama simbol	Fungsi	Simbol
1	Terminator	Menggambarkan sumber dan tujuan data di luar sistem	
2	Proses	Menggambarkan entitas atau proses dimana aliran data masuk dikonfirmasikan ke aliran data keluar	
3	Data Flow	Menggambarkan aliran data	
4	File	Menggambarkan tempat data disimpan	

2.2.7. Kamus Data

Kamus data adalah daftar yang mencatat tentang banyaknya proses yang terjadi dalam sebuah sistem. Secara umum kamus data diklasifikasikan menjadi dua yaitu:

1. Kamus data elementer yaitu daftar tentang semua elemen data yang berhubungan dengan sistem sehingga data yang mengalir dapat didefinisikan dan dapat tersimpan secara lengkap.
2. Kamus Data Komposit, yaitu daftar tentang semua elemen data yang berhubungan dengan sistem dimana elemen data komponen ini terdiri dari dua elemen data elemen yang saling berkaitan.

2.2.8 Pengolahan Data

Pengelolaan data adalah serangkaian operasi atas informasi yang direncanakan guna mencapai tujuan atau hasil yang diinginkan.

Tujuan utama dalam pengolahan data dalam sebuah database adalah agar dapat memperoleh kembali data yang ingin dicari dengan mudah dan cepat, selain itu pemanfaatan database memiliki beberapa tujuan. Secara lengkap pemanfaatan database memliki beberapa tujuan dan dilakukan untuk memenuhi sejumlah tujuan (obyektif) antara lain:

a. Kecepatan dan Kemudahan (*Speed*)
b. Efesiensi ruang (*Space*)
c. Ketersediaan (*Availability*)
d. Kelengkapan (*Completely*)
e. Keamanan (*Security*)
f. Kebersamaan Pemakai (*Sharability*)

2.2.9. Teknologi informasi

Teknologi Informasi biasa disebut TI, IT (*Information Technology*) atau Infotech. Berbagai definisi teknologi informasi telah diutarakan oleh beberapa ahli, diantaranya:

a. Haag den Keen (1996), Teknologi Informasi adalah seperangkat alat yang membantu Anda bekerja dengan informasi dan melakukan tugas-tugas yang berhubungan dengan pemrosesan informasi.
b. Martin (1999), Teknologi Informasi tidak hanya terbatas pada teknologi komputer (perangkat keras atau lunak) yang digunakan untuk memproses dan menyimpan informasi, melainkan juga mencakup teknologi komunikasi untuk mengirimkan informasi.
c. Williams dan Swayer (2003), Teknologi Informasi adalah teknologi yang menggabungkan komputasi (komputer) dengan jalur komunikasi berkecepatan tinggi yang membawa data, suara dan video.

Dari definisi diatas terlihat bahwa teknologi informasi baik secara implisit maupun eksplisit tidak sekedar berupa tekologi komputer, tetapi juga teknologi telekomunikasi. Dengan kata lain, yang disebut teknologi informasi adalah gabungan antara teknologi komputer dan telekomunikasi.

2.2.10. Rekayasa Perangkat Lunak (RPL)

Istilah Rekayasa Perangkat Lunak (RPL) secara umum disepakati sebagai terjemahan dari istilah *Software Engineering*. Istilah Software Engineering mulai dipopulerkan tahun 1968 pada *Software Engineering Conference* yang diselenggarakan oleh NATO. Sebagian orang mengartikan RPL hanya sebatas pada bagaimana membuat program komputer. Padahal ada perbedaan yang mendasar antara perangkat lunak (*software*) dan program komputer.

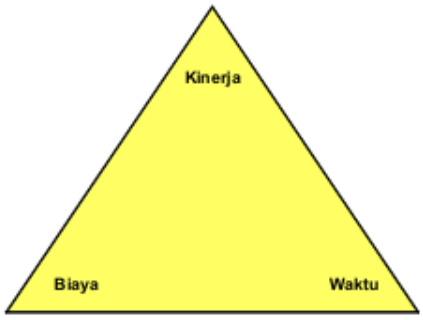
Perangkat lunak adalah seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program atau prosedur. Program adalah kumpulan perintah yang dimengerti oleh komputer sedangkan prosedur adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi (O'Brien, 1999). Pengertian RPL sendiri adalah sebagai berikut:

"Suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, desain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan."

Jelaslah bahwa RPL tidak hanya berhubungan dengan cara pembuatan program komputer. Pernyataan " semua aspek produksi" pada pengertian di atas, mempunyai arti semua hal yang berhubungan dengan proses produksi seperti manajemen proyek, penentuan personil, anggaran biaya, metode, jadwal, kualitas sampai dengan pelatihan pengguna merupakan bagian dari RPL.

1. Tujuan Rekayasa Perangkat Lunak

Secara umum tujuan RPL tidak berbeda dengan bidang rekayasa yang lain. Perhatikan gambar berikut ini.



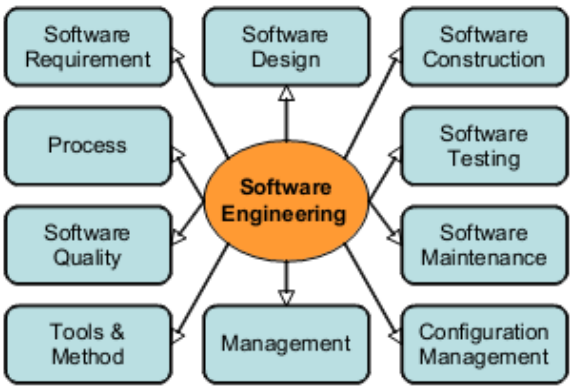
Gambar 2.7 Tujuan RPL

Dari gambar di atas dapat diartikan bahwa bidang rekayasa akan selalu berusaha menghasilkan output yang kinerjanya tinggi, biaya rendah dan waktu penyelesaian yang tepat. Secara lebih khusus kita dapat menyatakan tujuan RPL adalah:

a. Memperoleh biaya produksi perangkat lunak yang rendah.
b. Menghasilkan perangkat lunak yang ki nerjanya tinggi, andal dan tepat waktu.
c. Menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis platform.
d. Menghasilkan perangkat lunak yang biaya perawatannya rendah.

2. Ruang Lingkup

Sesuai definisi yang telah disampaikan sebelumnya, maka ruang lingkup RPL dapat digambarkan sebagai berikut.



Gambar 2.8 Ruang Lingkup RPL

a. *Software requirements* berhubungan dengan spesifikasi kebutuhan dan persyaratan perangkat lunak.

b. *Software design* mencakup proses penentuan arsitektur, komponen, antarmuka, dan karakteristik lain dari perangkat lunak.
c. *Software construction* berhubungan dengan detil pengembangan perangkat lunak, termasuk algoritma, pengkodean, pengujian, dan pencarian kesalahan.
d. *Software testing* meliputi pengujian pada keseluruhan perilaku perangkat lunak.
e. *Software maintenance* mencakup upaya-upaya perawatan ketika perangkat lunak telah dioperasikan.
f. *Software configuration management* berhubungan dengan usaha perubahan konfigurasi perangkat lunak untuk memenuhi kebutuhan tertentu.
g. *Software engineering management* berkaitan dengan pengelolaan dan pengukuran RPL, termasuk perencanaan proyek perangkat lunak.
h. *Software engineering tools and methods* mencakup kajian teoritis tentang alat bantu dan metode RPL.
i. *Software engineering process* berhubungan dengan definisi, implementasi, pengukuran, pengelolaan, perubaha n dan perbaikan proses RPL.
j. *Software quality* menitikberatkan pada kualitas dan daur hidup perangkat lunak.

2.2.11. Pengertian Microsoft Visual Studio 2010

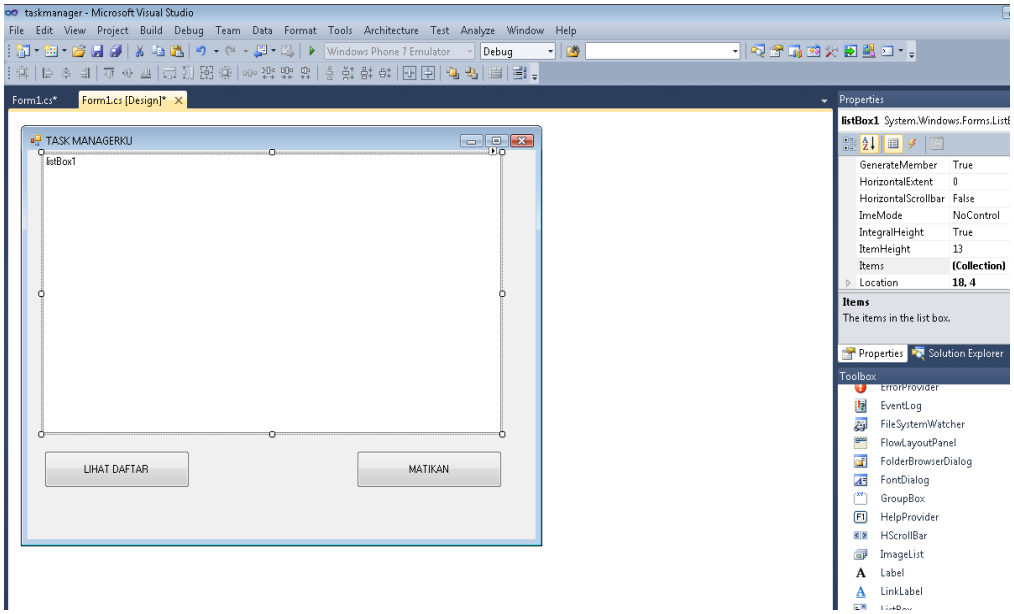
Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, Integrated Development Environment (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk Microsoft Intermediate Language di atas

.NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).Visual Basic merupakan turunan bahasa BASIC dan menawarkan pengembangan aplikasi komputer berbasis grafik dengan cepat, akses ke basis data menggunakan Data Access Objects (DAO), Remote Data Objects (RDO), atau ActiveX Data Object (ADO), serta menawarkan pembuatan kontrol ActiveX dan objek ActiveX. Beberapa bahasa skrip seperti Visual Basic for Applications (VBA) dan Visual Basic Scripting Edition (VBScript), mirip seperti halnya Visual Basic, tetapi cara kerjanya yang berbeda.

Pada tahun 1992 Visual Basic versi 2.0 dirilis untuk meningkatkan dan memperbaiki kinerja dari Visual Basic versi 1.0. Pada tahun 1993 Microsoft mengeluarkan kembali versi terbaru dari Visual Basic ini yakni Visual Basic 3.0 yang ditambahkan diantaranya adanya kontrol data yang berfungsi untuk mengakses database, dan juga sudah tersedia Jet Engine versi 1.1 yang digunakan sebagai tool koneksi database melalui DAO (Data Access Object). Pada tahun 1995 Visual Basic 4.0 dirilis dan sudah dimasukkan teknologi OLE (Object Linking Embedding) serta kemampuan untuk membuat control sendiri (OCX). Pada tahun 1997 Visual Basic 5.0 dirilis dengan memasukkan teknologi baru yang mendukung COM serta memungkinkan membuat kontrol ActiveX sendiri ataupun DLL. Pada tahun 1998 Visual BAsic 6.0 dirilis dengan teknologi yang lebih ditingkatkan lagi khususnya dalam mengakses SQL Server dan mengusung ADO (ActiveX Data Object) yang dirancang untuk meningkatkan kinerja dalam mengakses database pada perusahan besar.

Pada tahun 2000 Microsoft memperkenalkan pertama kali teknologi .Net tetapi belum banyak menarik minat. Hingga pada tahun 2002 Microsoft merilis Visual Studio .NET 2002 yang dipaket dengan Visual Basic .NET, Visual C++ .NET, Visual C# .NET dan Visual J#. NET. Pada tahun 2003 Visual Studio 2003 dirilis untuk memperbaiki kinerja dari visual Studio 2002 dengan meluncurkan .NET Framework versi 1.1. Hingga pada tahun 2009 Visual Studio sudah pada versi 2010, dimana didalamnya sudah ditambahkan bahasa pemrogramana yang baru yaitu F#.



Gambar 2.8 Microsoft Visual Studio 2010

2.2.12. Pengertian ODBC

OpenDatabase Connectivity (disingkat menjadi ODBC) adalah sebuah standar terbuka untuk konektivitas antar mesin basis data. Standar ini menyediakan API yang dapat digunakan untuk menjalankan dan mengoneksikan sebuah aplikasi dengan sebuah system manajemen basis data (SMBD). Para desainer ODBC membuatnya dengan tujuan agar ODBC terbebas dari penggunaan bahasa pemrograman tertentu, sistem manajemen basis data tertentu, dan system operasi tertentu.

Open Database Connectivity (ODBC) adalah suatu kerangka kerja umum untuk mengakses dan mengubah isi database. Ketika sebuah tipe database yang baru diinstal, administrator hanya perlu menginstal driver ODBC yang mendukung platform dan perangkat lunak yang ada ODBC harus berfungsi normal.

ODBC dirancang untuk memaksimumkan interoperabilitas - yaitu kemampuan sebuah aplikasi untuk mengakses sistem manajemen database (DBMS) yang berbeda dengan kode sumber yang sama. Aplikasi Database memanggil fungsi-fungsi dalam antarmuka ODBC, yang diimplementasikan

dalam modul spesifik -database yang disebut driver. Penggunaan aplikasi isolat driver dari panggilan database-spesifik dengan cara yang sama bahwa driver printer mengisolasi program pengolah kata dari perintah spesifik -printer. Karena driver yang dimuat pada waktu bersamaan, pengguna hanya harus menambahkan driver baru untuk mengakses sebuah DBMS baru; tidak perlu mengkompilasi ulang aplikasi.

Spesifikasi ODBC menawarkan API prosedural untuk menggunakan query dengan bahasa SQL untuk mengakses sebuah basis data. Sebuah implementasi ODBC, akan menyediakan satu aplikasi atau lebih, pustaka inti ODBC, dan juga "driver basis data". Pustaka inti ODBC, yang bersifat independen terhadap aplikasi dan juga DBMS, bertindak sebagai interpreter antara aplikasi dan juga driver basis data, sementara driver basis data mengandung detail-detail mengenai SMBD tertentu. Sehingga, dengan cara seperti ini, para programmer dapat menulis aplikasi basis data, tanpa harus memahami sistem manajemen basis data tertentu, mengingat semuanya telah ditangani oleh ODBC. Akan tetapi, para pembuat driver basis data ODBC hanya harus mengetahui bagaimana caranya memasukkan driver basis data ke dalam pustaka inti ODBC. Dengan begitu, ODBC ini dapat disebut sebagai sistem yang modular.

Dengan menggunakan laporan ODBC dalam program, maka dapat mengakses file dalam beberapa database yang berbeda, termasuk akses, dBase, DB2, Excel, dan Teks. Selain perangkat lunak ODBC, modul yang terpisah atau driver yang diperlukan untuk setiap database yang akan diakses. Para pendukung utama dan pemasok mendukung program ODBC adalah Microsoft.

ODBC digunakan setiap kali sebuah aplikasi membutuhkan kemerdekaan database serta data yang akan diakses secara bersamaan dari database yang berbeda. Pelaksana permintaan melalui driver ODBC lebih lambat dibandingkan dengan metoda pribumi mengeksekusi query mentah di database SQL screen editor. Namun ODBCs adalah cara terbaik dan paling efisien untuk menyambung ke database yang berbeda secara bersamaan.

Jika menggunakan ODBC, maka apabila ingin melakukan perubahan field pada report, perlu dilakukan verify database yang berhubungan dengan database secara langsung



Gambar 2.9 Tampilan ODBC

2.2.13. Pengertian XAMPP

XAMPP adalah sebuah software web server apache yang didalamnya sudah tersedia database server mysql dan support php programming. XAMPP merupakan software yang mudah digunakan, gratis dan mendukung instalasi di Linux dan Windows. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia Apache Web Server, MySQL Database Server, PHP Support (PHP 4 dan PHP 5) dan beberapa module lainnya. Hanya bedanya kalau yang versi untuk Windows sudah dalam bentuk instalasi grafis dan yang Linux dalam bentuk file terkompresi tar.gz. Kelebihan lain yang berbeda dari versi untuk Windows adalah memiliki fitur untuk mengaktifkan sebuah server secara grafis, sedangkan Linux masih berupa perintah-perintah di dalam console. Oleh karena itu yang versi untuk Linux sulit untuk dioperasikan. Dulu XAMPP untuk Linux dinamakan LAMPP, sekarang diganti namanya menjadi XAMPP FOR LINUX.