BAB II

TINJAUAN PUSTAKA

2.1 Sejarah Perusahaan

Perseroan Komanditer CV. Multi Karya Mandiri sebuah perusahaan penjualan komputer yang menyediakan berbagai perlengkapan komputer baik *hardware, accessories,* serta layanan *service* bagi perusahaan pemerintah, swasta maupun masyarakat umum. CV. Multi Karya Mandiri berdiri pada pertengahan April 2000 dengan bidang usaha sebagai *dealer* dan *retail* komputer (*hardware*), instalasi jaringan komputer, serta layanan *service* komputer dan aksesoris komputer.

Sejak berdiri hingga saat ini, perusahan ini telah memberikan layanan terbaiknya kepada begitu banyak perusahaan - perusahan , pemerintah, swasta maupun masyarakat umum. Dukungan penuh dari tenaga-tenaga professional dan berpengalaman menjadikan perusahaan ini yakin, bahwa kualitas pelayanan yang diberikan akan semakin baik hari demi hari.

Adapun visi dan misi dari CV. Multi Karya Mandiri antara lain :

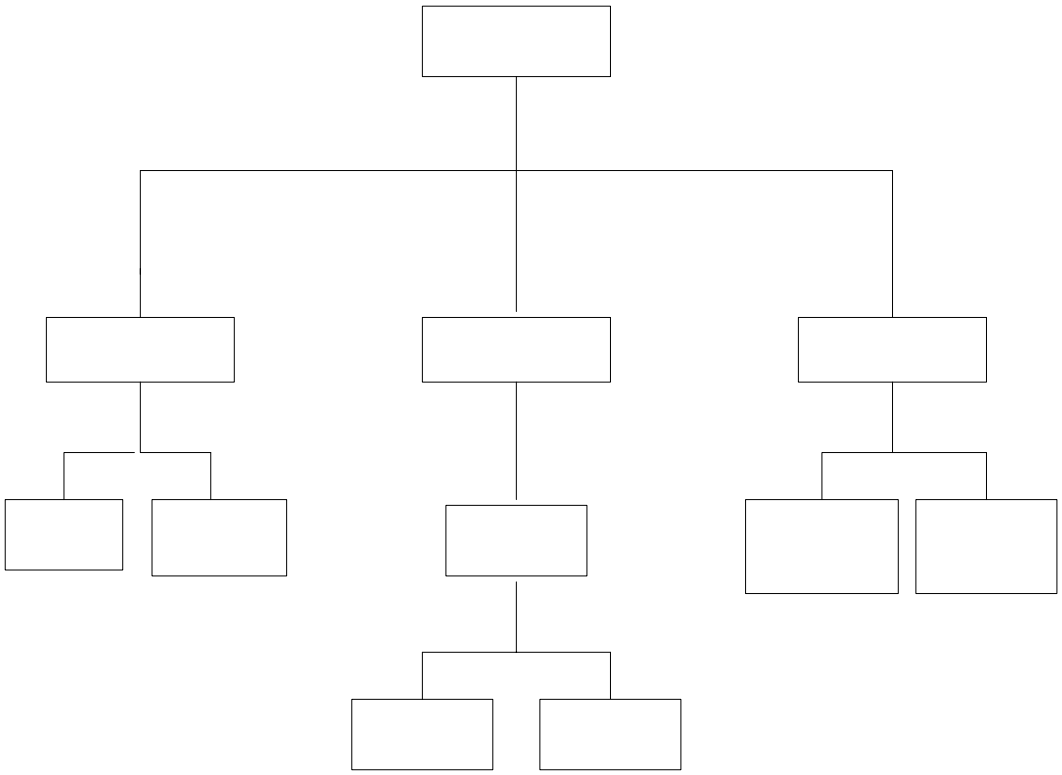
1. Memberikan pelayanan yang baik dan respons yang cepat kepada pelanggan.
2. Menciptakan konsep desain yang inovatif dan kreatif dengan ditunjang peralatan teknologi yang memadai.
3. Membentuk *Team Work* yang *solid* untuk setiap proyek guna memperoleh hasil kerja yang optimal.

4. Menjalin hubungan dan kerjasama yang baik melalui Integritas, kepercayaan dan kesepakatan bersama.
5. Memberikan *Service* berupa perbaikan dan pemeliharaan atas hasil kerja untuk memenuhi kesepakatan-kesepakatan kami sesuai dengan kebutuhan pelanggan.

Berkembangnya teknologi internet yang telah menjadi *platform* penting dalam Telekomunikasi, membuat CV Multi Karya Mandiri memperluas misi untuk menjadi penopang pertumbuhan untuk memenuhi kebutuhan masyarakat akan komputer. Dari mulai layanan konsultasi komputer sederhana hingga penjualan *part* komputer baik regional maupun nasional.

2.1.1 Struktur Organisasi Perusahaan

Dari bentuk diagram organisasi pada suatu perusahaan dapat dianalisa sifat dan bentuk keputusan dari level tertinggi unit perusahaan tersebut ke level di bawahnya. Menurut Robert G. Murdick studi atas organisasi dalam perancangan sistem informasi merupakan tahap pertama yaitu tahap studi (*Study phase*), perencanaan suatu sistem informasi yang dimulai dari suatu organisasi baru membuat suatu rancangan sistem informasi dianalisa secara menyeluruh termasuk analisa terhadap stuktur organisasi yang ada pada perusahaan tempat sistem informasi hendak dibangun.



Gambar 2.1 Struktur Organisasi CV. Multi Karya Mandiri

2.2 Pengertian Sistem

Suatu sistem dapat didefinisikan sebagai suatu kesatuan yang terdiri dari data atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan suatu organisasi seperti perusahaan atau suatu bidang fungsional cocok dengan definisi ini, karena organisasi terdiri dari berbagai sumber daya yang bekerja menuju tercapainya suatu tujuan tertentu yang ditentukan pemilik atau manajemen.

Menurut Tavri D mahyuzir dikemukakan bahwa :

Sistem adalah kumpulan elemen – elemen yang saling berkaitan dan bertanggung jawab memproses masukan sehingga menghasilkan keluaran (Tavri D . Mahyuzir, 1989:1).

Elemen – elemen yang menyusun sebuah sistem tersebut terdiri dari :

1. Tujuan

   Merupakan tujuan dari sistem tersebut berupa tujuan usaha, kebutuhan, masalah, prosedur pencapaian tujuan.

2. Batasan

   Merupakan batasan – batasan yang ada dalam mencapai tujuan dari sistem, dimana batasan ini dapat berupa peraturan – peraturan, biaya – biaya, personil, peralatan, dll

3. *Control*

   Merupakan pengawas dari pelaksanaan pencapaian tujuan sistem yang dapat berupa control pemasukan data (*input*), *control* keluaran dan (*output*), control pengoperasian, dll.

4. *Input*

   Merupakan bagian dari sistem yang bertugas untuk menerima data masukan dimana data dapat berupa asal masukan, frekuensi pemasukan data, jenis pemasukan data, dll.

5. Proses

   Merupakan bagian yang memproses masukan data menjadi informasi sesuai dengan keinginan penerima, proses dapat berupa : klarifikasi, peringkasan, pencarian, dll.

6. *Output*

   Merupakan keluaran atau tujuan akhir dari sistem, output dapat berupa laporan, grafik, dll.

7. Umpan Balik

   Merupakan elemen – elemen sistem yang tugasnya apakah sistem berjalan sesuai keinginan, umpan balik dapat berupa perbaikan, pemeliharaan, dll.

Suatu sistem dapat dikatakan sebagai kerangka terpadu yang mempunyai satu sasaran atau lebih. Sistem dapat dikatakan berhasil apabila sasaran atau tujuannya tersebut telah tercapai.

2.3 Pengertian Data dan Informasi

Data dan informasi mempunyai ruang lingkup yang sangat erat. Data dapat didefinisikan sebagai suatu bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi. Data dapat berbentuk symbol – symbol semacam huruf – huruf atau alphabet, angka – angka, bentuk suara – suara, sinyal sinyal, gambar – gambar, dan sebagainya.

Menurut Drs John Longkutoy istilah data adalah fakta atau sebagian fakta yang mengandung arti, yang dihubungkan dengan kenyataan, symbol – symbol, gambar, kata, angka atau symbol yang menunjukan suatu ide, objek, kondisi atau kondisi dan lain sebagainya. Dalam kegiatan suatu perusahaan misalnya dari hasil transaksi penjualan oleh *salesman*, dihasilkan faktur – faktur yang merupakan data dari penjualan pada suatu periode tertentu.

Faktur – faktur penjualan tersebut masih belum dapat bercerita banyak kepada manajemen. Untuk keperluan pengambilan keputusan maka faktur – faktur tersebut perlu diolah lebih lanjut untuk menjadi suatu informasi. Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian – kejadian (*event*) yang nyata digunakan untuk pengambilan keputusan. Maka dari data diatas dapat disimpulkan bahwa sumber informasi adalah data yang sudah diolah atau disiapkan dalam bentuknya dan digunakan untuk pengambilan keputusan.

2.4 Sistem Informasi

2.4.1 Definisi Sistem Informasi

Sistem Informasi diartikan sebagai suatu sistem organisasi yang merupakan kombinasi dari orang – orang, fasilitas, teknologi, media, prosedur – prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi yang penting, memproses tipe transaksi rutin tertentu memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian – kejadian internal dan external

yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan yang cerdik.

Fungsi dari Sistem Informasi adalah untuk memproses data menjadi informasi yang dibutuhkan oleh subsistem manajerial, operasional, dan lingkungan (terutama oleh subsistem manajerial dalam proses pengambilan keputusan atau "*Decision Making*").

2.4.2 Komponen Sistem

John Burch dan Griary Grudnitski mengemukakan bahwa Sistem Informasi terdiri dari komponen – komponen yang disebutnya dengan istilah blok bangunan yaitu blok masukan, blok model, blok kendali, blok keluaran, blok teknologi, blokbasis data, dan blok kendali sehingga suatu sistem, keenam blok tersebut masing masing saling berinteraksi satu dengan yang lainnya membentuk suatu kesatuan untuk mencapai sasarannya.

1. Blok Masukan

   Input mewakili data yang masuk kedalam sistem informasi, input disini termasuk metode – metode dan media untuk menagkap data yang akan dimasukan yang dapat berupa dokumen – dokumen dasar.

2. Blok Model

   Blok inbi terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan dibasis data dengan cara yang sudah tentu untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran

   Produk dari Sistem Informasi adlah keluaran yang merupakan informasi yang berkwalitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi

   Teknologi merupakan “kotak alat” dari pekerjaan sistem informasi. Teknologi digunakan untuk menerima inputan, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.

5. Blok Basis Data

   Data didalam basis data perlu diorganisasikan sedemikian rupa sehingga informasi yang dihasilkan berkwalitas organisasi. basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut DBMS.

6. Blok Kendali

   Beberapa pengendalian perlu dirancang dan diterapkan untuk menyediakan bahwa hal – hal yang dapat merusak sistem dapat dicegah, ataupun bila terlanjur terjadi kesalahan – kesalahan dapat langsung diatasi.

2.4.3 Pengertian *Inventory*

Inventory merupakan kata lain dari persediaan, istilah persediaan disini maksudnya menujukkan barang-barang yang dimiliki perusahaan. Persediaan

dapat mengambil bentuk yang tergantung pada jenis usaha yang ditekuni oleh perusahaan yang bersangkutan. Pada perusahaan yang bergerak dibidang penjualan produk, persediaan barang merupakan salah satu unsur yang paling efektif dalam operasional perusahaan, yang secara berkelanjutan digunakan dalam kegiatan penjualan barang harian yang dapat disajikan dalam bentuk laporan persediaan barang.

Menurut HAN[3] *inventory* adalah " *serangkaian kebijaksanaan dan pengendalian yang memonitor tingkat persediaan dan menentukan tingkat persediaan yang harus dijaga, kapan persediaan harus diisi, dan berapa besar pesanan harus dilakukan* ".

2.4.4 Pengertian Sistem Informasi *Inventory*

Berdasarkan dari pengertian sistem informasi dan *inventory* maka penulias dapat menarik kesimpulan bahwa pengertian dari sistem informasi *inventory* adalah sistem yang mampu memberikan layanan tentang informasi yang berupa data, dimana dalam hal ini data yang berhubungan dengan persedian barang.

2.5 Pengertian Sistem Basis Data

Basis data (*database*) adalah kumpulan dari berbagai data yang saling berhubungan satu dengan yang lainnya. Basis data tersimpan di perangkat keras, serta dimanipulasi dengan menggunakan perangkat lunak. Pendefinisian basis data meliputi spesifikasi dari tipe data, struktur dan batasan dari data atau informasi yang akan disimpan. *Database* merupakan salah satu komponen yang

penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna atau *user*. Penyusunan basis data meliputi proses memasukkan data kedalam media penyimpanan data dan diatur dengan menggunakan perangkat Sistem Manajemen Basis Data (*Database Management System DBMS*). Manipulasi basis data meliputi pembuatan pernyataan (*query*) untuk mendapatkan informasi tertentu, melakukan pembaharuan atau penggantian (*update*) data, serta pembuatan *report* data. Tujuan utama *DBMS* adalah untuk menyediakan tinjauan abstrak dari data bagi *user*. Jadi sistem menyembunyikan informasi mengenai bagaimana data disimpan dan dirawat, tetapi data tetap dapat diambil dengan efisien.

Berikut ini ada beberapa pengertian atau definisi lain dari basis data yang dikembangkan atau dasar sudut pandang yang berbeda

a. Penggunaan istilah basis data menurut Elmasri R. (1994) lebih dibatasi pada arti implisit yang khusus yaitu :
    1. Basis data merupakan penyajian suatu aspek dari dunia nyata (“real word” atau “miniword”)
    2. Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit. Sehingga data yang tekumpul secara acak dan tanpa mempunyai arti, tidak dapat disebut basis data.
    3. Basis data perlu dirancang , dibangun, dan data dikumpulkan dalam suatu tujuan. Basis data dapat digunakan oleh beberapa pemakai dan beberapa aplikasi sesuai dengan kebutuhan pemakai.

Dari keterangan diatas dapar disimpulkan bahwa basis data mempunyai berbagai sumber data dalam pengumpulan data, bervariasi derajat interaksi kejadian dari dunia nyata.

b. Penggunaan istilah basis data menurut Pustaka Fathan99 lebih dibatasi pada arti implisit yang khusus yaitu :
    1. Himpungan kelompok data (*file* atau arsip) yang saling berhubungan dan saling berorganisasi sedemikian rupa agar dapat dimanfaatkan kembali dengan cepat dan mudah.
    2. Kumpulan data yang saling berhubungan dan disimpan bersama sedemikian rupa tanpa pengulangan yang tidak perlu (*redudancy*) untuk memenuhi berbagai kebutuhan.
    3. Kumpulan *file* atau arsip atau tabel yang saling berhubungan dan disimpan di dalam media penyimpanan elektronik.

Adapun komponen – komponen sistem basis data meliputi :

1. Perangkat keras (*Hardware*) sebagai pendukung operasi pengolahan data.
2. Pengguna (*User*)
3. Sistem Operasi (*Windows XP*)
4. Sistem Pengelolaan BasisData (DBMS) (Access, SQL)
5. Program Aplikasi Lainya (VB, Delphi)

2.5.1 *Database Management System* (DBMS)

Manajemen Sistem Basis Data (*Database Managgement System* / DBMS) adalah perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan

dan utilitas kumpulan data dalam jumlah besar. DBMS dapat menjadi alternatif penggunaan secara khusus untuk aplikasi, semisal penyimpanan data dalam file dan menulis kode aplikasi yang spesifik untuk pengaturannya.

Penggunaan DBMS untuk suatu aplikasi tergantung pada kemampuan dan dukungan DBMS yang beroperasi secara efisien. Sehingga agar bisa menggunakan DBMS dengan baik, perlu diketahui cara kerja dari DBMS tersebut. Pendekatan yang dilakukan untuk menggunakan DBMS secara baik meliputi : implementasi DBMS dan arsitektur secara mendetail untuk dapat memahami desain dari suatu basis data

2.6 Konsep dasar Rekayasa Perangkat Lunak

Dalam pengembangan perangkat lunak terdapat proses didalam pengembangannya, mungkin kita bertanya apakan sebenarnya perangkat lunak itu. Dalam bukunya Roger S. Pressman, Ph.D mendefinisikan sebagai berikut :

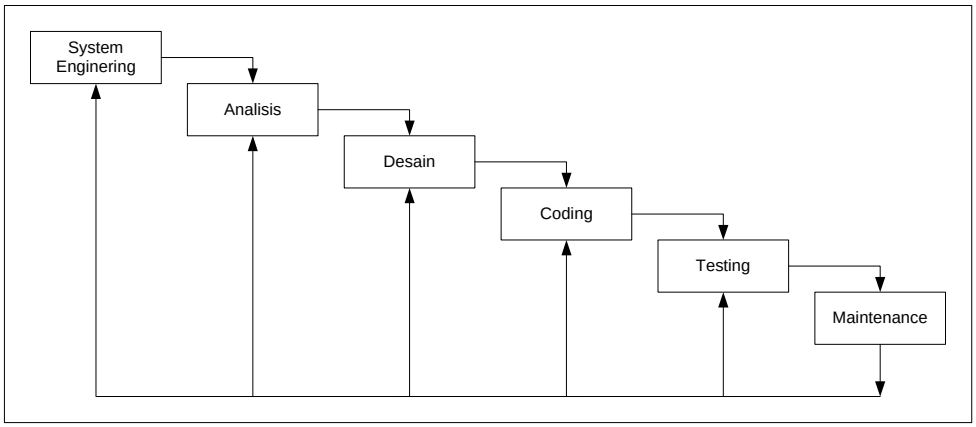
"*(1) perintah (proses computer) yang bila dieksekusi memberikan fungsi dan unjuk kerja yang diinginkan, (2) struktur data yang memungkinkan program memanipulasi informasi secara profesional, dan (3) Dokumen yang menggambarkan operasi dan kegunaan program. Proses perangkat lunak adalah sebuah kerangka kerja untuk tugas – tugas yang dibutuhkan untuk membangun perangkat lunak dengan kualitas tinggi.*

2.6.1 Model Proses Perangkat Lunak

Dalam tahap perancangan suatu sistem diperlukan adanya teknik-teknik penyusunan sistem untuk menganalisa dan mendokumentasikan data yang mengalir didalam sistem tersebut

2.6.2 Model Proses *Waterfall*

Model proses *waterfall* ialah model proses pembuatan sebuah *system* yang terstruktur yang amat mudah dimengerti, seperti yang telah diketahui bahwa dengan model proses *waterfall* sebuah system dapat dikembangkan lagi. Seperti yang diutarakan oleh PRESS [5] " *waterfall adalah pendekatan orientasi objek ke pengembangan perangkat lunak yang tersruktur dan saling berhubungan antara satu tahap dan tahap lainnya untuk mendapatkan hasil maksimal"*



Gambar 2.2 *Model Sekuensial Linear (Waterfall)*

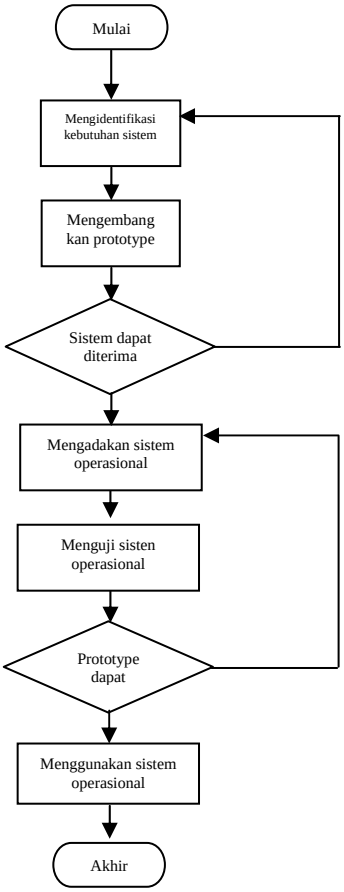
2.6.3 Model Proses *Prototipe*

Metode yang digunakan untuk membangun sistem informasi yang terintegrasi yaitu metode *prototipe*. "*Prototipe adalah Model atau simulasi yang akan dikembangkan*"PRESS [5].

Langkah-langkah yang terdapat *prototipe* Jenis II adalah sebagai berikut :

1. Mengidentifikasi kebutuhan pemakai. Analisis sistem mewawancarai pemakai untuk mendapatkan gagasan dari apa yang diinginkan pemakai terhadap sistem.
2. Mengembangkan *Prototipe*. Analisis sistem dapat bekerjasama dengan spesialis informasi lain, menggunakan satu atau lebih peralatan *prototyping* untuk mengembangkan sebuah *prototipe*.
3. Menentukan apakah *Prototipe* dapat diterima. Pemakai memberikan masukan kepada analis apakah *prototipe* memuaskan. Jika ya, langkah 4 akan diambil; jika tidak, *prototipe* direvisis dengan mengulangi langkah 1, 2, dan 3 dengan pengertian yang lebih baik mengenai kebutuhan pemakai.
4. Mengkodekan sistem operasional. *Programmer* menggunakan *prototipe* sebagai dasar untuk pengkodean (*coding*) sistem operasional.
5. Menguji sistem operasional. Programmer menguji sistem.
6. Menentukan jika sistem operasional dapat diterima. Pemakai memberikan masukan kepada analis apakah sistem dapat diterima. Jika ya, langkah 7 dilakukan; jika tidak, langkah 4 dan 5 diulangi.
7. Menggunakan sistem operasional

Berikut ini gambar *Prototipe* Jenis II :



Gambar 2.3 Model *Prototipe*

2.7 Rekayasa Perangkat Lunak

Definisi perangkat lunak yang diusulkan oleh Fritz Bauner adalah :

"*Rekayasa perangkat lunak adalah pengembangan dan penggunaan prinsip pengembangan untuk memperoleh perangkat lunak secara ekonomis yang realiabel dan bekerja secara efisien pada mesin nyata.*"

Internasional (IEEE) telah mengembangkan definisi yang lebih komperehensif sebagai berikut :

"Rekayasa perangkat lunak: (1) Aplikasi dari sebuah pendekatan kuantifiabel, disiplin, dan sistematis kepada pengembangan, operasi, dan pemeliharaan perangkat lunak; yaitu aplikasi dari rekayasa perangkat lunak, (2) Studi tentang Pendekatan – pendekatan seperti pada butir "

2.7.1 Diagram Konteks (*Context Diagram*)

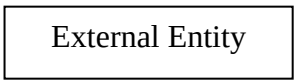
Diagram konteks adalah arus data yang berfungsi untuk menggambarkan keterkaitan aliran – aliran data sistem dengan bagian – bagian luar (kesatuan luar) Kesatuan luar ini merupakan sumber arus data atau tujuan data yang berhubungan dengan sistem informasi tersebut.

"*Model sistem pokok (fundamental system model) mewakili elemen software dengan masukan (input) dan keluaran (output) yang diindikasikan dengan anak panah masuk dan keluar memperlihatkan suatu hubungan antara sistem dengan lingkungannya yang menjadi sumber data"*.

2.7.2 *Data Flow Diagram* (DFD)

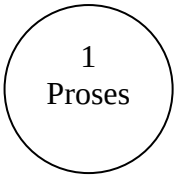
Diagram aliran data atau *data flow diag*ram (DFD) menurut JOG [1] adalah "*sebuah teknik grafis yang menggambarkan aliran informasi yang diaplikasikan pada saat bergerak dari input menjadi output. Data flow diagram merupakan gambaran secara logika dan tidak tergantung pada hardware, software, struktur data, ataupun organisasi file yang digunakan*".

Beberapa simbol yang dipergunakan untuk menggambarkan sistem antara lain:



External Entity, merupakan simbol yang mewakili elemen yang berada di luar sistem, tetapi memiliki hubungan interaksi dengan sistem.

Gambar 2.4 External entity



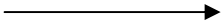
Proses, merupakan simbol yang mewakili kegiatan untuk mengubah data menjadi informasi atau masukan untuk data lain.

Gambar 2.5 Proses



Data Store, merupakan simbol yang mewakili tempat penyimpanan dari data dan dapat dipergunakan bila dibutuhkan.

Gambar 2.6 *Data Store*



Data flow, merupakan simbol yang mewakili arah aliran data yang berasal dari satu proses, *data store* atau elemen lingkungan menuju ke satu proses, *data store* atau *enviromental element* lainnya.

Gambar 2.7 *Data Flow*

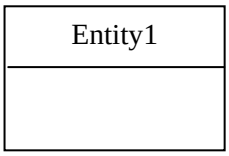
Permodelan sistem dengan menggunakan DFD memiliki *level* yang menandai penggambaran sistem mulai dari umum hingga detail. *Level* yang menandai hubungan yang paling umum dan menggambarkan sistem secara

keseluruhan disebut *context diagram*. Dari *context diagram* selanjutnya dapat diuraikan ke dalam DFD *level* 0, 1, hingga *level* dimana proses dianggap cukup detail dan jelas serta tidak bisa diuraikan lebih detail.

Dalam penggambaran proses, perlu diberikan penomoran, demikian juga dengan *level* uraian dari proses tersebut. Contohnya proses dengan nomor "1" pada DFD *level* "0" bila diuraikan pada DFD *level* "1", maka penomoran proses perlu didahului dengan nomor proses yang diuraikan, yaitu "1.1", "1.2", "1.3" dan seterusnya. Hal yang sama juga akan berlaku pada penomoran DFD *level* selanjutnya.

2.7.3 *Entity Relationship Diagram* (ERD)

ERD merupakan *data modeling* yang dipergunakan untuk mendokumentasikan data sistem. Dokumentasi dilakukan dengan cara menentukan kelompok data apa saja yang terdapat dalam tiap *entity* dan bagaimana hubungan antara *entity* satu dengan lainnya. *Entity* dapat berupa *environmental element*, *resource* dan transaksi yang sangat diperlukan dan didokumentasikan dalam bentuk data. Simbol-simbol yang dipergunakan dalam *ERD* antara lain: ***Data flow***, merupakan simbol yang mewakili arah aliran data yang berasal dari satu proses, *data store* atau elemen lingkungan menuju ke satu proses, *data store* atau *enviromental element* lainnya.



Entity, yang termasuk ke dalam *entity* antara lain *environmental element*, *resource*, dan transaksi yang sangat penting bagi sistem. Setiap *entity* memiliki *attribute* merupakan karakteristik yang menandai atau dimiliki oleh *entity*. Gambar 2.8 *Entity*

Relationship, menandakan hubungan diantara *entity*, simbol *relationship* kadang diwakili sebagai garis yang menghubungkan entitiy.

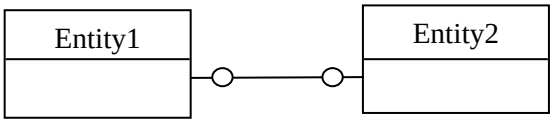
Gambar 2.9 *Relationship*

Selain istilah dan simbol di atas, terdapat juga istilah yang menggolongkan jenis relasi yang dilakukan antar *entity*, yaitu:

a. *Cardinality*, menandai jumlah *entity* yang muncul dalam relasi dengan *entity* lainnya. Nilai *cardinality* ada dua yaitu “1” atau “*many*”. Bentuk relasi yang dapat dihasilkan ada tiga yaitu :

1. Satu ke Satu (*One To One Relationship*)

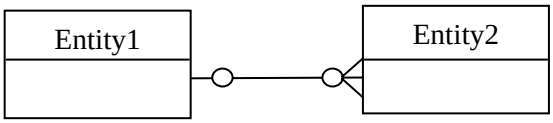
   Hubungan antara entity pertama dengan entity kedua adalah satu berbanding satu. (Lihat Gambar 2.10).



Gambar 2.10 *One to One Relationship*

2 Satu ke Banyak (*One To Many*)

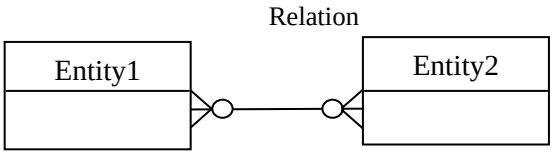
Himpunan entity pertama dapat berhubungan satu atau lebih entity pada himpunan kedua, tetapi tidak sebaliknya. (Lihat Gambar 2.11).



Gambar 2.11 *One to Many Relationship*

2. Banyak ke Banyak (*Many To Many*)

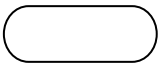
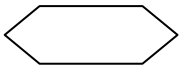
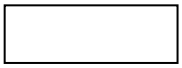
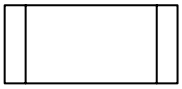
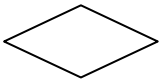
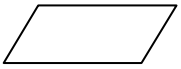
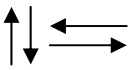
Setiap entity pada himpunan entity pertama dapat berhubungan dengan banyak entity pada himpunan entity kedua, begitu pula sebaliknya. (Lihat Gambar 2.12).



Gambar 2.12 *Many to Many Relationship*

2.7.4 *Flowchart*

Merupakan diagram yang menggambarkan langkah dari sebuah proses. Dengan menggunakan *flowchart* maka kerja dari suatu proses dapat dipantau untuk pengembangan atau perbaikan. Beberapa simbol yang umum dipergunakan dapat dilihat pada Gambar 2.13.

SIMBOL	KETERANGAN
	**Terminal Point Symbol.** digunakan untuk menunjukkan awal dan akhir dari suatu proses.
	**Preparation Symbol.** Simbol **Persiapan** digunakan untuk memberi nilai awal Suatu besaran atau *variabel* (harga awal).
	**Process Symbol.** Simbol **Proses** atau **Pengolahan** digunakan untuk mewakili suatu proses, seperti pengolahan aritmatika atau pemindahan data.
	**Predefined Process Symbol.** Simbol **Proses Terdefenisi** digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain atau untuk proses yang detilnya dijelaskan terpisah, misalnya berbentuk subroutine.
	**Decision Symbol.** Simbol **Keputusan** digunakan untuk mewakili operasi perbandingan logika atau suatu penyeleksian kondisi di dalam program.
	**Input/Output Symbol.** Simbol **Input/Output** digunakan untuk menyatakan dan mewakili data masukan atau keluaran.
	**Connector Symbol.** Simbol **Penghubung** digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama.
	**Off-page Connector.** Simbol **Penghubung Halaman lain** digunakan untuk menunjukkan hubungan arus proses yang terputus dengan sambungannya ada di halaman yang lain.
	**Flow Lines Symbol.** Simbol **Garis Alir** digunakan untuk menunjukkan aliran atau arus dari proses.

Gambar 2.13 Simbol *Flowchart*

2.7.5 Kamus Data

Kamus data (*data dictionary*) menurut JOG [6] *" merupakan catalog data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi untuk meendefinisikan data yang mengalir disistem dengan lengkap"* .

Kamus data dan komponen – komponen lainnya yang dikumpulkan saat analisa sistem sangat dibutuhkan dalam perancangan sistem dan dapat juga dipergunakan oleh programmer yang membutuhkan penjelasan tambahan, sedangkan kamus data saat desain sistem merupakan petunjuk penting untuk pembuatan sistem.

Selain dapat menjelaskan suatu model, kamus data juga berfungsi untuk menghindari penggunaan kata yang sama, karena kamus data disusun menurut abjad.

1. Notasi Kamus Data

   Notasi kamus data lebih mudah menjelaskan data dibanding dengan narasi.

2. Notasi Struktur Data

   Digunakan untuk membuat spesifikasi elemen data. Notasi yang umum digunakan adalah : (Lihat Tabel 2.2)

3. Notasi Tipe Data

   Digunakan untuk membuat spesifikasi format input maupun output suatu data. Notasi yang umum digunakan adalah : (Lihat Tabel 2.1)

Tabel 2.1 Notasi Tipe Data

Notasi	**Keterangan**
X	Setiap Karakter
9	Angka Numerik
A	Karakter Alfabet
Z	Angka nol ditampilkan sebagai spasi kosong
.	Titik, sebagai pemisah ribuan
,	Koma, sebagai pemisah pecahan
-	Tanda Penghubung (contoh : 022—2510884)
/	Tanda Pembagi (Contoh : 03/06/2009)

Tabel 2.2 Notasi Struktur Data

Notasi	**Keterangan**
=	Terdiri dari
+	*And* (dan)
( )	Pilihan (boleh ada atau tidak)
{ }	Iterasi / Pengulangan
[ ]	Pilih salah satu pilihan
1	Pemisah pilihan didalam tanda [ ]
*	Keterangan / Catatan
@	Petunjuk (*Key Field*)

2.8 Sistem Komputer

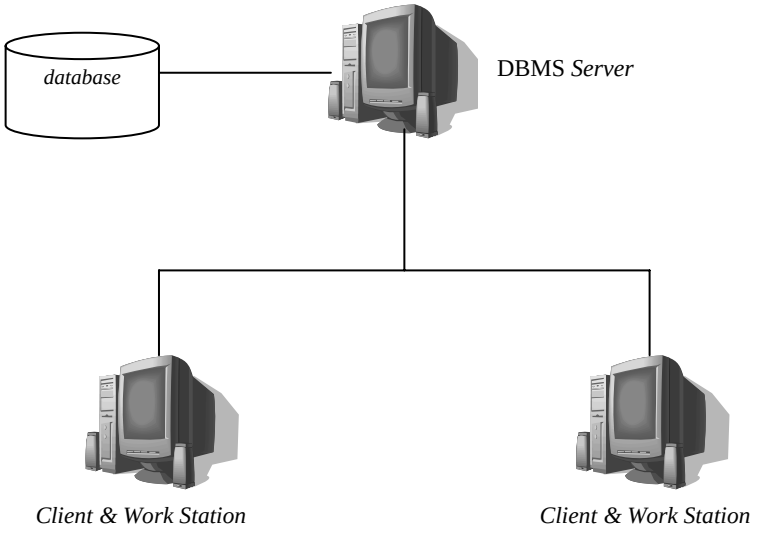
Sistem yang berbasis komputer sangat diperlukan sekali dalam pengolahan data untik menghasilkan informasi yang dibutuhkan. Elemen yang mendukung dalam sistem komputer yaitu :

1. *Hardware* (perangkat keras), yaitu peralatan yang ada pada sistem komputer secara fisik.
2. *Software* (perangkat lunak), yaitu program-program yang berisis perintah-perintah untuk melakukan pengolahan data, dimana program dibuat sesederhana mungkin.
3. *Brainware* yaitu manusia yang dapat mengoprasikan, serta mengatur sistem komputer.

2.8.1 Sistem *Client Server*

Sistem *client-server* atau disebut dengan sistem sentralisasi yang diterapkan dalam sebuah jaringan. *Client-server* terdiri dari dua komponen, yaitu *client* dan *server*. Adapun pengertian client server menurut WIL[7] " *model konektivitas LAN (Local Area Network) yang mengenal adanya client dan server dimana masing-masing memiliki fungsi yang berbeda satu sama lain, dan dalam pelaksanaan tugasnya server sebagai pembagi data pada client* ".

Client berisisi aplikasi basis data dan *server* berisi DBMS dan basis data. Pada sistem *Client Server* untuk memenuhi kebutuhan, *client* akan mengirimkan *message* (perintah) *query* pengambilan data. Selanjutnya, *server* yang menerima *message* tersebut akan menjalankan *query* tersebut dan hasilnya akan dikirimkan kembali ke *client.* Dengan begitu transfer datanya jauh lebih efisien. Adapun bentuk dari Sistem *Client Server* yang sederhan adalah sebagai berikut :



Gambar 2.14 *Sistem Client Server Sederhana*

2.8.2 Topologi *Star*

Topologi *star* adalah topologi dimana " *beberapa node dihubungkan dengan suatu node pusat (central node) yang membentuk seperti bintang, dimanan semua komunikasi diatur lansung oleh central node, yang manna central node ini mengatur semua arus keluar masuk informasi.* " JOG[1]

Topologi ini mudah dikembangkan, baik untuk penambahan maupun pengurangan terminal. Banyak terminal yang dapat terhubung tergantung pada jumlah port yang tersedia pada hub yang digunakan. Pada *port star* ini, hub yang digunakan akan menjadi titik kritis, sehingga perlu adanya perhatian dan pemeliharaan terhadap hub tersebut.

2.9 Perangkat Lunak Pendukung

Berisi tentang teori singkat mengenai software pembangun sistem yang dipergunakan.

2.9.1 *Structure Query Language* (SQL)

SQL merupakan bahasa yang memungkinkan pemakai untuk melakukan akses data ke *DBMS* dan pemakai dapat menggambarkan data yang ingin diperoleh (Hoffman, 2001). SQL juga memungkinkan pemakai untuk melakukan manipulasi data (*input*, *edit*, dan *delete*). Dalam perkembangannya, SQL telah menjadi bahasa standar dalam kegiatan manipulasi data pada *database* dari berbagai *vendor* yang ada di dunia. Pada subbab ini akan diberikan penjelasan tentang dasar penggunaan dari perintah SQL.

Beberapa Perintah yang umum dipergunakan dalam SQL adalah sebagai berikut:

1. SELECT, merupakan perintah untuk melakukan pengambilan data yang berada dalam tabel
2. Conditional, merupakan perintah untuk mengambil data dengan kondisi tertentu, menggunakan “WHERE” dan relational operator seperti “<”, “> “,”=”, ”<>”.
3. Joins, merupakan bentuk perintah untuk melakukan pengambilan data dari dua atau lebih tabel sekaligus.
4. *Aggregate Function* seperti *Sum(), Count(), Avg(), Max(), Min()*. Bila ingin mengetahui berapa jumlah “id_barang” dengan id_kelompok tertentu pada tabel barang.
5. *CREATE TABLE*, perintah ini dipergunakan untuk membuat sebuah tabel baru dengan *field*-nya.
6. *ALTER TABLE*, perintah ini berfungsi melakukan perubahan pada struktur tabel (bukan isi tabel).
7. *INSERT*, merupakan perintah yang dipergunakan untuk menambahkan data pada tabel.
8. *UPDATE*, berfungsi untuk mengubah data pada tabel.
9. *DELETE*, merupakan perintah untuk menghapus data pada tabel baik satu atau seluruhnya. Untuk menghapus data tertentu maka dapat dipergunakan *WHERE* sedangkan untuk menghapus seluruh isi tabel maka *WHERE* tidak dipergunakan lagi.

2.9.2 *Visual Basic*

Microsoft Visual Basic adalah bahasa pemrograman yang digunakan untuk membuat palikasi *Windows* yang berbasis grafis (GUI – *Graphical User Interface*). Visual Basic merupakan *event-driven programming* (pemrograman terkendali kejadian) artinya program menunggu sampai adanya respon dari pemakai berupa *event*/kejadian tertentu (tombol diklik, menu dipilih, dan lain-lain). Ketika *event* terdeteksi, kode yang berhubungan dengan *event* (prosedur *event*) akan dijalankan.

2.9.2.1 Sejarah Singkat Visual Basic

Berikut ini adalah point-point penting dalam sejarah perkembangan Visual Basic, sebagai berikut :

1. Visual Basic pertama kali diperkenalkan tahun 1991.
2. Berikutnya dirilis Visual Basic 3.0 tahun 1993.
3. Visual Basic 4.0 dirilis pada akhir tahun 1995 (tambahan dukungan untuk aplikasi 32 bit).
4. Visual Basic terbaru adalah versi 6.0 yang dirilis pada akhir tahun 1998.

   Microsoft umumnya membuat tiga edisi Visual Basic yaitu :

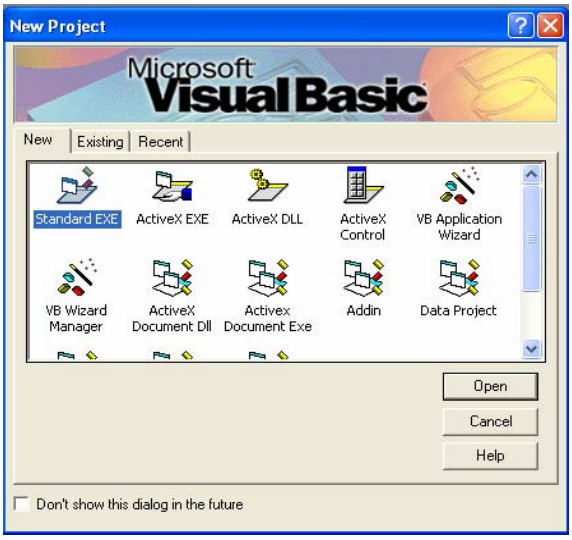
   a. *Standard Edition* merupakan produk dasar.
   b. *Profesional Edition* berisi tambahan *Microsoft Jet Data Access Engine* (*database*) dan pembuatan *server OLE automation*.
   c. *Enterprise Edition* adalah edisi *client-server*.

2.9.2.2 IDE Visual Basic

Langkah awal dari belajar Visual Basic adalah mengenal IDE (Integrated Developement Environment) Visual Basic yang merupakan Lingkungan Pengembangan Terpadu bagi programmer dalam mengembangkan aplikasinya. Dengan menggunakan IDE programmer dapat membuat user interface, melakukan koding, melakukan testing dan debuging serta menkompilasi program menjadi executable. Penguasaan yang baik akan IDE akan sangat membantu programmer dalam mengefektifkan tugas-tugasnya sehingga dapat bekerja dengan efisien

Sesaat kita aktif di IDE Visual Basic, maka kita akan dihadapkan kepada suatu pilihan terhadap jenis Project yang ingin kita buat sebagaimana yang ditunjukkan oleh Gambar 2.15

Sebagai langkah awal dari proses belajar adalah memilih project Standard EXE.



Gambar 2.15 Dialog box New Visual Basic 6.

2.9.2.3 Jendela IDE

IDE *Visual Basic 6* menggunakan model MDI (*Multiple Document Interface*). Berikut ini adalah gambar yang menunjukan bagian-bagian dan nama-nama jendela yang dapat tampil pada IDE *Visual Basic*. Mungkin pada IDE anda hanya ditampilkan sebagian jendela dibandingkan dengan Gambar 2.16 anda tidak perlu terkejut, karena penampilan dari jendela - jendela tersebut dapat diatur dengan perintah pada menu *View*. Sebagai langkah awal dari proses belajar, tidak semua jendela akan kita gunakan, tetapi hanya beberapa yang penting, sedangkan yang lainnya bersifat khusus.

Adapun jendela - jendela yang perlu kita perhatikan adalah sebagai berikut:

a. **Menu Bar**, digunakan untuk memilih tugas-tugas tertentu seperti menyimpan project, membuka project, dll
b. **Main Toolbar**, digunakan untuk melakukan tugas-tugas tertentu dengan cepat.
c. **Jendela Project**, jendela ini berisi gambaran dari semua modul yang terdapat dalam aplikasi anda. Anda dapat menggunakan icon Toggle Folders untuk menampilkan modul-modul dalam jendela tersebut secara di group atau berurut berdasarkan nama. Anda dapat menggunakan Ctrl+R untuk menampilkan jendela project, ataupun menggunakan icon Project Explorer.
d. **Jendela Form Designer**, jendela ini merupakan tempat anda untuk merancang user interface dari aplikasi anda. Jadi jendela ini menyerupai kanvas bagi seorang pelukis.

e. **Jendela Toolbox**, jendela ini berisi komponen-komponen yang dapat anda gunakan untuk mengembangkan user interface.

f. **Jendela Code**, merupakan tempat bagi anda untuk menulis koding. Anda dapat menampilkan jendela ini dengan menggunakan kombinasi Shift-F7.

g. **Jendela Properties**, merupakan daftar properti-properti object yang sedang terpilih. Sebagai contohnya anda dapat mengubah warna tulisan (foreground) dan warna latarbelakang (background). Anda dapat menggunakan F4 untuk menampilkan jendela properti.

h. **Jendela Color Palette**, adalah fasilitas cepat untuk mengubah warna suatu object.

i. **Jendela Form Layout**, akan menunjukan bagaimana form bersangkutan ditampilkan ketika runtime.



j. **Pointer** bukan merupakan suatu kontrol; gunakan icon ini ketika anda ingin memilih kontrol yang sudah berada pada form.

k. **PictureBox** adalah kontrol yang digunakan untuk menampilkan image dengan format: BMP, DIB (bitmap), ICO (icon), CUR (cursor), WMF (metafile), EMF (enhanced metafile), GIF, dan JPEG.

Gambar 2.16 Komponen IDE

l. **Label** adalah kontrol yang digunakan untuk menampilkan teks yang tidak dapat diperbaiki oleh pemakai.

m. **TextBox** adalah kontrol yang mengandung string yang dapat diperbaiki oleh pemakai, dapat berupa satu baris tunggal, atau banyak baris.

n. **Frame** adalah kontrol yang digunakan sebagai kontainer bagi kontrol lainnya.

o. **CommandButton** merupakan kontrol hampir ditemukan pada setiap form, dan digunakan untuk membangkitkan event proses tertentu ketika pemakai melakukan klik padanya.

p. **CheckBox** digunakan untuk pilihan yang isinya bernilai yes/no, true/false.

q. **OptionButton** sering digunakan lebih dari satu sebagai pilihan terhadap beberapa option yang hanya dapat dipilih satu.

r. **ListBox** mengandung sejumlah item, dan user dapat memilih lebih dari satu (bergantung pada property MultiSelect).

s. **ComboBox** merupakan konbinasi dari TextBox dan suatu ListBox dimana pemasukkan data dapat dilakukan dengan pengetikkan maupun pemilihan.

t. **HScrollBar dan VScrollBar** digunakan untuk membentuk scrollbar berdiri sendiri.

u. **Timer** digunakan untuk proses background yang diaktifkan berdasarkan interval waktu tertentu. Merupakan kontrol non-visual.

v. **DriveListBox, DirListBox, dan FileListBox** sering digunakan untuk membentuk dialog box yang berkaitan dengan file.

w. **Shape dan Line** digunakan untuk menampilkan bentuk seperti garis, persegi, bulatan, oval.

x. **Image** berfungsi menyerupai image box, tetapi tidak dapat digunakan sebagai kontainer bagi kontrol lainnya. Sesuatu yang perlu diketahui bahwa kontrol image menggunakan resource yang lebih kecil dibandingkan dengan PictureBox

y. **Data** digunakan untuk data binding

z. **OLE** dapat digunakan sebagai tempat bagi program eksternal seperti Microsoft Excel, Word, dll.

2.9.2.4 Jenis *File* Pada Visual Basic

Proyek merupakan cikal bakal dari aplikasi dan terdiri dari beberapa *file* yang saling terkait satu sama lain, *file-file* tersebut mempunyai ekstensi yang berbeda-beda sesuai dengan isi dan fungsinya. Sebuah *Proyek* dapat terdiri atas:

a. File. VBF untuk tiap proyek

b. File FRM untuk tiap form aplikasi pada sebuah proyek.

c. File. VBX atau OCX untuk tiap custom pada sebuah proyek.

d. File. BAS untuk tiap modul standar pada proyek.

e. File CLS untuk tiap modul class pada sebuah proyek.

f. File. FRX untuk tiap form pada sebuah proyek yang memiliki control berproperti menggunakan data *binary*.

g. File. RES yaitu sebuah *resource file* pada sebuah proyek.

Dalam hal ini tidak perlu membuat seluruh *file* di atas. Karena apa saja yang dibutuhkan tergantung dari proyek yang sedang dikerjakan.

2.9.2.5 Komponen Pada Visual Basic

Komponen-komponen yang sering dipakai diantaranya:

a. Modul *Form* (File berekstensi .FRM).

   Merupakan kumpulan *form* yang berisi deskripsi secara grafis mengenai tampilan serta kontrol yang digunakan dalam sebuah proyek. Termasuk di dalamnya pengaturan properti yang digunakan oleh masing-masing kontrol. Disampimg itu rutin yang ditulis untuk masing-masing *obyek* pada *form* juga tersimpan pada *file* ini. Untuk aplikasi yang terdiri dari banyak *form*, masing-masing *form* tersimpan dalam sebuah *file* berekstensi.FRM.

b. Modul *Class* (File Berekstensi .CLS)

   Fungsinya sama dengan modul *form* tetapi terlihat dalam bentuk tampilan. Melalui modul ini bisa dicoba membuat sendiri obyek yang ingin ditampilkan. Masing-masing obyek yang dibuat harus mempunyai properti dan modul tersendiri.

c. Modul Standar (*File* berekstensi. BAS).

   Modul ini berisi deklarasi tipe, konstanta, variabel eksternal *procedure* (prosedur yang menangani kejadian) dan *public procedure*. Biasanya *fil*e ini berisikan perhitungan atau logika yang kompleks.

d. *File Recourse*(File berekstensi.RES).

   File ini berisi *bitmap*, text string, atau data lainnya yang dapat diubah tanpa perlu mengedit kembali rutin. Biasanya *file* ini digunakan untuk

mengubah bahasa pada user *interface* alokasi untuk pemakai berbahasa asing (selain yang digunakan oleh si pembuat aplikasi).

e. *OLE Custom Control* dan *Insertable Objects* (File berekstensi. OCX)

   File ini berisi *custom* dan *Insertable Objects* yang ditambahkan pada proyek anda.

f. VBX *Custom Controls* (File berekstensi. VBX).

   Sama dengan *file* berekstensi .OCX namun hanya digunakan pada Visual Basic 4.0 versi 16 *bit*.