



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Asropudin (2013:19) menjelaskan bahwa “Komputer adalah alat bantu pemrosesan data secara elektronik dan cara pemrosesan datanya berdasarkan urutan instruksi atau program yang tersimpan dalam memori masing-masing”.

Sutarman (2012:3) menjelaskan bahwa “Komputer adalah alat elektronik yang dapat melakukan perhitungan numerik, dapat membaca *input* data dan mengolahnya sesuai dengan program yang ditetapkan untuk menghasilkan sebuah output, alat yang digunakan untuk menyimpan data, serta bekerja secara otomatis sesuai dengan perintah yang diterapkan dalam program”.

Jadi komputer merupakan sebuah alat bantu dalam mengolah data elektronik yang dapat melakukan pekerjaan melalui instruksi-instruksi tertentu yang akan menghasilkan informasi dan mempermudah dalam membantu pengolahan data.

2.1.2. Pengertian Program

Sutarman (2012:03) menjelaskan bahwa “Program adalah barisan perintah/instruksi yang disusun sehingga dapat dipahami oleh komputer dan kemudian dijalankan sebagai barisan perhitungan numerik, di mana barisan perintah tersebut berhingga, berakhir, dan menghasilkan *output*”.

2.1.3. Pengertian Sistem

Sutarman (2012:05) menjelaskan bahwa “Sistem adalah kumpulan elemen yang saling berhubungan dan berinteraksi dalam satu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama”.

2.1.4. Pengolahan Data

Sutarman (2012:04) menjelaskan bahwa “Pengolahan data adalah proses perhitungan/ transformasi data *input* menjadi informasi yang mudah dimengerti ataupun sesuai dengan yang diinginkan”.

Ladjamudin (2013:09) menjelaskan bahwa “Pengolahan data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan”.

2.1.5. Siklus Pengolahan Data

Sutarman (2012:04) menjelaskan bahwa ”Suatu proses pengolahan data terdiri atas tiga tahapan dasar, yang disebut dengan Siklus Pengolahan Data, yaitu *input*, *processing*, dan *output*”.



Gambar 2.1 Siklus Pengolahan Data

2.2. Teori Judul

2.2.1. Pengertian Aplikasi

Asropudin (2013:06) menjelaskan bahwa “Aplikasi adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word* dan *Microsoft excel*”.

2.2.2. Pengertian Pengolahan Data

Sutarman (2012:04) menjelaskan bahwa ”Pengolahan data adalah proses perhitungan/ transformasi data *input* menjadi informasi yang mudah dimengerti ataupun sesuai dengan yang diinginkan”.

Ladjamudin (2013:09) menjelaskan bahwa “Pengolahan data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan”.



2.2.3. Pengertian Data

Asropudin (2013:22) menjelaskan bahwa “Data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti. Data dapat diolah sehingga menghasilkan informasi”.

Ladjamudin (2013:20) menjelaskan bahwa “Data merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi”.

2.2.4. Aplikasi Pengolahan Data Barang Pada PT. Jasa Raharja (Persero) Sumatera Selatan

Aplikasi Pengolahan Data Kasus Pidana Umum pada Kejaksaan Tinggi Sumatera Selatan adalah suatu aplikasi yang digunakan untuk mengelola data-data mengenai kasus pidana umum pada Kejaksaan Tinggi Sumatera Selatan yang dapat mempermudah dalam proses pengolahan data kasus pidana umum.

2.3. Teori Khusus

2.3.1. Pengertian *Database*

Kadir (2008:3) menjelaskan bahwa “Secara umum, *database* berarti koleksi data yang terkait. Secara praktis, basis data dapat dianggap sebagai suatu penyusun data yang terstruktur yang disimpan dalam media pengingat (*hard disk*) yang tujuannya adalah agar data tersebut dapat diakses dengan mudah dan cepat”.

2.3.2. Pengertian Kamus Data

Ladjamudin (2013:70) menjelaskan bahwa “Kamus data berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan dalam sistem secara persis sehingga pemakain dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses”.

2.3.3. Data Flow Diagram

Ladjamudin (2013:64) menjelaskan bahwa “Data Flow Diagram merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. Salah satu keuntungan menggunakan diagram aliran data adalah memudahkan pemakai atau *user* yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan”.

2.3.3.1. Diagram Konteks

Ladjamudin (2013:64) menjelaskan bahwa “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”.

2.3.3.2. Diagram Nol/Zero

Ladjamudin (2013:64) menjelaskan bahwa “Diagram nol adalah diagram yang menggambarkan proses dari data flow diagram. Diagram nol memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, menunjukan tentang fungsi-fungsi utama atau proses yang ada, aliran data dan eksternal *entity*”.

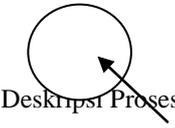
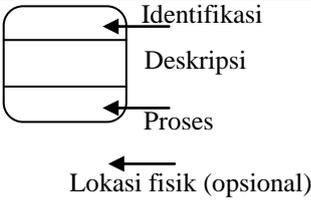
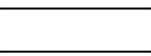
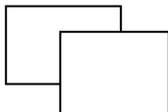
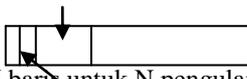
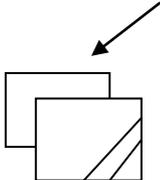
2.3.3.3. Diagram Rinci

Ladjamudin (2013:64) menjelaskan bahwa “Diagram rinci adalah diagram yang menguraikan proses apa yang ada dalam diagram *zero* atau diagram level di atasnya”.

Tabel 2.1 Simbol-simbol data flow diagram

NO	Nama Simbol	Simbol	Keterangan
1	Arus Data		

Lanjutan Table 2.1 Simbol-simbol data flow diagram

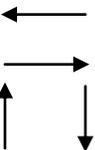
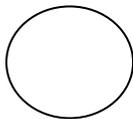
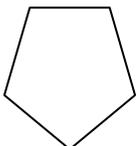
NO	Nama Simbol	Simbol	Keterangan
2	Proses		
3	Penyimpanan Data		
4	Entitas Luar		
5	Arus Material		
6	Penyimpanan Data yang ditunjukkan berulang kali pada satu diagram		Identifikasi  N baris untuk N pengulangan (tidak termasuk yang pertama)
7	Simpanan luar yang ditunjukkan berulang kali pada satu diagram		

Sumber : Ladjamudin (2013:72)

2.3.4. Pengertian Flowchart

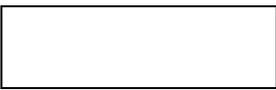
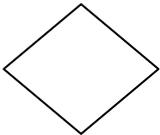
Ladjamudin (2013:263) menjelaskan bahwa “Flowchart adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Flowchart merupakan cara penyajian dari suatu algoritma”.

Table 2.2 *Flow Direction Symbols*

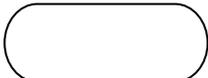
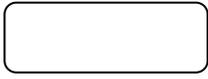
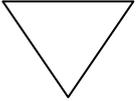
No	Simbol	Keterangan
1		Simbol arus / flow Untuk mengatur jalannya arus suatu proses
2		Simbol Communication link Untuk menyatakan bahwa adanya transisi suatu data/informasi dari satu lokasi ke lokasi lainnya
3		Simbol Connector Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang sama
4		Simbol Offline Connector Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda

Sumber : Al-Bahra (2013:266)

Table 2.3 *Processing Symbols*

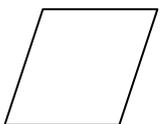
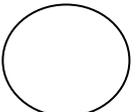
No	Simbol	Keterangan
1		Simbol Offline Connector Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda
2		Simbol Manual Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer (manual)
3		Simbol Decision/logika Untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya/tidak

Lanjutan Table 2.3 *Processing Symbols*

No	Simbol	Keterangan
4		Simbol <i>Predefined Proses</i> Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
5		Simbol Terminal Untuk menyatakan permulaan atau akhir suatu program
6		Simbol <i>Keying Operation</i> Untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard
7		Simbol <i>Off-line Storage</i> Untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
8		Simbol Manual input Untuk memasukkan data secara manual dengan menggunakan <i>online keyboard</i>

Sumber : Ladjamudin (2013:267)

Table 2.4 *Input-output Symbols*

No	Simbol	Keterangan
1		Simbol <i>Input-output</i> Untuk menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya
2		Simbol <i>Punched Card</i> Untuk menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu
3		Simbol <i>Magnetic-tape unit</i> Untuk menyatakan <i>input</i> berasal dari <i>pita magnetic</i> atau <i>output</i> disimpan ke <i>pita magnetic</i>

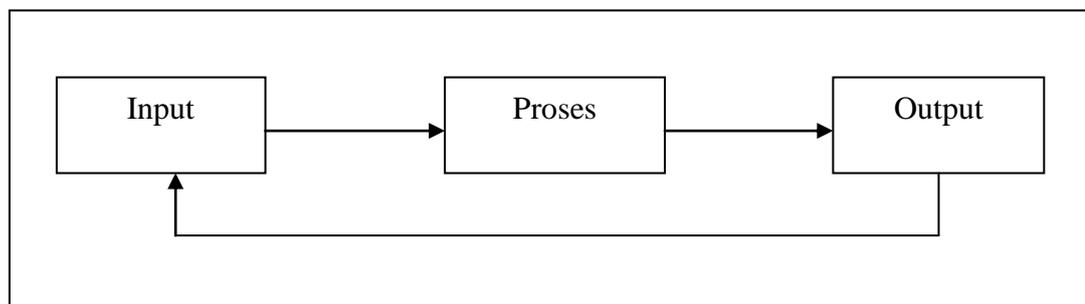
Lanjutan Table 2.4 *Input-output Symbols*

No	Simbol	Keterangan
4		Simbol Disk storage Untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i>
5		Simbol Document Untuk mencetak laporan ke <i>printer</i>
6		Simbol Display Untuk menyatakan peralatan <i>output</i> yang digunakan berupa layar (video, komputer)

Sumber : Ladjamudin (2013:268)

2.3.4.1. Kaidah-Kaidah Pembuatan Flowchart

Dalam pembuatan *flowchart* tidak ada rumus atau kaidah baku yang bersifat mutlak. Karena *flowchart* merupakan gambaran hasil pemikiran dalam menganalisa suatu masalah dengan komputer, sehingga *flowchart* yang dihasilkan dapat bervariasi antara satu pemrogram dengan pemrogram lainnya. Setiap pengolahan selalu terdiri dari 3 (tiga) bagian utama, yaitu:



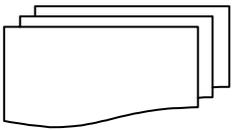
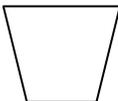
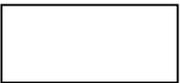
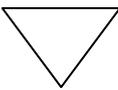
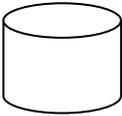
Gambar 2.2 alur pengolahan data pada *flowchart*

2.3.5. Blockchart

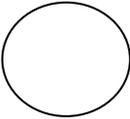
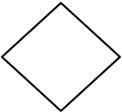
Kristanto (2008:75) menjelaskan bahwa “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi”.

Adapun simbol-simbol digunakan dalam *blockchart* adalah sebagai berikut:

Tabel 2.5 Simbol-simbol *blockchart*

No	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan.
2.		Multi dokumen.
3.		Proses manual.
4.		Proses yang dilakukan oleh komputer.
5.		Menandakan dokumen yang diarsifkan(arsif manual)
6.		Data penyimpanan(<i>data storage</i>)
7.		Proses apa saja yang tidak terdefinisi termasuk aktifitas fisik.
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang ain.

Lanjutan Tabel 2.5 Simbol-simbol *blockchart*

No	Simbol	Keterangan
9.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
10.		Terminasi yang menandakan awal dan akhir dari suatu aliran.
11.		Pengambilan keputusan (<i>decision</i>).
12.		Layar peraga (<i>monitor</i>).
13.		Pemasukan data secara manual.

Sumber : Kristanto (2008:77)

2.3.6. Entity Relationship Diagram (ERD)

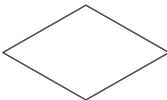
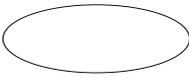
Ladjamudin (2013:142) menjelaskan bahwa “ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak”.

Simbol-simbol yang digunakan dalam ERD, yaitu:

Tabel 2.6 Simbol-simbol *Entity Relationship Diagram (ERD)*

No	Nama Simbol	Simbol	Keterangan
1.	<i>Entitas</i>		Orang, tempat atau benda memiliki nama tunggal.

Lanjutan Tabel 2.6 Simbol-simbol *Entity Relationship Diagram (ERD)*

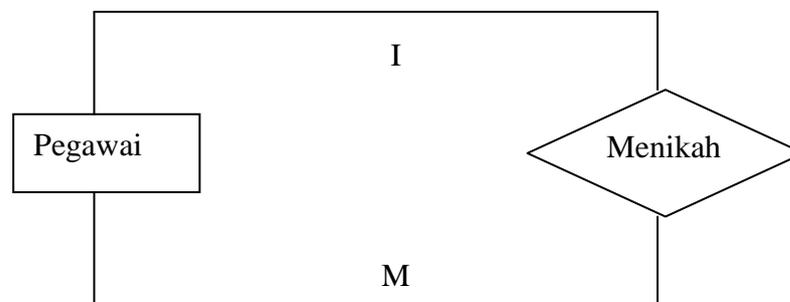
No	Nama Simbol	Simbol	Keterangan
2.	<i>Relationship</i>		Menunjukkan hubungan antar 2 entitas. Dideskripsikan dengan kata kerja.
3.	<i>Atribut</i>		Properti dari entitas harus digunakan oleh minimal 1 proses bisnis dipecah dalam detail.
4.	<i>Link</i>		Penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.

Sumber : Ladjamudin (2013:143)

Menurut Ladjamudin (2013:142), derajat yang sering dipakai di dalam ERD sebagai berikut:

1. *Unary Relationship* adalah model *Relationship* yang terjadi diantara *entity* yang berasal dari *entity* set yang sama.

Contoh:



Gambar 2.3 contoh *Unary Relationship*

Gambar diatas, *relationship* Menikah menunjukkan *relationship* satu-ke-satu antara *instance-instance* dari entitas PEGAWAI.

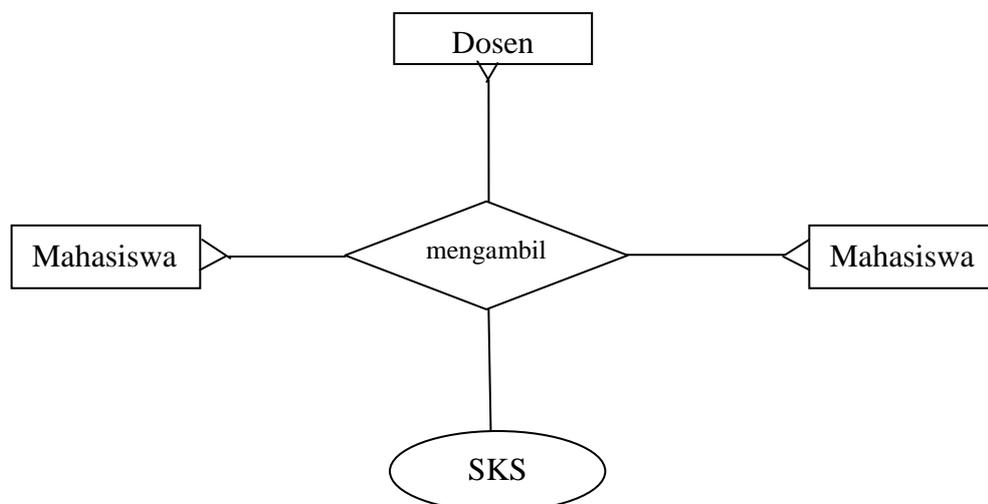
2. *Binary Relationship* adalah model *relationship* antara *instance-instance* dari satu tipe *entitas* (dua *entity* yang berasal dari *entity* yang sama). *Relationship* ini paling umum digunakan dalam pembuatan model data.
Contoh:



Gambar 2.4 contoh *Binary Relationship*

Gambar diatas menunjukkan bahwa *relationship* bekerja untuk *relationship* satu-ke-banyak, artinya seorang mahasiswa hanya dapat bekerja untuk satu perusahaan sedangkan sebuah perusahaan memiliki banyak pegawai.

3. *Ternary Relationship*, merupakan *relationship* antara *instance-instance* dari tiga tipe entitas secara sepihak.
Contoh:



Gambar 2.5 contoh *Ternary Relationship*



Gambar diatas menunjukkan keterkaitan antara *Instance-instance* satu dengan yang lain. Masing-masing entitas memungkinkan berpartisipasi satu atau banyak dalam suatu *relationship ternary*.

2.4. Teori Program

2.4.1. Microsoft Visual Basic 6.0

2.4.1.1. Mengenal Microsoft Visual Basic 6.0

Microsoft Visual Basic merupakan salah satu aplikasi pemrograman visual yang memiliki bahasa pemrograman yang cukup populer dan mudah untuk dipelajari. Basis bahasa pemrograman yang digunakan dalam *Visual Basic* adalah bahasa BASIC (*Beginners All-Purpose Symbolic Instruction Code*) yang merupakan salah satu bahasa pemrograman tingkat tinggi yang sederhana dan mudah dipelajari. Dengan *Visual Basic*, kita bisa membuat program dengan aplikasi GUI (*Graphical User Interface*) atau program yang memungkinkan pengguna komputer berkomunikasi dengan komputer tersebut menggunakan *grafik* atau gambar.

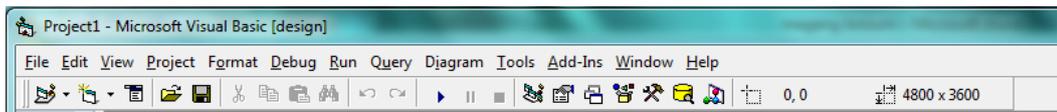
Microsoft Visual Basic 6.0 menyediakan berbagai perangkat kontrol yang dapat digunakan untuk membuat program aplikasi dalam sebuah form baik aplikasi kecil, sederhana hingga aplikasi pengolahan database. (Madcoms, 2008:1)

2.4.1.2. Komponen Microsoft Visual Basic 6.0.

Komponen yang terdapat pada *Microsoft Visual Basic 6.0*. adalah sebagai berikut :

1. Main Window

Main window terdiri dari title bar (baris judul), *menu bar* dan *toolbar*. Baris judul berisi proyek, mode operasi *Visual Basic* sekarang dan *form* yang aktif. *Menu bar* merupakan menu *drop-down* dimana kita dapat mengontrol operasi dari lingkungan *Visual Basic*. *Toolbar* berisi kumpulan gambar yang mewakili perintah yang ada di menu. Jendela utama juga menampilkan lokasi dari *form* yang aktif terhadap sudut kiri atas layar, juga lebar dan panjang dari *form* yang aktif.



Gambar 2.6. Jendela utama Microsoft Visual Basic 6.0

Adapun fungsi dari masing-masing jendela utama *Microsoft Visual Basic 6.0* ini adalah sebagai berikut :

Tabel 2.7. Komponen-komponen Toolbar

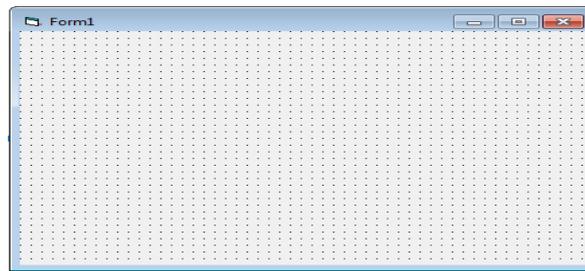
Gambar Toolbar	Nama Toolbar	Fungsi
	<i>Add Project</i>	Untuk menambahkan <i>Project</i> ke dalam project yang sudah ada.
	<i>Add Form</i>	Untuk Menambahkan <i>Form</i> pada <i>Project</i> .
	<i>Menu Editor</i>	Untuk menampilkan <i>Menu Editor</i> untuk mengubah tampilan menu.
	<i>Open Project</i>	Untuk membuka proyek <i>Visual Basic</i> yang sudah ada.
	<i>Save Project</i>	Untuk menyimpan proyek <i>Visual Basic</i> .
	<i>Cut</i>	Untuk memotong elemen yang dipilih pada layar.
	<i>Copy</i>	Untuk meng- <i>copy</i> elemen yang dipilih pada layar.
	<i>Paste</i>	Untuk menyalin elemen yang sebelumnya sudah dipotong (<i>cut</i>) atau disalin (<i>copy</i>).
	<i>Undo</i>	Untuk membatalkan perintah.
	<i>Redo</i>	Untuk mengulang perintah atau mengembalikan perintah yang baru saja dibatalkan.

Gambar Toolbar	Nama Toolbar	Fungsi
	<i>Start</i>	Untuk menjalankan proyek yang dibuat pada program.
	<i>Break</i>	Untuk Memberhentikan <i>running</i> program dan masuk ke <i>Code Editor</i> .
	<i>End</i>	Untuk menghentikan <i>running</i> program.
	<i>Project Explorer</i>	Untuk menampilkan jendela <i>project explorer</i> .
	<i>Property Window</i>	Untuk menampilkan jendela <i>properties window</i> .
	<i>Form Layout</i>	Untuk menampilkan jendela <i>form layout window</i> .
	<i>Object Browser</i>	Untuk menampilkan jendela <i>object browser</i> .
	<i>Toolbox</i>	Untuk menampilkan jendela <i>toolbox</i> .

2. Form

Bagian ini disebut halaman *interface*, yang dapat melakukan desain terhadap program. *Form* adalah daerah kerja utama, di mana kita akan membuat program-program aplikasi *Visual Basic*. Pada *form* ini, kita akan meletakkan berbagai macam objek interaktif seperti teks, gambar, tombol-tombol perintah, *scrollbar* sebagainya. Jendela *form* ini pada awalnya kelihatan kecil tetapi ukurannya bisa berubah-ubah sesuai dengan kebutuhan aplikasi kita.

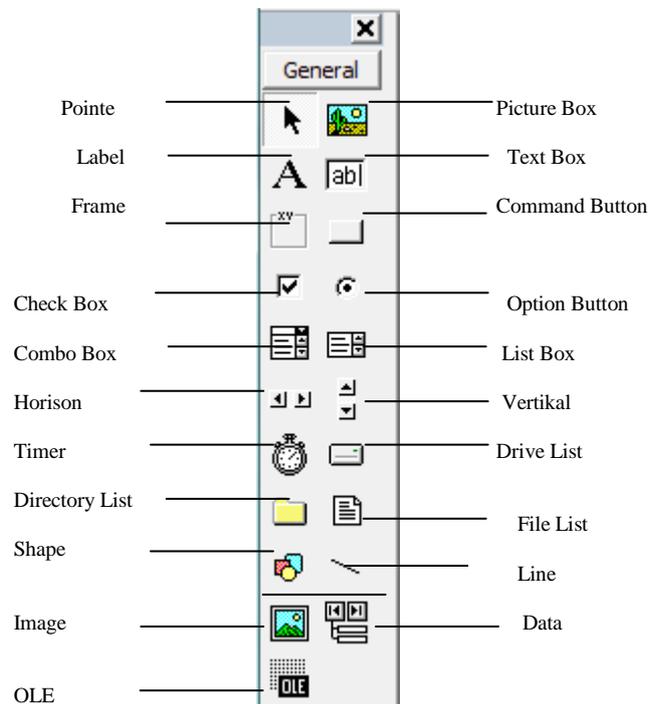
Apabila program aplikasi kita jalankan, semua yang terdapat dalam *form* akan ditampilkan pada layar *window*. Jendela *form* inilah yang nantinya akan menjadi latar belakang aplikasi kita.



Gambar 2.7. Jendela Form

3. Toolbox

Seluruh komponen/objek yang digunakan diletakkan pada bagian ini. Pada tampilan standar komponen yang ditampilkan hanyalah komponen yang paling sering dipakai. Untuk dapat menambah atau mengurangi komponen sesuai dengan yang dibutuhkan.



Gambar 2.8. Komponen Toolbox

Adapun pengertian dari masing-masing komponen *Toolbox* yang ada pada sebuah objek *Microsoft Visual Basic 6.0* ini antara lain:

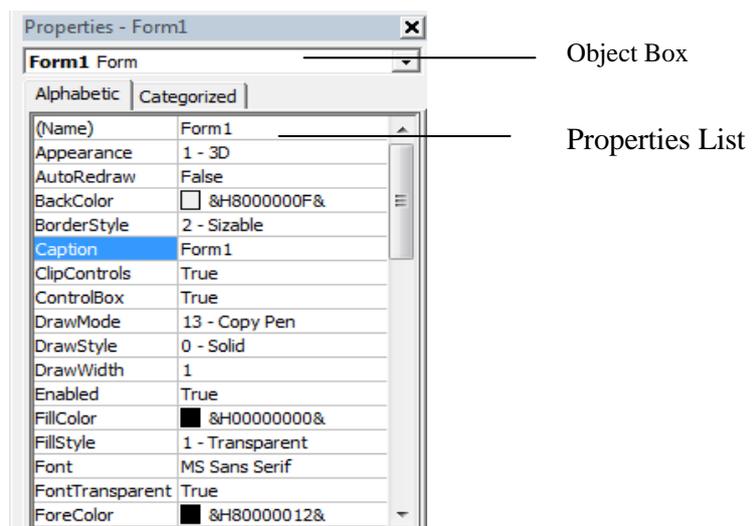
Tabel 2.8. Komponen-komponen Toolbox

Gambar Kontrol	Nama Kontrol	Fungsi
	<i>Label</i>	Untuk menampilkan teks yang tidak bisa diubah oleh pemakai program kita.
	<i>Text Box</i>	Untuk membuat teks dimana teksnya bisa diubah oleh pemakai.
	<i>Command Button</i>	Untuk membuat sebuah tombol pelaksanaan perintah.
	<i>Check Box</i>	Untuk membuat kotak yang mudah pemakaiannya atau memberikan perintah pilihan benar/salah, ya/tidak.
	<i>Option Button</i>	Untuk pemilihan dua keadaan dari banyak pilihan, namun hanya satu pilihan saja yang bisa diaktifkan pada suatu saat.
	<i>Combo Box</i>	Sebagai unit kontrol kombinasi antara <i>text box</i> dengan <i>list box</i> .
	<i>List box</i>	Untuk menampilkan daftar pilihan yang bisa digulung.
	<i>Horizontal Scollbar</i>	Untuk penggulangan dengan jangka lebar dengan indikasi posisi pemilihan dalam posisi horizontal.
	<i>Vertical Scrollbar</i>	Untuk penggulangan dengan jangka lebar dengan indikasi posisi pemilihan dalam
	<i>Timer</i>	Untuk penghitung waktu <i>event</i> dalam interval yang ditentukan.
	<i>Picture Box</i>	Untuk menampilkan gambar statis maupun gambar aktif dari sumber di luar dirinya.

Gambar Kontrol	Nama Kontrol	Fungsi
	<i>Line</i>	Untuk membuat garis lurus di dalam <i>form</i> .
	<i>Data</i>	Untuk menampilkan database pada suatu <i>form</i> .
	<i>Object Linking dan embedding (OLE)</i>	Untuk memungkinkan pemakai menempelkan suatu objek dari aplikasi yang mendukung OLE ke aplikasi <i>Visual Basic</i> .

4. Window Property

Jendela *Properties* adalah jendela yang mengandung semua informasi mengenai objek yang terdapat pada aplikasi *Microsoft Visual Basic 6.0*. *Properties* adalah sifat dari sebuah objek misalnya nama, warna, ukuran, posisi dan sebagainya. Setiap objek sebagian besar memiliki jenis *properties* yang sama tapi ada Pula yang berbeda. Di bagian yang paling atas terdapat kotak yang menunjukkan nama objek yang sedang aktif, sedang propertinya ditampilkan di bagian bawah dari jendela *properties* tersebut.



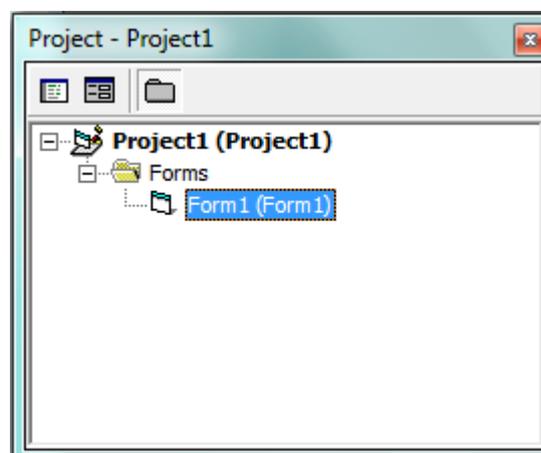
Gambar 2.9. Jendela *Properties*

Property window berisi:

- a. *Object Box*: berisi daftar komponen yang dipakai dalam *form*.
- b. *Sort Tab* : jika dipilih *Alphabetic*, maka akan menampilkan struktur yang terurut berdasarkan *Alphabetic*. *Categorized* menampilkan properti yang dikelompokkan berdasarkan *font*, *position*, dan lain-lain.
- c. Kolom : kolom kiri berisi daftar properti pada objek yang terseleksi, sedangkan kolom kanan untuk mengganti nilai properti yang diinginkan.

5. Project Explorer

Project Explorer adalah jendela yang mengandung semua *file* didalam aplikasi *Visual Basic*. Setiap aplikasi dalam *Microsoft Visual Basic 6.0* disebut dengan istilah *Project* (proyek), dan setiap proyek bisa mengandung lebih dari satu *file*, baik *form*, *module*, *report*, *data environment* atau file lain pendukung *project*. Pada *Project Explorer* ditampilkan semua file yang terdapat pada aplikasi (proyek).



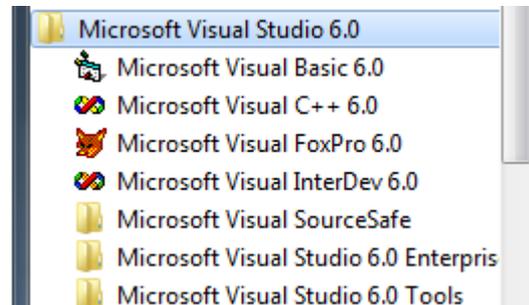
Gambar 2.10. Project Explorer

2.4.1.3. Menjalankan Microsoft Visual Basic 6.0

Beberapa langkah dalam menjalankan *Microsoft Visual Basic 6.0* adalah :

1. Klik **Start** dari menu **Task bar**.
2. Pilih item pilihan **Program**.

3. Pilih **Microsoft Visual Studio 6.0**, Selanjutnya pilih sub pilihan **Microsoft Visual Basic 6.0** . Lihat gambar di bawah untuk lebih jelas.



Gambar 2.11. Memulai Microsoft Visual Basic 6.0

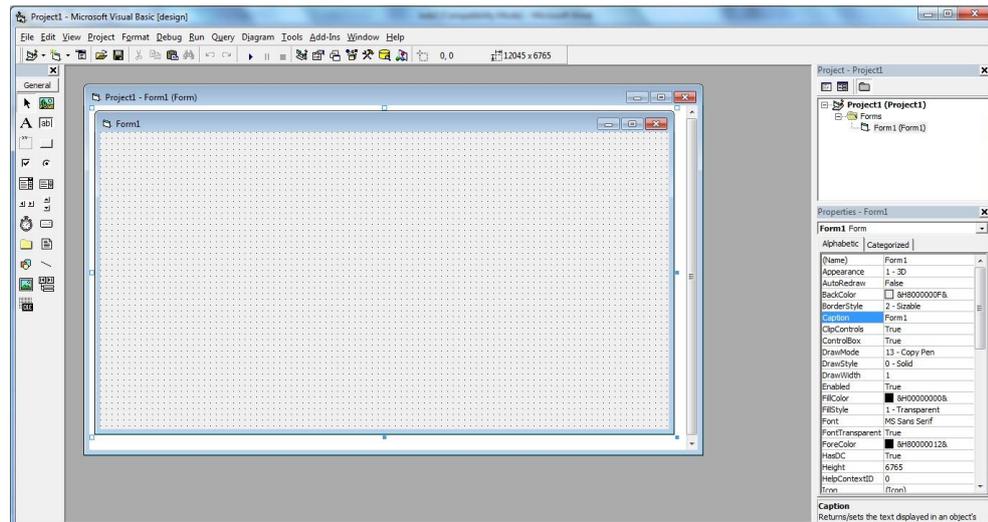
4. Setelah itu muncul **Dialog Box New Project** seperti gambar di bawah ini



Gambar 2.12. Kotak Dialog New Project

Kotak dialog ini meminta konfirmasi untuk memilih jenis project yang ingin disusun. Klik salah satu ikon jenis project atau tipe project, kemudian klik tombol **open**. Jika tidak memilih klik tombol **cancel**.

5. Setelah memilih salah satu tipe dari project, misalnya jenis **Standar EXE**, maka akan muncul tampilan baru seperti gambar di bawah ini.



Gambar 2.13. Tampilan IDE Microsoft Visual Basic 6.0

Tampilan ini disebut dengan lingkungan Integrated Development Environment (IDE). Dari tampilan ini sudah siap uuntuk mendefinisikan aplikasi EXE yang akan dibuat, yaitu dengan mengoptimalkan elemen- elemen IDE nya.

2.4.2. Microsoft Access 2007

2.4.2.1. Pengenalan Microsoft Access 2007

Microsoft Office Access 2007 tergolong sebagai DBMS (*Data Base Management System*). Perangkat lunak ini bermanfaat untuk mengelola data dengan cara yang sangat fleksibel dan cepat. Berikut adalah sejumlah aktivitas yang terkait dengan data yang didukung oleh perangkat lunak tersebut.

1. Menyimpan data dengan ukuran yang tidak terbatas
2. Mengatur data yang tersimpan agar selalu konsisten
3. Mengambil data yang tersimpan dengan mudah dan cepat
4. Memungkinkan untuk memilih hanya data tertentu yang diambil



5. Memungkinkan untuk mengkombinasikan teks, gambar, dan bahkan suara.
6. Memasukkan data yang dapat ditangani melalui suatu formulir.
7. Menyajikan informasi dengan cepat dan mudah melalui suatu fasilitas pembangkit laporan.
8. Mengimpor data dari format lain (misal *Ms. Excel*).
9. Mengekspor data kedalam format lain (misalnya ke bentuk RTF atau *file Teks*).

2.4.2.2 . Konsep Database dalam Access

Database adalah suatu wadah untuk mengelola data. Sebuah database melibatkan beberapa tabel (Kadir 2010:3).

Database menyatakan suatu wadah untuk mengelola data. *Database* mencakup sejumlah tabel dan berbagai objek yang terkait dengan pengelolaan data. Objek selain tabel ini dapat berupa formulir (*form*), *query*, laporan (*report*), halaman (*page*), makro, dan modul. Berikut adalah penjelasan singkat tentang objek-objek tersebut.

1. *Tabel*, merupakan objek yang digunakan untuk menyimpan kumpulan data yang terkait. Misalnya, *tabel* teman, digunakan untuk menyimpan data nama, tanggal lahir dan alamat teman.
2. *Formulir*, adalah tampilan atau antarmuka yang memudahkan dalam memasukkan data atau menyajikan informasi.
3. *Query*, adalah suatu bentuk untuk memperoleh informasi dari *database*.
4. *Report*, adalah dokumen yang digunakan untuk menyajikan informasi dari *database*.
5. *Page*, adalah objek yang memungkinkan pengaksesan informasi dari database ke dalam bentuk halaman *web*.
6. *Macro*, berisi kumpulan kode yang digunakan untuk mengotomasikan suatu tindakan tertentu dengan sekali langkah.



-
7. *Modul*, adalah program yang ditulis dengan menggunakan VBA (*Visual Basic for Applications*) yang ditujukan untuk memperluas fungsi sebuah *database*.