Pemrograman Visual

Microsoft Visual Basic 6.0

Eko Purwanto

Katalog Dalam Terbitan (KDT)
Pemrograman Visual dengan Visual Basic 6.0 ;
Eko Purwanto

Penulis : Eko Purwanto
Editor : C. Suryanto.
Tata Letak : Ach Chotim DC
Desain Sampul : S. Nugraha
Diterbitkan Oleh :

DUTA PUBLISHING INDONESIA
JL. PULANGGENI NO. 8 SOLO TELP. (0271) - 712432

Cetakan I : Juni 2012
Perpustakaan Nasional RI :
Eko Purwanto.

380 hlm., 20, 5 cm
ISBN: 978-602-9348-15-6

Kata Pengantar

Mata kuliah Pemrograman Visual dengan Visual Basic 6.0 di STMIK Duta Bangsa Surakarta diberikan untuk memperkenalkan Lingkungan Visual Basic, pembuatan *interface* dengan menggunakan *tools* yang ada serta mengimplementasikan program sederhana dengan bahasa Visual Basic. Disamping itu juga memahami struktur dan perintah-perintah program bahasa Visual Basic.

Buku pemrograman Visual dengan Visual Basic 6.0 ini terdiri dari 13 Bab sesuai dengan sistem perkuliahan yang ada di STMIK Duta Bangsa Surakarta untuk satu semester. Pada setiap pertemuan diberikan contoh-contoh program dan latihannya, diharapkan dengan mencoba contoh program yang ada dan mengerjakan latihannya, mahasiswa lebih mudah untuk memahami materi yang diberikan.

Penyusun berharap semoga modul ini bisa membantu mahasiswa STMIK Duta Bangsa Surakarta dalam belajar, khususnya untuk pemrograman Visual Basic. Dalam kesempatan ini penyusun harapkan segala macam kritik yang bersifat membangun demi perbaikan modul ini di masa mendatang. Terima Kasih.

Penyusun

BAB I

MENGENAL VISUAL BASIC 6.0

A. Pendahuluan

Bahasa Basic merupakan bahasa populer dan disukai banyak programer karena kemudahannya serta bahasanya yang cukup familier tanpa mengurangi perfomance kinerjanya. Sejak kemunculannya pada tahun 1960, bahasa Basic telah mengalami perkembangan yang pesat sekali. Di tahun 1970 digunakan oleh Bill Gates dan Paul Allen untuk mengontrol mikrokomputer Alltair dengan menggunakan pita kaset. Kemudian bahasa Basic diikuti oleh pengembang-pengembang software lain dengan nama yang berbeda, namun aturan dan bahasa yang digunakan adalah sama. Munculnya GW-Basic, Qbasic, Quick Basic dan lain sebagainya semakin mempopulerkan bahasa Basic ini untuk digunakan pada mikrokomputer sebagai bahasa pemrograman untuk membuat aplikasi.

B. Materi

1. Mengenal Visual Basic

Visual Basic untuk DOS dan Windows diperkenalkan pada tahun 1991. Versi 3.0 dari Visual Basic dikeluarkan pada tahun 1993 dan lebih mengalami kemajuan yang pesat dibandingkan dengan versi sebelumnya. Visual Basic 3.0 masih menggunakan kode-kode yang bekera dalam 16 bit. Kemudian pada akhir tahun 1995 dikeluarkan Visual Basic versi 4.0 yang mendukung proses 32 bit. Pada akhir tahun 1996 dikeluarkan Visual Basic versi 5.0 dengan kelebihan yang dapat mendukung control Activex dan mulai menghapus atau menghilangkan dukungan terhadap proses 16 bit. Dan versi yang dipakai dalam modul ini adalah Visual Basic versi 6.0 yang dikemas dalam satu paket Microsoft Visual Studio 6.0.

Visual Basic 6.0 ialah bahasa pemrograman event-driven yang berasal dari **BASIC**. *Event driven* artinya program menunggu sampai adanya respons dari pemakai berupa kejadian tertentu, misalnya tombol diklik atau menu

dipilih. Ketika event terdeteksi, event yang berhubungan akan melakukan aksi sesuai dengan kode yang diberikan. Ada tiga edisi yang dikeluarkan Microsoft, yaitu:

a. **Standard Edition**

Standard Edition sangat merekomendasikan bagi pemula yang ingin mempelajari Visual Basic 6.0 dan mempunyai fasilitas sebagai berikut:

- Kemampuan aplikasi 32 bit yang berjalan di Microsoft Win 9x dan Win NT untuk pemula.
- Terdiri dari control seperti grid, tab, dan Data Bound.
- Memuat Learn Visual Basic Now dan Online Help.
- Microsoft Developer Network CD berisi dokumentasi.

b. **Professional Edition**

Professional Edition umumnya digunakan oleh para professional yang sudah cukup mendalami Visual Basic 6.0. Tidak terlalu banyak perbedaan dengan Standard Edition, hanya ada beberapa tambahan, diantaranya:

- ActiveX Control, termasuk Internet Control
- IIS (Internet Information Server)
- Dynamic HTML Page Designer

c. **Enterprise Edition**

Lebih ditekankan untuk membuat aplikasi yang bersifat *server based*, tapi program-program aplikasi standard dapat berjalan dengan baik jika menggunakan versi ini. Fasilitas tambahan antara lain:

- Application Performance Explorer
- IIS (Internet Information Server)
- Support for Microsoft Transaction Server 2.0
- SQL Debugging
- Visual Component Manager
- Visual Database Tool

Visual Basic merupakan salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi Windows. Visual Basic merupakan salah satu bahasa pemrograman komputer yang mendukung object (*Object Oriented Programming* = OOP). Sa

yangnya, Visual Basic sampai saat ini hanya dapat berjalan diatas lingkungan sistem operasi Windows. Untuk kalangan sistem operasi yang lain seperti Linux misalnya, Visual Basic masih belum bisa berjalan optimal walaupun saat ini sudah mulai dirintis sebuah framework berbasis .NET agar dapat menjalankan aplikasi VB.NET diatas platform Linux.

2. Mengenal Integrated Development Environment (IDE) VB 6.0

Kepopuleran Visual Basic sebenarnya datang dari lingkungannya yang sering disebut *Integrated Development Environment* atau IDE. IDE membantu membangun sebuah aplikasi besar, menulis sebuah program, menjalankan program, dan menghasilkan sebuah *executable file*. Executable File yang dihasilkan oleh Visual Basic bersifat independen, dan karena itu file tersebut dapat dijalankan pada komputer tanpa harus menginstall Visual Basic.

Pemograman visual merupakan dimensi baru dalam pembuatan aplikasi karena dapat langsung menggambarkan objek-objek ke layar sebelum dieksekusi. Dalam lingkungan pengembangan visual, sekarang objek yang anda buat hasilnya langsung tampil di layar. Objek yang dibuat itu akan sama hasilnya pada saat program dijalankan. Dengan demikian tidak perlu lagi melakukan pengubahan kode program secara manual. Setelah semua objek diletakkan dalam suatu form, maka semua atribut objek tersebut akan disimpan dalam suatu kode program yang dapat langsung dijalankan.

a. Menjalankan IDE

Untuk membuka program Visual Basic 6.0 (yang dikemas dalam Microsoft Visual Studio 6.0), ada beberapa cara yang bisa dilakukan diantaranya sebagai berikut:

- Klik **Start – Program - Microsoft Visual Studio 6.0 – Microsoft Visual Basic 6.0**. Maka pada layar awal akan muncul tampilan seperti pada gambar 1.1.
- Dengan membuat *short cut* pada jendela desktop dan untuk memulainya cukup melakukan *double-click* pada short cut tersebut.

- Ketika Visual Basic diinstall, file-file Visual Basic (*.frm, *.bas, *.vpb) di daftarkan pada sistem operasi Window, karena itu untuk memulai visual basic anda dapat melakukan double-click pada file-file tersebut

b. Memilih Tipe Project

Visual Basic menyediakan banyak jenis modul aplikasi. Beberapa pilihan yang terdapat pada kotak dialog New Project adalah sebagai berikut :

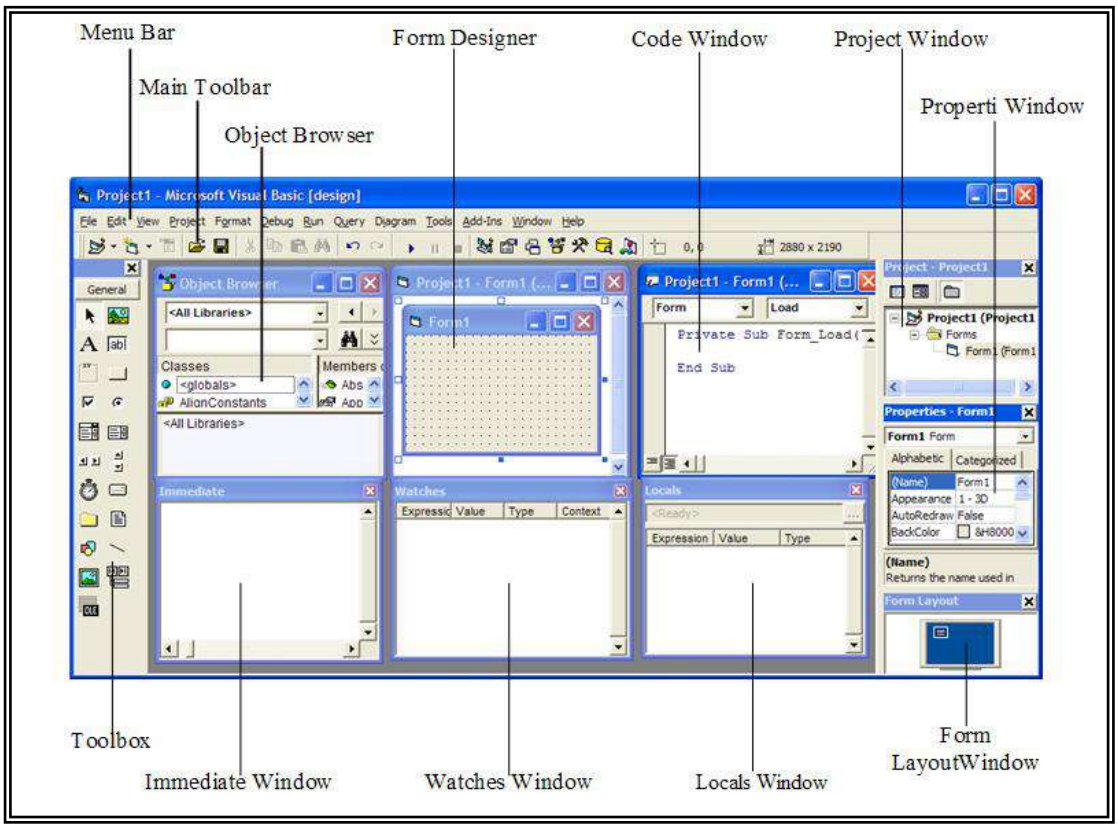
- Standard EXE : membuat aplikasi Visual Basic Standar
- Active EXE : membuat aplikasi ActiveX
- Active DLL : membuat library ActiveX
- ActiveX Control : membuat kontrol ActiveX
- VB Application Wizard : membuat aplikasi dengan bantuan Wizard
- VB Wizard Manager : pusat pengelolaan Wizard Visual Basic
- IIS Application :membuat aplikasi IIS (Internet Information Server)
- DHTML Application :membuat applikasi DHTML (Dynamic Hypertext Mark-up Language) untuk internet.

Tab **Existing** untuk menampilkan aplikasi yang sudah ada dan Tab **Recent** menampilkan aplikasi yang pernah dibuka terakhir kali.

Untuk memulai program standar pilihlah **Standard EXE**, kemudian klik pada tombol **Open**. Setelah itu akan muncul window Project1-Microsoft Visual Basic seperti pada gambar 1.2.

Gambar 1.1 Kotak Dialog New Project

Sekarang kita akan mengenal bagian-bagian dari IDE (Integrated Development Environment) yang kita gunakan seperti pada gambar 1.2



Gambar 1.2 IDE Visual Basic

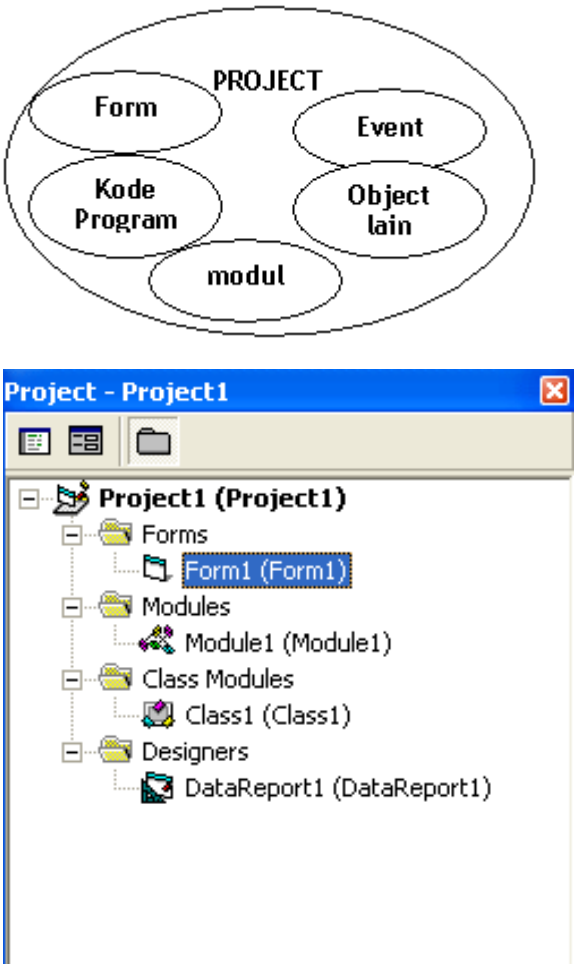
3. Komponen IDE dalam VB 6.0

Jendela IDE Visual Basic memiliki sejumlah menu, toolbar, dan window seperti yang terlihat pada gambar 1.2. Berikut ini akan dijelaskan kegunaan masing-masing item :

a. Project Window

Jika membuat program aplikasi, akan terdapat jendela project yang berisi semua file yang dibutuhkan untuk menjalankan program aplikasi Visual Basic yang dibuat. Pada jendela project terdapat tiga icon yaitu icon **View Code** untuk menampilkan jendela editor, icon **View Object** untuk menampilkan bentuk formulir (Form) dan icon **Toggle Folders** digunakan untuk menampilkan folder (tempat penyimpanan file).

Pertama kali ketika menggunakan program Visual Basic maka komponen project yang akan diload, selanjutnya adalah menambah form-form atau membuat modul atau mungkin membuat kode program. Secara lebih sederhana posisi project dalam setiap komponen yang ada adalah seperti terlihat pada gambar dibawah ini :

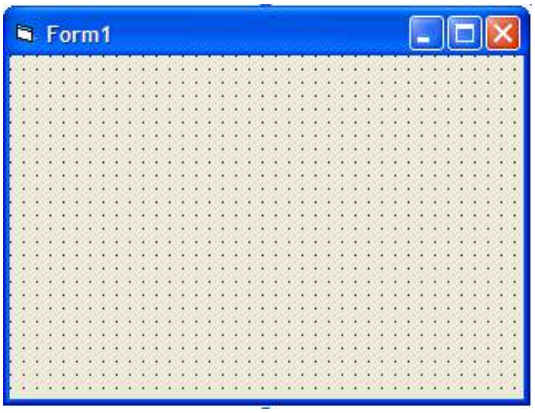


Gambar 1.3 Keterkaitan Project dengan komponen lainnya

b. Form Designer

Form digunakan ketika akan meletakkan object-object apa saja yang akan digunakan dalam program, object-object yang terdapat dalam toolbox, diletakkan dan didesain dalam bagian form. Form sebenarnya adalah suatu objek yang dipakai sebagai tempat bekerja program aplikasi.

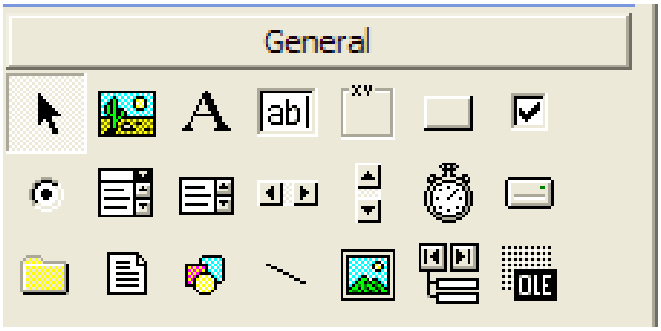
Secara otomatis akan tersedia form yang baru jika kita membuat program aplikasi baru, yaitu dengan nama **Form1**. Umumnya dalam suatu form terdapat garis titik-titik yang disebut dengan **Grid.**



Gambar 1.4 Form Kosong

c. Toolbox

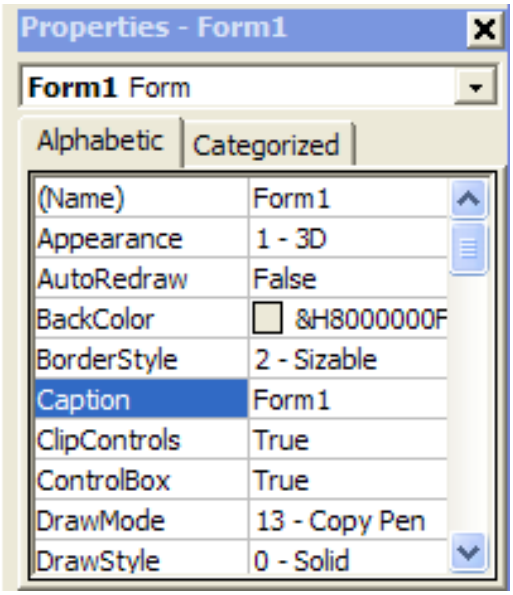
Toolbox adalah kotak alat yang berisi icon-icon untuk memasukkan objek tertentu ke dalam jendela form. Kita dapat memodifikasi toolbox, misalnya menambah komponen icon dengan cara melakukan klik kanan pada toolbox lalu memilih **Components** atau **Add Tab.**



Gambar 1.5 Kumpulan Toolbox

d. Properties Window

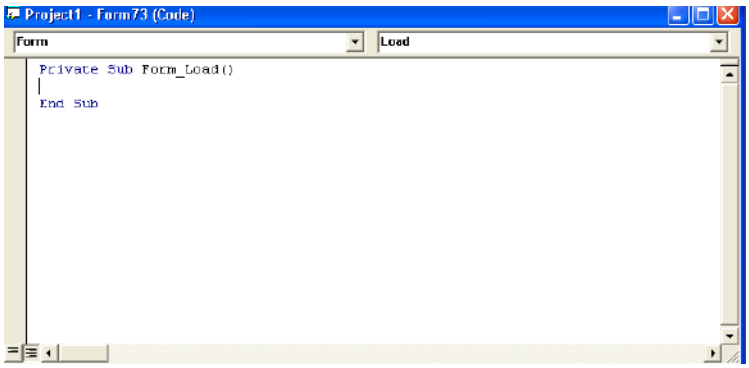
Properties digunakan untuk menentukan setting suatu objek. Suatu objek biasanya mempunyai beberapa properti yang dapat diatur langsung dari jendela **Properties** atau lewat kode program.



Gambar 1.6 Property untuk Form 1

e. Code Window

Digunakan untuk menulis kode program yang menentukan tingkah laku dari form dan objek-objek yang ada pada aplikasi bersangkutan. Kode program adalah serangkaian tulisan perintah yang akan dilaksanakan jika suatu objek dijalankan. Kode program ini akan mengontrol dan menentukan jalannya suatu objek.



Gambar 1.7 Suatu Prosedur dalam code window

f. Module

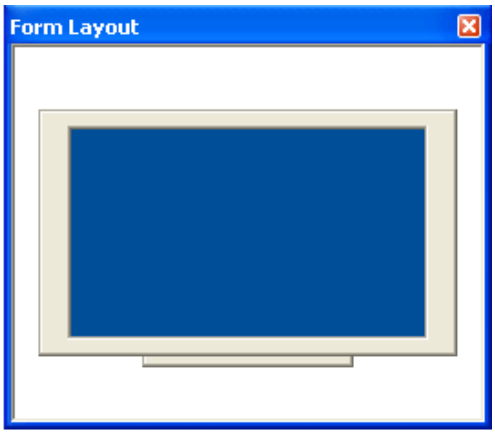
Module dapat disejajarkan dengan form, tetapi tidak mengandung object. Module dapat berisi kode-kode program atau procedure yag dapat digunakan dalam program aplikasi.

g. Color Pallete Window

Digunakan untuk menentukan warna dari suatu objek seperti warna dari objek yang sedang aktif saat ini.

h. Form Layout Window

Menunjukan bagaimana peletakan sebuah form akan ditampilkan pada saat dijalankan.



Gambar 1.8 Bentuk Form Layout window

i. Immediate Window

Digunakan untuk memasukkan ekpresi untuk melihat hasilnya dengan menggunakan perintah “print” atau ‘?’. Jendela ini biasanya digunakan bersama watch window pada saat sebuah program di-debug. Short-cut untuk jendela ini adalah Ctrl-G.

j. Object Browser

Digunakan untuk menyelusuri external libraries sehingga anda dapat mempelajari objek-objek dan properti, kejadian (events), dan method yang dimilikinya. Short-cut untuk jendela ini adalah F2.

k. Locals Window

Akan aktif hanya pada saat program di jalankan, berisi nilai dari sejumlah variabel yang bersifat lokal pada sebuah prosedur atau module.

l. Watch Window

Digunakan untuk memonitor nilai dari suatu varibel baik bersifat lokal maupun global.

m. Call Stack Window

Hanya ditampilkan jika suatu program yang dijalankan dihentikan untuk sementara (break) dan menekan Ctrl-L. Menunjukan semua prosedur yang menunggu prosedur aktif telah selesai. Jendela ini penting untuk proses debugging untuk mengetahui jalur eksekusi program hingga sampai situasi sekarang

4. Menu, Toolbar dan ToolBox

a. Menu

Menu yang akan dibahas secara garis besar, sebab sebagian dari menu berkaitan dengan Visual Basic lanjutan. Berikut ini akan dibahas mengenai menu yang sifatnya umum dan sering dipakai.

- ***Menu File*** berisi perintah antara lain, *open* untuk membuka suatu project dan *save* untuk menyimpan project atau form, *make* digunakan untuk mengkompile project menjadi Executable file.
- ***Menu Edit*** berkaitan dengan perintah-perintah editing seperti *cut, copy, paste, find, replace, undo* dan *redo*.
- ***Menu View*** digunakan untuk menampilkan window-window pada IDE Visual Basic yang telah dijelaskan pada gambar 1.3.
- ***Menu Project*** digunakan untuk menambahkan objek-objek baru seperti pada form, standart module (bas), class module, User control module, dll.
- ***Menu Format*** digunakan untuk mengatur posisi dan ukuran satu atau beberapa kontrol.
- ***Menu Debug*** berisi perintah-perintah yang digunakan pada saat program sedang di debug. Aplikasi yang dieksekusi tahap demi tahap, nilai dari variabel yang bersangkutan ditampilkan, dan menambahkan *break point* untuk memonitor jalannya program.
- ***Menu Run*** berisi perintah-perintah untuk menjalankan aplikasi yang dibangun.

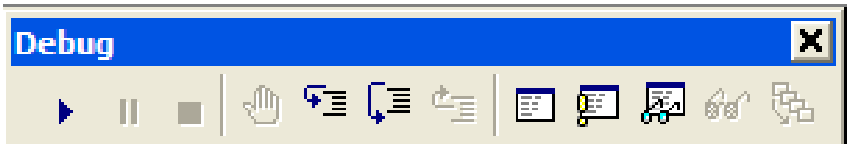
b. Toolbar

Visual Basic memiliki sejumlah toolbar. Toolbar-toolbar tersebut dapat kita letakkan pada posisi sebelah atas pada IDE Visual Basic atau dibiarkan menjadi window di dalam IDE Visual Basic. Toolbar-toolbar tersebut antara lain :

- ***Standard Toolbar*** berisi tool yang digunakan untuk perintah-perintah seperti membuka atau menyimpan sebuah project.

Gambar 1.9 StandarToolbar

- ***Debug Toolbar*** digunakan berisi perintah seperti yang terdapat pada menu debug, jika toolbar tersebut tidak terdapat pada IDE, anda dapat memilih menu view - toolbars untuk menampilkan toolbar tersebut.



Gambar 1.10 Debug Toolbar

- ***Edit Toolbar*** berfungsi pada saat kita mengedit sebuah kode, menammbahkan break point, dan bookmarks.

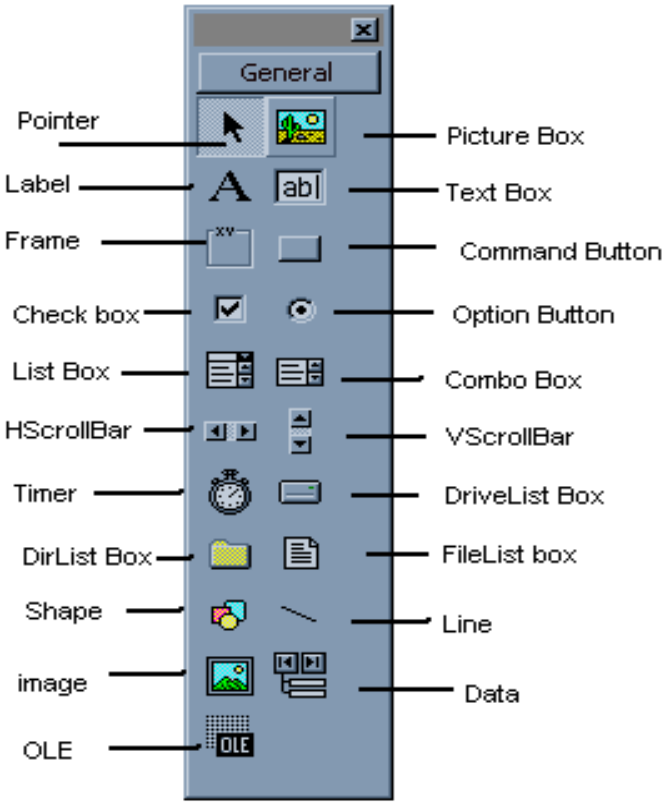
Gambar 1.11 Edit Toolbar

- ***Form Editor Toolbar*** digunakan untuk mengatur posisi control yang ada pada form.

Gambar 1.12 Form Editor toolbar

c. ToolBox

Selain Toolbar, Visual Basic juga mempunyai Jendela Toolbox. Jendela ini berisi sejumlah control yang digunakan untuk mendesain antar muka (*User Interface*) yang akan diletakkan pada Form. Berikut ini akan dijelaskan masing-masing dari control yang ada pada toolbox :



Gambar 1.13 Form Editor toolbar

- ***Pointer*** bukan sebuah kontrol, pointer digunakan untuk memilih control yang sudah ada pada sebuah form.
- ***Picture Box Control***, control ini digunakan untuk menampilkan gambar seperti BMP, DIB (bitmap), ICO (icon), CUR (cursor), WMF (metafile), EMF (enhanced metafile), GIF, and JPEG.
- ***Label Control***, digunakan sebagai text static yang tidak bisa diedit, biasanya digunakan untuk memberi nama pada control yang lain seperti TextBox.
- ***TextBox Control***, control yang berisi string dari karakter yang dapat diisi oleh user.

- ***Frame Control***, digunakan untuk mengelompokkan control-control yang lain.
- ***Command Button Control***, command button hampir muncul pada setiap form, biasanya digunakan untuk menangkap kejadian berupa mouse click.
- ***Check Box Control***, control ini digunakan untuk pilihan ya/tidak atau benar/salah.
- ***Option Button Control***, digunakan dalam suatu group, dimana seorang user dapat memilih salah satu saja dalam suatu group.
- ***ListBox Control***, berisi sejumlah item dimana seorang user dapat memilih salah satu atau beberapa item.
- ***ComboBox Control***, control ini merupakan gabungan antara TextBox dan ListBox Control, dimana daftar item hanya akan ditampilkan jika user men-click pada panah kebawah disebelah kanan Control tersebut.
- ***HscrollBar*** dan ***VscrollBar Control***, kedua control ini digunakan untuk ScrollBar.
- ***Timer Control***, merupakan control yang tidak terlihat pada saat program dijalankan. Control tersebut membangkitkan kejadian pada selang waktu tertentu.
- ***DriveList Box, DirList Box,*** dan ***FileList Box Control***, ketiganya digunakan untuk membuat suatu dialog untuk sistem file. Control tersebut jarang digunakan sebab ada Common dialog Control yang akan kita bicarakan nanti sebagai ganti dari ketiga control diatas.
- ***Shape*** dan ***Line*** Control, digunakan untuk memperindah tampilan dari suatu form.
- ***Image Control***, control ini mirip dengan PictureBoxControl tetapi digunakan sebagai container control- control yang lain.
- ***Data Control***, merupakan kunci dari data binding pada Visual Basic. Dengan menggunakan control ini, kita dapat menghubungkan sebuah database pada Visual Basic.
- ***OLE Control***, digunakan sebagai tempat untuk program ekternal yang ada pada windows seperti spread sheet yang dihasilkan oleh Microsoft

Excel. Dengan menggunakan control tersebut kita dapat menampilkan program lain pada sebuah aplikasi.

Berikut ini penjelasan dan fungsi dari masing-masing kontrol yang ada pada toolbox Visual Basic 6.0.

Tabel 1.1 Penjelasan Toolbox Visual Basic

Bentuk	Nama Kontrol	Fungsi
	Pointer	Pointer ini bukan kontrol tetapi penunjuk kontrol yang berfungsi untuk memindahkan atau mengubah ukuran kontrol yang ada pada form.
	PictureBox	Untuk menampilkan file gambar (Bitmaps, Icon, Gif, JPEG dsb) baik gambar statis maupun aktif. Standar penulisan Pic_, misal : Pic_Foto
	Label	Untuk menampilkanteks, tetapi pemakai tidak bisa beriteraksi dengannya. Standar penulisannya Lbl_, Misal : Lbl_kota
	TextBox	Untuk menempatkan teks pada form dan pemakai dapat mengedit teks tersebut. Standar penulisannya Txt_, misal : Txt_Alamat
	Frame	Untuk mengelompokkan beberapa kontrol (Group) pada suatu form. Standar penulisannya Fra_, misal : Fra_identitas
	Command Button	Untuk membuat tombol pelaksana suatu perintah atau tindakan ketika digunakan. Standar penulisannya Cmd_, misal : Cmd_Simpan
	CheckBox	Untuk membuat kotak check yang dapat memilih satu atau banyak pilihan yang ada. Standar

		penilisannya Chk_, misal : Chk_Hobby
	Option Button	Untuk memilih dan mengaktifkan satu keadaan dari banyak pilihan yang ada. Standar penulisannya Opt_, misal : Opt_Agama
	ComboBox	Sebagai tempat mengetikan pilihan atau memilih suatu pilihan lewat Drop Down-List. Standar penulisannya Cbo_, misal : Cbo_Ukuran
	ListBox	Untuk menampilkan daftar pilihan yang dapat diguung secara horisontal maupun vertikal. Standar penulisannya Lis_, misal : Lis_jenis
	HscrollBar	Untuk menggulung suatu area kerja dengan jangka lebar pada posisi horisontal. Standar penulisannya Hsb_, misal : Hsb_warna
	VscrollBar	Untuk menggulung suatu area kerja dengan jangka lebar pada posisi vertikal. Standar penulisannya Vsb_, misal : Vsb_Warna
	Timer	Untuk mengoperasikan waktu kejadian pada rutin program dalam interval yang ditentukan. Standar penulisannya Tmr_, misal : Tmr_Isi
	DriveListBox	Untuk menampilkan daftar drive komputer yang aktif dan dapat dipilih sebuah drive. Standar penulisannya Drv_, misal: Drv_kerja
	DirListBox	Untuk menampilkan daftar directory dan path pada drive kerja terpilih. Standar penulisannya

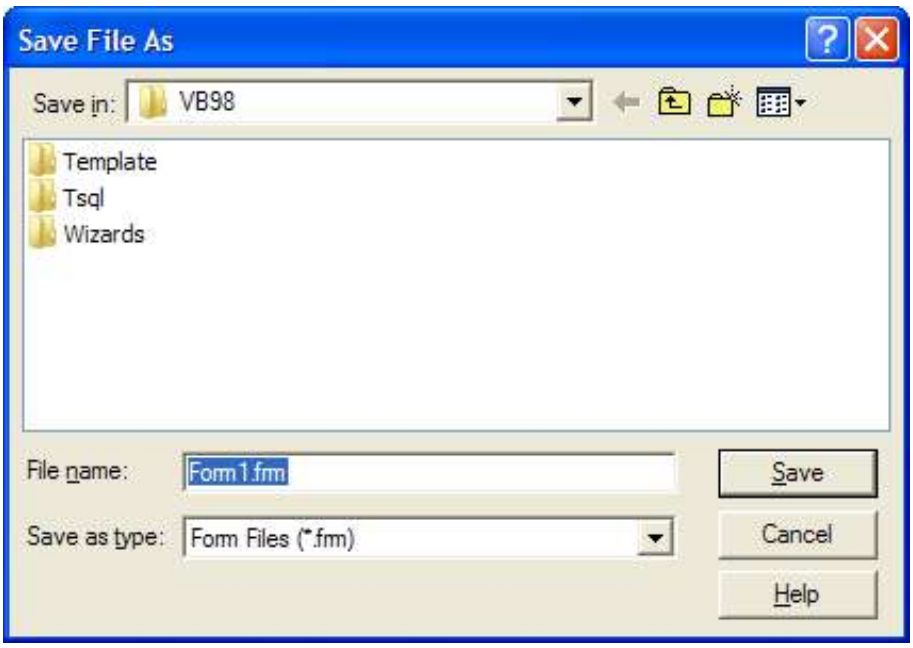
		Dir_, misal : Dir_Surat
	FIleListBox	Untuk menampilkan daftar file pada directory dan drive yang aktif. Standar penulisannya Fil_, misal : Fil_gambar
	Shape	Untuk membentuk objek dua dimensi seperti square, oval, ellips dan lain-lain. Standar penlisannya ShP_, misal : Shp_Lingkaran
	Line	Untuk menggambar garis lurus dengan banyak variasi dengan ketebalan yang bisa diatur. Standar penulisannya Lin_, misal : :Lin_Satu
	Image	Untuk menampilkan gambar icon, bitmap atau metafile pada form. Standar penulisannya Img_, misal : Img_Foto
	Data Control	Sebagai sarana akses data dalam suatu database. Fasilitas ini ada dalam konsep DAO. Standar penulisannya Dat_, misal : Dat_mhs
	OLE	Untuk menghasilkan proses Link dan Embed objek antar aplikasi. Standar penulisannya Ole_, misal : Ole_Sales

5. Menyimpan Project

Langkah-langkah menyimpan:

- ✓ Pada menu File, klik perintah Save Project As kemudian akan muncul Kotak dialog File Project seperti terlihat pada gambar 1.12. Melalui kotak dialog tersebut dapat menyimpan program Visual Basic tersebut.
- ✓ Pilih direktori kerja anda misalkan D:\VB\Latihan01 dengan mengklik pada kontrol combo box.

- ✓ Simpan form dengan nama Form1, kemudian klik tombol Save. Simpan Project dengan nama Project1, kemudian klik tombol Save.



Gambar 1.14 Save File Dialog

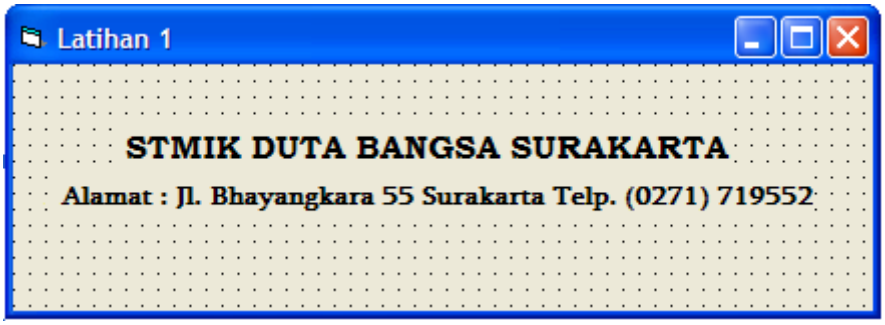
6. Menjalankan dan Menghentikan Program

Langkah-langkah menjalankan/menghentikan program:

- ✓ Klik tombol *Start* ( ) pada *ToolBar* atau dengan menekan tombol F5 atau juga dapat menggunakan Start pada menu *Run*.
- ✓ Sedangkan untuk menghentikan program yang sedang berjalan dapat menggunakan tombol *End* ( ) pada ToolBar atau menggunakan End pada menu Run.

C. Latihan

- ✓ Letakkan object Label pada Form. Atur property Form dan Label seperti terlihat pada gambar di bawah ini :



- ✓ Jalankan program yang Anda buat, lihat hasilnya
- ✓ Hentikan program kemudian simpan program yang Anda buat
- ✓ Untuk menyimpan pilih save As Project pada Menu File. Pada saat project tersebut disimpan ada 2 macam file yaitu :
    - File Project (*.vbp) beri nama prLat01.vbp
    - File Form1 (*.frm) beri nama frLat01.frm

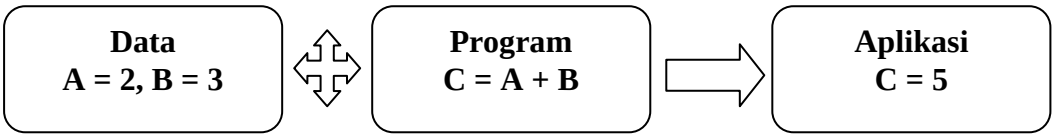
BAB II

ELEMEN-ELEMEN PROGRAM

A. Pendahuluan

Sebelum kita bekerja dengan program visual basic 6.0, perlu kiranya kita memahami terlebih dahulu tentang data beserta tipe-tipenya, konstanta, variable dan seluruh pendukung dalam pembuatan program.

Suatu data sangat terkait erat dengan program, karena data merupakan nilai yang dibutuhkan oleh aplikasi sedangkan program merupakan instruksi yang digunakan untuk mengolah data tersebut. Jadi data dan program merupakan unsur-unsur utama dalam membuat suatu program aplikasi yang sempurna (Subari & Yuswanto, 2008:41).



Gambar 2.1 keterkaitan Data dan Program

B. Materi

1. Tipe Data dalam Visual Basic

Data memiliki tipe yang berbeda-beda dan biasanya data dikelompokkan pada kelompok yang sejenis agar tidak terjadi operasi matematika diantara data yang berbeda jenis, contohnya kita tidak bisa menjumlahkan nilai suatu jarak (Km) dengan Massa (Kg) karena keduanya memiliki tipe data yang berbeda. Begitu juga dalam suatu program, setiap nilai harus dikelompokkan pada jenis-jenis tertentu yang disebut dengan tipe data.

Microsoft Visual Basic menyediakan beberapa tipe data seperti Integer, Long, Single, Double, Currency, String, Byte, Boolean, Date, Object dan Variant.

Tabel 2.1 Tipe Data dalam Visual Basic

Type	Range
Integer	-32768 s/d 32767
Long	-2147483.648 s/d 2147483647
Single	Negatif : -3.40282E38 s/d -1.401298E-45 Positif : 1.401298E-45 s/d 3.402823E38
Double	Negatif : -1.7976931348232E308 s/d -4.94065645841247e-324 Positif : 4.94065645841247e-324 s/d 1.79769313486232E308
Currency	-922337203685477.5808 s/d 922337203685477.5807
String	0 s/d 2 milyar karakter
Byte	0 s/d 255
Boolean	True (benar) atau false (salah)
Date	1 January 100 s/d 31 Desember 9999
Object	Referensi Object
Variant	Null, Error dan tipe seluruh tipe data yang lain

Keterangan :

Integer, Long : tipe data untuk angka bulat

Single, Double : tipe data untuk angka pecahan/desimal

Currency : tipe data untuk angka mata uang

String : tipe data untuk teks

Boolean : tipe data logika (True/False)

Date : tipe data waktu/tangggal

Object : tipe data untuk sebuah objek misalnya gambar

Variant : tipe data variant

2. Variabel dalam Visual Basic

Variabel digunakan untuk menyimpan nilai atau data yang dimiliki program aplikasi yang kita buat. Nilai yang ditampung atau disimpan oleh suatu variabel dapat berubah selama program berjalan. Misalnya kita bisa menyimpan nilai ujian mid di variabel A dan nilai ujian akhir di variabel B dan setiap mahasiswa nilainya pasti berbeda

a. Deklarasi Variabel

Deklarasi variabel harus diletakkan sebelum baris-baris perintah yang menggunakan variabel tersebut. Ada dua cara dalam pendeklarasian variabel, yaitu **Deklarasi Eksplisit** dan **Deklarasi Implisit**.

Deklarasi Eksplisit dilakukan dengan cara menuliskan sebuah kata kunci diikuti nama variabel serta tipe datanya. Aturan penulisan deklarasi variabel dengan cara eksplisit sebagai berikut :

<katakunci> <namavariabel> As <TipeData>

Tabel 2.2 Kata kunci deklarasi Variabel secara eksplisit

Kata Kunci	Penggunaan
Static	Berlaku pada level prosedur
Dim	Berlaku pada level prosedur dan modul
Private	Berlaku pada level modul/form
Public	Berlaku pada level modul dan aplikasi
Global	Berlaku pada level modul dan aplikasi

Contoh deklarasi variabel secara eksplisit adalah :

```
Dim Nama As String
Private Nilai As Integer
Static Jumlah As Integer
Public Alamat As Variant
```

Deklarasi Implisit dilakukan tanpa menggunakan kata kunci. Pada deklarasi implisit, sebuah variabel langsung digunakan disertai sebuah karakter khusus yang menandakan tipe data variabel tersebut.

Tabel 2.3 Beberapa karakter untuk deklarasi Variabel Implisit

Tipe Data	Karakter
Integer	%
Long	&
Single	!
Double	#
Currency	@
String	$

Contoh deklarasi secara implisit :

```
Judul$ = "Cepat Mahir Visual Basic 6.0"
Harga% = 25000
```

Sebaiknya setiap variabel yang digunakan dalam kode program dideklarasikan terlebih dahulu, agar alur jalannya program lebih terkontrol dan mudah dipahami.

b. Aturan Penamaan Variabel

Visual Basic 6.0 tidak memperhatikan penulisan huruf besar atau kecil. Variabel dengan nama **Alamat** akan dianggap sama dengan **ALAMAT** atau **aLaMAt**

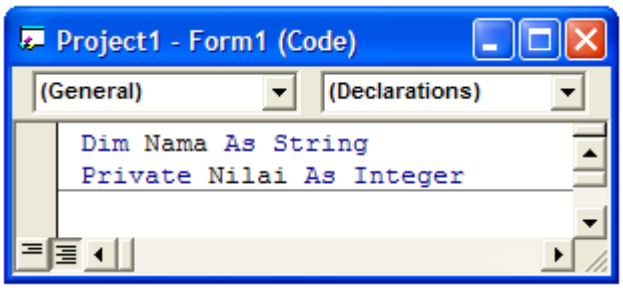
Beberapa aturan yang digunakan dalam penamaan variabel adalah sebagai berikut :

- Harus unik, tidak boleh ada variabel dengan nama sama pada satu ruang lingkup yang sama.
- Tidak boleh lebih dari 255 karakter, tetapi hanya 40 karakter pertama yang dianggap sebagai nama variabel. Karakter sisanya diabaikan.
- Tidak boleh menggunakan spasi, tanda +,-,*,/,<,>,:,=,#,koma dll
- Harus dimulai dari huruf, bukan angka atau karakter lainnya
- Tidak boleh menggunakan reserved word milik Visual Basic 6.0

c. Ruang Lingkup Variabel

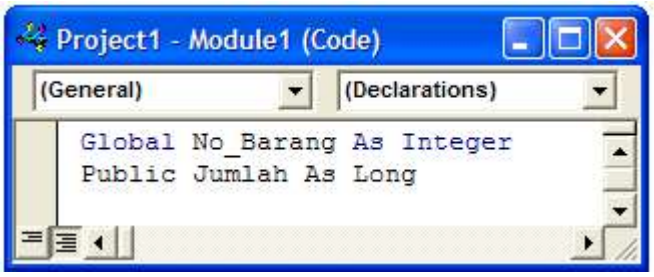
Ada tiga level ruang lingkup variabel, yaitu : **level lokal**, **level form/modul,** dan **level global**.

- Variabel **level lokal** adalah variabel yang hanya dikenali dan dapat digunakan dalam prosedur tempat variabel tersebut dideklarasikan. Prosedur di bagian lain tidak dapat mengkses variabel tersebut. Variabel ini dideklarasikan menggunakan perintah **Dim, Private**, dan **Static.**
- Variabel dengan **level form/modul** dideklarasikan dibagian **general declarations** dalam jendela kode program form/modul. Nilai variabel ini dapat dikenali dan dipakai pada semua prosedur yang ada dalam form/modul tersebut. Deklarasi variabel ini juga menggunakan pernyataan **Dim**. Perbedaan dengan variabel **lokal** adalah tempat dekalrasi variabel tersebut. Contoh deklarasi variabel level form/modul: :



Gambar 2.2 Deklarasi variabel level form/modul

- Variabel **level global** (aplikasi) dideklarasikan di bagian **general declarations** dalam jendela kode program form/modul menggunakan pernyataan **Public** atau **Global**. Variabel ini dapat dipanggil dan dipakai oleh semua form/modul dalam program aplikasi yang kita buat, dan juga dapat dipanggil dan dipakai oleh seluruh prosedur yang ada dalam form/modul tersebut. Contoh deklarasi variabel level global adalah sebagai berikut:



Gambar 2.3 Deklarasi variabel level global (aplikasi)

3. Konstanta dalam Visual Basic

Konstanta adalah sejenis variabel yang nilainya tetap dan tidak dapat diubah selama program berjalan. Ada jua jenis konstanta yaitu konstanta intrinsik dan konstanta yang dibuat pemrogram.

Konstanta intrinsik adalah konstanta yang sudah tersedia secara otomtais dalam Visual Basic 6.0. Contoh konstanta intrinsik adalah vbYes,vbModal, dan lain-lain. Pemrogram juga dapat membuat sebuah konstanta sendiri dengan cara mendeklarasikannya terlebih dahulu.

a. Deklarasi Konstanta

Aturan penulisan deklarasi konstanta adalah sbb:

[<KataKunci> Const <NamaKonstanta> [As <TipeData>] = <nilai>

Kata Kunci : Pilihan kata kunci yang dapat digunakan pada deklarasi konstanta adalah Private dan Public

Nama Konstanta : Nama konstanta yang dideklarasikan

Tipe Data : Tipe data dari konstanta tersebut

Bagian yang diberi tanda kurung siku "[" dan "]" pada aturan penulisan di atas berarti boleh dituliskan, dan boleh juga tidak dituliskan.

Contoh deklarasi konstanta :

```
Const NamaUsaha = "CV. Oryn Cellular"
Public Const Jumlah = 500000
Private Const Total = 5
Const Nama As String = "Dwi Apri Setyorini"
```

b. Ruang Lingkup Konstanta

Konsep ruang lingkup konstanta hampir sama dengan konsep ruang lingkup variabel. Yaitu terdiri dari **level lokal, level form/modul** dan **level global.** Ketiga level tersebut hanya dibedakan berdasarkan kata kunci yang digunakan atau letak deklarasi konstanta.

4. Operator

Operator adalah suatu tanda yang digunakan untuk menghubungkan satu variabel atau konstanta dengan variabel atau konstanta lain dengan tujuan melakukan berbagai manipulasi dan pengolahan data. Pada Ms. Visual Basic 6.0 terdapat bermacam-macam operator :

a. Operator Penugasan (assignment)

Operator penugasan ditimbulkan dengan tanda sama dengan (=) dan berfungsi untuk memasukkan suatu data ke dalam suatu variabel.

Contoh :

```
A = 12
A = A+1
Text1.text = "STMIK Duta Bangsa"
```

b. Operator Aritmatika

Operator aritmatika digunakan untuk melakukan operasi aritmatika. Operator aritmatika mempunyai hirarki paling tinggi dibanding operator

pembanding dan operator logika. Penulisan operator aritmatika dengan hirarki dari paling tinggi ke paling rendah, sebagai berikut :

Tabel 2.4 Operator Aritmatika

Operator	Operasi
^	Pemangkatan
-	Tanda Negatif
* , /	Perkalian & Pembagian
\	Pembagian Integer
Mod	Modulus (Sisa hasil bagi)
+, -	Penambahan dan Pengurangan
+, &	Penggabungan String

Contoh :

```
A = (2^4)*5, B = 25 Mod 4
```

c. Operator Pembanding

Operator pembanding digunakan untuk membandingkan suatu data (ekspresi) dengan data (ekspresi) lain dan menghasilkan nilai logika (boolean) Benar atau Salah. Tentu saja antara kedua data yang dibandingkan harus mempunyai tipe data yang sama. Bentuk dari operator relasional/pembanding seperti pada tabel berikut :

Tabel 2.5 Operator Pembanding

Operator	Operasi
=	Sama dengan
<>	Tidak sama dengan
< , >	Lebih kecil, lebih besar,
<= , >=	lebih kecil sama dengan, lebih besar sama dengan
Like	Mempunyai ciri yang sesuai
Is	Sama referensi objek

Contoh :

```
A <> B , B>C, C<A
```

d. Operator Logika

Operator logika digunakan untuk mengekspresikan satu atau lebih data (ekspresi) logika (boolean) yang menghasilkan data logika baru. Tabel operator logika dengan hierarki dari atas ke bawah adalah sebagai berikut :

Tabel 2.6 Operator Logika

Operator	Keterangan
Not	Tidak
And, Or,Xor	Dan, Atau, Exclusive Or
Eqv	Equivalen
Imp	Implikasi

Contoh :

```
X > 5 And X < 10, X = 3 Or x = 5
```

5. Kontrol Program

Ada banyak perintah di Visual Basic 6.0. yang digunakan untuk mengontrol jalannya program yang akan dibuat. Fungsi kontrol program ini dibentuk dengan logika pemograman yang nantinya akan berguna sebagai validasi terhadap data-data yang masuk maupun yang keluar dari program tersebut.

a. Pencabangan On Error

Pencabangan ini dipakai untuk penangan kesalahan (error) dalam program. Bentuk penulisannya ada tiga macam, yaitu : On Error GoTo <baris>, On Error Resume Next, On Error GoTo 0

- **On Error GoTo <baris>**

  Menyebabkan penanganan error aktif, sehingga jika terjadi kesalahan program maka kesalahan yang terjadi tersebut akan menunjuk ke <baris> untuk proses selanjutnya.

  Contoh :

```
Private Sub Command1_Click()
On Error GoTo Pesan
.......
.......
Pesan:
     MsgBox "Ada Kesalahan Program"
End Sub
```

  Kesalahan apapun yang terjadi, baik kesalahan dalam program maupun kesalahan dalam logika pemograman, maka kesalahan tersebut tidak akan terlihat dimana letak kesalahannya karena kesalahan apapun yang muncul maka kesalahan tersebut akan memunculkan tulisan pada baris <Pesan>

- **On Error Resume Next**

  Menyebabkan jika terjadi kesalahan, program akan melanjutkan ke perintah yang mengikuti (dibawah) perintah yang salah.

  Contoh :

```
Private Sub Command1_Click()
On Error Resume Next
MsgBox "Pesan dikerjakan setelah mengabaikan Error"
End Sub
```

- **On Error GoTo 0**

  Menyebakan perangkap kesalahan yang sebelumnya dipasang menjadi tidak aktif (membatalkan penanganan kesalahan).

  Contoh :

```
Private Sub Command1_Click()
On Error GoTo 0
MsgBox "Jika Error lanjutkan ke baris berikutnya"
End Sub
```

b. Statement End

Statement End dipakai untuk memaksa kontrol program berhenti dari suatu procedure atau suatu blok program. Beberapa contoh bentuk pernyataan End.. adalah sebagai berikut :

- **End**

  Statement End biasanya digunakan untuk mengakhiri penggunaan suatu program, misalkan pada program terdapat suatu tombol command button bercaption “Keluar” maka isi untuk perintah command button kelaur tersebut adalah dengan “End”.

  Contoh :

```
Private sub cmdExit_click()
    End
End sub
```

- **End Function**

  Jika dalam program menggunakan function, terutama function yang akan dibentuk sendiri, maka diakhiri penulisan function tersebut harus ditutup dengan end function.

  Contoh :

```
Private Function Tambah(a,b as Integer) as Integer
     Tambah = a + b
End Function
```

- **End if**

  Ekpresi **End If** digunakan untuk mengakhiri penggunaan ekpresi **If.. Then… Else…,**setiap perintah If harus ditutup atau diakhiri dengan End If

  Contoh :

```
Private sub cmdOK_click()
If  Text1.Text = 1 then
     Msgbox "Nilainya Satu"
Else
     Msgbox " Nilainya selain Satu"
End If
End sub
```

  Ekspresi If di atas akan menunjukkan jika nilai yang terdapat dalam object TextBox bernilai 1, maka akan muncul tulisan pesan "Nilainya Satu" selain itu muncul tulisan "Nilainya selain Satu".

- **End Property**

  End property biasanya digunakan ketika akan bekerja dengan menggunakan class.

- **End Select**

  Salah satu pernyataan pencabangan yang bersyarat adalah menggunakan Select Case, diakhir pernyataan tersebut jangan lupa dituliskan End Select yang menyatakan akhir dari pencabangan tersebut.

  Contoh :

```
Select Case Text1.Text
Case "1"
    MsgBox "Data anda satu"
Case "2"
    MsgBox "Data anda dua"
End Select
```

- **End Sub**

  Setiap objek yang diletakkan dalam sebuah form akan membentuk suatu modul atau procedure sendiri, proses selanjutnya tergantung dari event yang akan dilakukan dengan objek tersebut. Diakhir pendeklarasian sebuah modul harus ditutupi dengan End Sub.

  Contoh :

```
Private Sub Command1_Click()
        ……
        ……
        ……
End Sub
```

- **End Type**

  Pendeklarasian End Type digunakan jika mendeklarasikan sekumpulan data bertipe record data, dan disetiap akhir pendeklarasian tersebut harus diakhiri dengan End Type.

  Contoh :

```
Type DATASISWA
     NIM  as string
     NAMA as string
     Alamat as string
End Type
```

- **End With**

  Pendeklarasian End With digunakan jika kita ingin mengakhiri penggunaan with diawal sebuah pendeklarasian, perintah with dan end with dapat digunakan untuk menyingkat suatu penulisan objek yang berulang-ulang. Contoh berikut menyatakan object textbox yang digunakan dan beberapa atribut yang digunakan dalam object tersebut.

  Contoh :

```
With Text1
     .Text = "Sani"
     .Font = Arial
     .FontBold = True
     .FontSize = 15
End With
```

c. Komentar Program

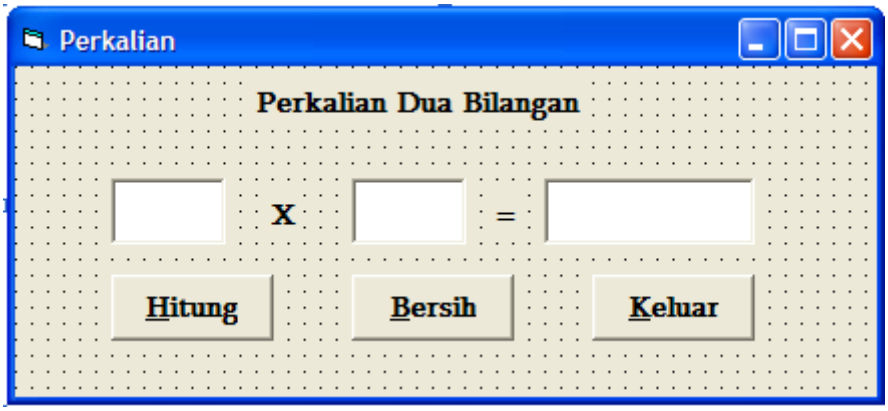
Komentar dapat ditambahkan pada suatu baris program dengan menuliskan tanda petik satu (‘) didepan statement yang ingin dinyatakan sebagai suatu komentar, sifat komentar ini tidak akan dikerjakan dan hanya sebagai komentar bagi program dan visual basic tidak akan menganggapnyanya sebagai kode sehingga tidak akan dijalankan.

Contoh :

```
Private Sub Command1_click()
    A = 3        'Set nilai A = 3
    B = 5        'Set Nilai B = 5
    C = A + B    'Nilai A dan B akan ditambahkan
                 'dan disimpan dalam variabel C.
End Sub
```

6. Contoh Program

a. Kita akan mencoba membuat program perkalian dua bilangan. Letakkan beberapa object yang ada dalam form, atur beberapa setting property yang ada seperti terlihat pada gambar di bawah ini:



Kontrol	Property	Setting Value
Form	Name	frKali
	Caption	Perkalian
Label	Name	lblJudul
	Caption	Perkalian Dua Bilangan
Label	Name	lblKali
	Caption	X
Text	Name	txtBil1
	Text	Blank
Text	Name	txtBil2
	Text	Blank
Text	Name	txtHasil
	Text	Blank
Command	Name	cmdHitung
	Caption	&Hitung
Command	Name	cmdBersih
	Caption	&Bersih
Command	Name	cmdKeluar
	Caption	&Keluar

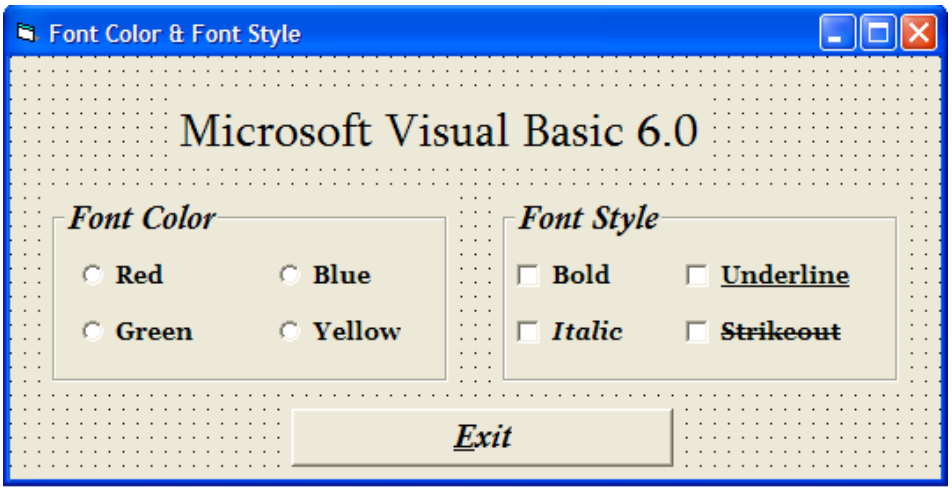
Ketikkan kode program seperti berikut:

```
Private Sub cmdBersih_Click()
'Mengosongkan isi TextBox
txtBil1.Text = ""
txtBil2.Text = ""
txtHasil.Text = ""
End Sub

Private Sub cmdHitung_Click()
'Deklarasi Variabel Lokal
Dim a As Single
Dim b As Single
Hasil As Single

a = txtBil1.Text
b = txtBil2.Text
Hasil = a * b
txtHasil.Text = Hasil
End Sub

Private Sub cmdKeluar_Click()
'Mengakhiri Program
End
End Sub
```

Simpan program yang Anda buat dengan nama :

- File Project (*.vbp) beri nama prCoba01.vbp
- File Form1 (*.frm) beri nama frCoba01.frm

b. Membuat program untuk memilih font color dan font style. Letakkan beberapa object yang ada dalam form, atur beberapa setting property yang ada seperti terlihat pada gambar di bawah ini:



Kontrol	Property	Setting Value
Form	Name	frFont
	Caption	Font Color & Font Style
Label	Name	lblTeks
	Caption	Microsoft Visual Basic 6.0
Option	Name	optRed
	Caption	Red
Option	Name	optGreen
	Caption	Green
Option	Name	optBlue
	Caption	Blue
Option	Name	optYellow
	Caption	Yellow

Kontrol	Property	Setting Value
Check	Name	chkBold
	Caption	Bold
Check	Name	chkItalic
	Caption	Italic
Check	Name	chkUnder
	Caption	Underline
Check	Name	chkStrike
	Caption	Strikeout
Frame	Caption	Font Color
Frame	Caption	Font Style
Command	Name	cmdExit
	Caption	&Exit

Ketikkan kode program seperti berikut:

```
Private Sub optRed_Click()
'Membuat teks bewarna merah
lblTeks.ForeColor = vbRed
End Sub

Private Sub chkBold_Click()
'Membuat teks Tebal
If chkBold.Value = 1 Then
    lblTeks.FontBold = True
Else
    lblTeks.FontBold = False
End If
End Sub

'Untuk pilihan font color yang
'lain silakan Anda coba membuat
'kode programnya

'Untuk pilihan font style yang
'lain silakan Anda buat kode
'programnya
```

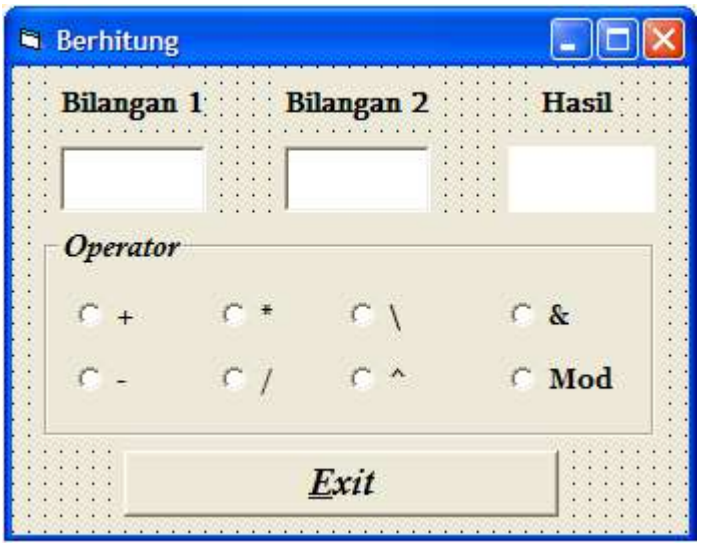
Simpan program yang Anda buat dengan nama:

a. File Project (*.vbp) beri nama prCoba02.vbp

b. File Form1 (*.frm) beri nama frCoba02.frm

C. Latihan

- Buat program penghitungan dengan menggunakan beberapa operator. Atur tampilan dan propertinya seperti di bawah ini:



- Simpan program yang Anda buat dengan nama:
  - File Project (*.vbp) beri nama prLat02.vbp
  - File Form1 (*.frm) beri nama frLat02.frm

BAB III
PROPERTY, METHOD DAN EVENT

A. Pendahuluan

Pemograman Visual Basic menggunakan suatu teknik pemrograman OOP (Object Oriented Programming) yaitu suatu teknik pemrograman yang memodelkan program sebagai sekumpulan objek yang saling memiliki hubungan. Setiap kontrol (objek) memiliki properti, method dan event yang berbeda-beda.

B. Materi

1. Memahami Property, Method dan Event

a. Properti

Properti adalah atribut-atribut yang melekat pada sebuah kontrol (objek) yang biasanya merupakan karakteristik penampilannya seperti warna, jenis huruf, ukuran dan sebagainya

Properti sebuah kontrol (objek) dapat diubah pada saat memprogram (menggunakan jendela properti) atau pada saat program dijalankan (menggunakan kode-kode program pada jendela kode).

Contoh : Command1**.Caption**="OK"

b. Method

Method adalah aksi atau perbuatan yang bisa dimiliki oleh kontrol (objek) sehingga user (programmer) dapat memakainya untuk memanipulasi sesuatu. Method tergantung dari instruksi yang diberikan oleh programmer melalui penulisan kode.

Contoh : Command1**.Click**

c. Event

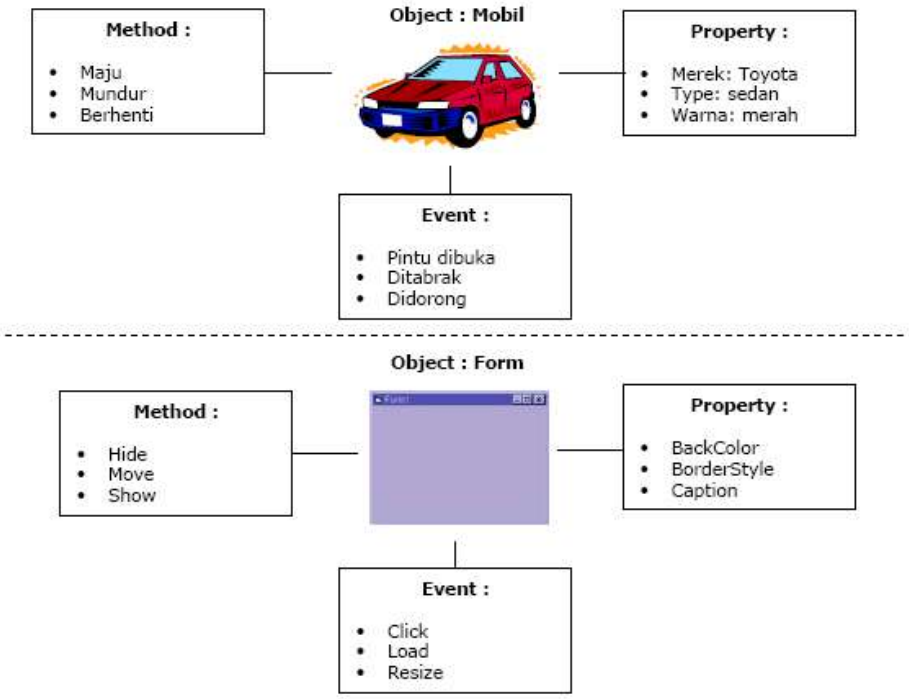
Event adalah kejadian atau segala sesuatu yang dapat dialami oleh sebuah objek. Suatu kontrol (objek) dapat memiliki banyak event.

Contoh : Command1**_Click( )**

Secara ringkasnya dapat dijelaskan sebagai berikut :

- **Property** : karakteristik yang dimiliki object
- **Method** : aksi yang dapat dilakukan oleh object
- **Event** : kejadian yang dapat dialami oleh object

Sebagai ilustrasi anda dapat menganggap sebuah mobil sebagai obyek yang memiliki property, method dan event. Perhatikan gambar berikut :



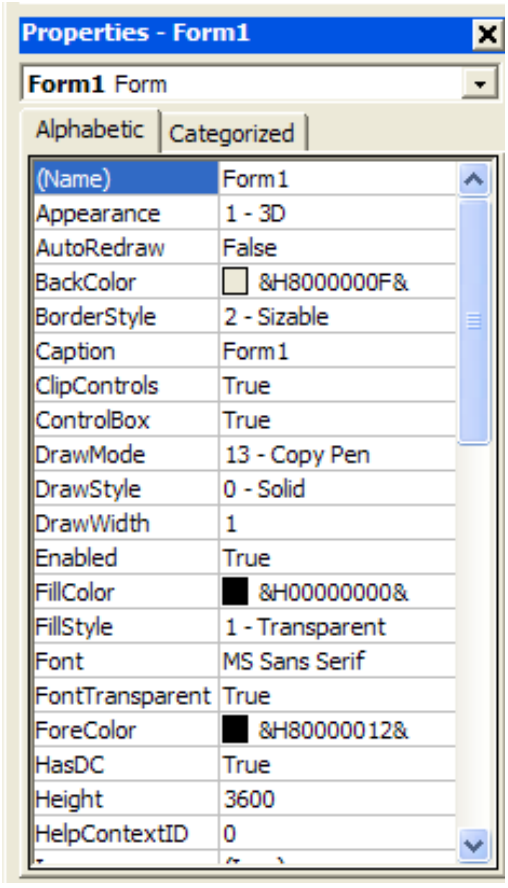
Gambar 3.1 Ilustrasi untuk mengambarkan Property, Method dan Event

2. Property

Form merupakan objek utama dalam permrograman Visual Basic, sebab form merupakan tempat dimana objek-objek lain seperti kontrol diletakkan seperti terlihat pada gambar 3.2. Form dan Kontrol merupakan objek-objek yang ada dalam Visual Basic yang memiliki properti, metode dan dapat menangkap suatu kejadian (*event*). Pada bagian ini akan dibahas tentang properti, metode (*methods*), dan kejadian yang berhubungan dengan objek-objek tersebut.

Gambar 3.2 Tampilan Form dari program Perkalian

Visual Basic mendukung properti-properti yang hanya dimiliki oleh sejumlah objek tertentu saja. Akan tetapi ada beberapa properti-properti yang sifatnya umum yaitu properti-properti ini hampir dimiliki oleh semua objek yang ada dalam Visual Basic. Properti-properti umum tersebut antara lain properti *name, top, left, height, weight, Foreground, Background, Font, Caption, Text, Enable, Visible, TabStop, TabIndex,* dan properti-properti umum lainnya.



Gambar 3.3 Jendela Properties untuk Form

a. Properti Name

Semua objek dalam Visual Basic mempunyai properti *Name*. Properti *Name* digunakan untuk memberikan identitas pada objek yang dipakai. Secara *default* sebuah form mempunyai properti *Name* "Form1", "Form2", dan seterusnya, lihat gambar 3.3. Dengan mengubah nilai properti ini dengan nama lebih deskripsi akan memudahkan kita untuk mengingat nama dari objek-objek yang dipakai dalam project yang bersangkutan.

Seorang programmer Visual Basic biasanya menggunakan prefik untuk penamaan sebuah kontrol atau form. Penggunaan prefik memudahkan seorang programer untuk mengetahui kontrol yang bersangkutan. Dengan kata lain penggunaan prefik adalah untuk membedakan sebuah kontrol dengan kontrol yang lain seperti nama *frmLatihan* untuk sebuah kontrol form dan *lblNama* untuk sebuah kontrol label. Tabel 3.1 berisi daftar penamaan prefik yang biasanya digunakan dalam Visual Basic.

Tabel 3.1 Daftar Penamaan Prefik

Prefik	Nama Elemen
cbo	Combo Box
chk	Check Box
cmd	Command
dat	Data
dir	Directory List Box
drv	Drive List Box
fil	File List Box
fra	Frame
fr	Form
grd	Grid
hsb	Horizontal ScrollBar
img	Image
lbl	Label

Prefik	Nama Elemen
lin	Line
lst	List Box
mnu	Menu
mod	Module
ole	OLE
opt	Option
pic	Picture
res	Resource
shp	Shape
tmr	Timer
txt	Text Box
typ	User-Defined Data Type
vsb	Vertical Scrool Bar

b. Properti Left, Top, Width, Height

Semua objek-objek yang terlihat (*visible*) memiliki properti *Left, Top, Width,* dan *Height*. Properti-properti ini digunakan untuk menempatkan posisi dan ukuran suatu objek. Nilai dari properti ini selalu relatif terhadap objek kontainernya (tempat dimana objek tersebut menempel) dan satuan ukuran default yang digunakan adalah *twips*.

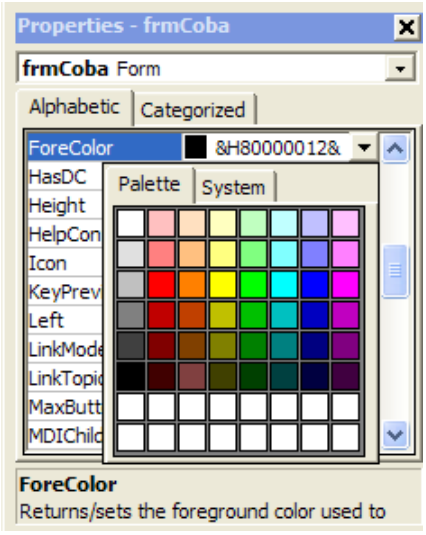
Properti-properti tersebut dapat diubah melalui jendela properti (*properties window*) seperti terlihat pada gambar 3.3 dengan memasukkan suatu nilai numerik pada saat waktu merancang atau dengan memindahkan atau mengubah ukuran form secara interaktif dengan menggunakan mouse. Selain itu pengubahan properti juga dapat dilakukan pada saat program dijalankan (*run time*) melalui kode seperti contoh dibawah ini :

```
'Membuat lebar frmCoba menjadi 2 kali lebar semula
frmCoba.Width = frmCoba.Width * 2

'Membuat tinggi frmCoba menjadi 3 kali tinggi semula
frmCoba.Height = frmCoba.Height * 3

'Meletakkan frmCoba di sudut kiri atas
frmCoba.Left = 0
frmCoba.Top = 0
```

Tidak semua kontrol dalam Visual Basic memiliki properti Left, Top, Width, dan Height. Sebagai contoh Timer tidak memiliki properti – properti tersebut.

c. Properti ForeColor dan BackColor

Hampir semua kontrol-kontrol Visual Basic juga mendukung kedua properti *ForeColor* dan *BackColor*, lihat gambar 3.4. Properti *ForeColor*t digunakan untuk mempengaruhi warna tulisan dan properti *BackColor* digunakan untuk memilih warna dasar dari objek. Dalam kasus tertentu, properti tersebut tergantung pada properti lain. Sebagai contoh mengganti warna dasar (*background*) tidak akan berpengaruh jika nilai dari properti *backstyle* bernilai *0-Transparant*.



Gambar 3.4 Properti ForeColor

Ada 2 macam warna yang dapat digunakan pada properti *BackColor* dan *ForeColor* yaitu Standart Color (tab System) dan Custom Color (tab Palette). Kedua jenis warna tersebut dapat dipilih dengan menggunakan tab, namun sebaiknya digunakan warna standar (*standart color*), kecuali ada alasan yang kuat untuk menggunakan warna khusus (*custom color*). Alasan penggunaan warna standar adalah warna ini akan bekerja dengan baik pada semua komputer dengan Sistem Operasi *Windows*.

Visual Basic telah menyediakan sejumlah konstanta simbolik yang menyatakan sebuah warna. Konstanta tersebut dapat dilihat pada tabel 3.2 atau dapat juga dilihat pada jendela *Object Browser*.

Ada beberapa cara untuk mengisi nilai properti *BackColor* dan *ForeColor* melalui kode:

```
'Mengatur property ForeColor dan BackColor
'Pada lblCoba

lblCoba.ForeColor = vbHighlightText
lblCoba.BackColor = vbHighlight
```

Properti ini juga dapat diisi dengan menggunakan konstanta simbol yang lain seperti *vbBlack*, *vbBlue*, *vbCyan*, *vbGreen*, *vbMagenta*, *vbRed*,

vbWhite, and *vbYellow* atau menggunakan nilai konstanta desimal/heksadesimal.

```
'Ketiga kode program di bawah ini mempunyai hasil sama
txtCoba.BackColor = vbCyan
txtCoba.BackColor = 16776960
txtCoba.BackColor = &HFFFF00
```

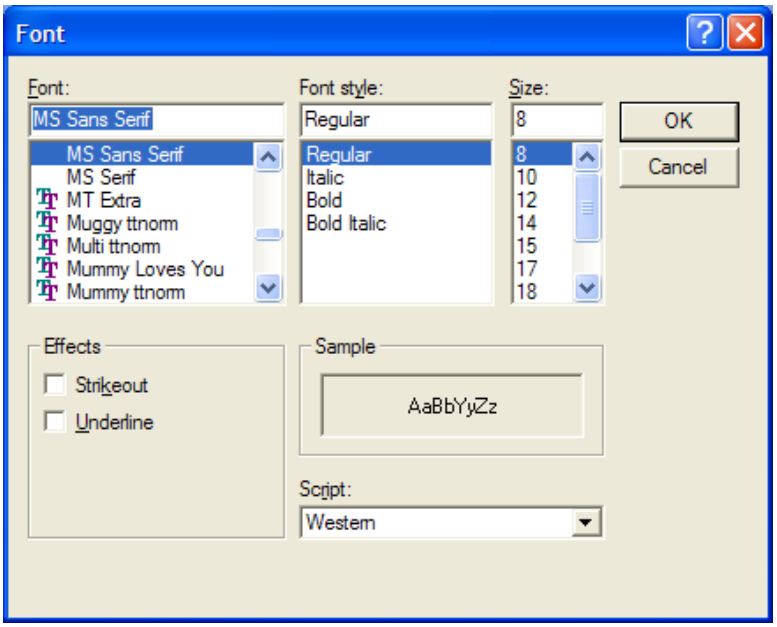
Atau dapat pula digunakan fungsi RGB yang membangun warna berdasarkan warna Merah (R), Hijau (G). dan Biru(B) (Red, Green, dan Blue) atau fungsi QBColor yang merupakan fungsi dari Quick Basic.

Tabel 3.2 Nilai Konstanta Warna

Konstatan	Nilai Heksadesimal	Keterangan
vb3DDKShadow	&H80000015	Darkest shadow
vb3Dface	&H8000000F	Dark shadow color for 3-D display elements
vb3Dhighlight	&H80000014	Highlight color for 3-D display elements
vb3Dlight	&H80000016	Second lightest of the 3-D colors after vb3Dhighlight
vb3Dshadow	&H80000010	Color of automatic window shadows
vbActiveBorder	&H8000000A	Active window border color
vbActiveTitleBar	&H80000002	Active window caption color
vbActiveTitleBarText	&H80000009	Text color in active caption, size box, scroll bar arrow box
vbApplicationWorkspace	&H8000000C	Background color of multiple-document interface (MDI) applications
vbButtonFace	&H8000000F	Face shading on command buttons
vbButtonShadow	&H80000010	Edge shading on command buttons
vbButtonText	&H80000012	Text color on push buttons
vbDesktop	&H80000001	Desktop color
vbGrayText	&H80000011	Grayed (disabled) text
vbHighlight	&H8000000D	Background color of items selected in a control
vbHighlightText	&H8000000E	Text color of items selected in a control
vbInactiveBorder	&H8000000B	Inactive window border color
vbInactiveCaptionText	&H80000013	Color of text in an inactive caption
vbInactiveTitleBar	&H80000003	Inactive window caption color
vbInactiveTitleBarText	&H80000013	Text color in inactive window caption, size box, scroll bar arrow box
vbInfoBackground	&H80000018	Background color of ToolTips
vbInfoText	&H80000017	Color of text in ToolTips
vbMenuBar	&H80000004	Menu background color
vbMenuText	&H80000007	Text color in menus
vbScrollBars	&H80000000	Scroll bar gray area color
vbTitleBarText	&H80000009	Text color in active caption, size box, scroll bar arrow box
vbWindowBackground	&H80000005	Window background color
vbWindowFrame	&H80000006	Window frame color
vbWindowText	&H80000008	Text color in windows

d. Properti Font

Pada waktu merancang, Properti Font dapat diisi dengan menggunakan *Font Dialog* seperti terlihat pada gambar 3.5.



Gambar 3.5 Kotak dialog Font

Cara lain untuk mengubah nilai properti bisa dilakukan melalui kode program, seperti contoh di bawah ini:

```
txtCoba.Font.Name = "Sylfaen"
txtCoba.Font.Size = 12
txtCoba.Font.Bold = True
txtCoba.Font.Italic = True
```

Font merupakan objek gabungan (*Compound Object*) yang masih mempunyai propeti-properti yang terpisah yaitu *Font Name, Font Style, Font Size* dan *Effects*

e. Properti Caption dan Text

Properti *Caption* digunakan untuk menentukan kata atau kalimat yang ditampilkan pada sebuah kontrol, seperti judul dari *Form* dan kalimat pada sebuah *Label*. Sedangkan Properti *Text* sama seperti properti *Caption* hanya saja kalimat dalam properti Text dapat diganti. Tidak ada kontrol yang mendukung kedua properti ini secara bersamaan. Kontrol *Label*, *CommandButton*, *CheckBox, OptionButton*, *Data*, dan *Frame* mendukung properti *Caption* sedangkan kontrol *TextBox*, *ListBox*, dan *ComboBox* mendukung properti *Text*. Khusus untuk properti *Caption* mendukung penggunaan karakter ampersand (&) untuk menentukan *hotkey* atau *shortcut* dari kontrol.

```
'Kedua kode program di bawah ini mempunyai hasil sama
txtCoba.Text = Text1.Text
txtCoba = Text1
```

f. Properti Enabled dan Visible

Secara default nilai dari properti *Enabled* dan*Visible* adalah True, tetapi mungkin pada saat program dijalankan mungkin sebuah kontrol ingin disembunyikan (*hide*) atau ditampilkan dengan keadaan tidak bisa digunakan (*disabled*). Nilai properti *Enable* dan *Visible* dapat diubah melalui kode program seperti terlihat pada kotak berikut ini.

```
'Membuat txtCoba disembunyikan
txtCoba.Visible = False

'Membuat txtCoba tidak bisa digunakan
txtCoba.Enabled = False
```

3.* Metode Umum (*Common Methods)

Pada bagian ini akan dijelaskan metode-metode yang sering digunakan antara lain Metode *Move*, *Refresh*, *SetFocus*.

a. Metode *Move*

Jika suatu kontrol mempunyai properti *Left*, *Top*, *Width*, dan *Height*, maka kontrol tersebut pasti juga mendukung metode *move*. Metode ini digunakan untuk mengubah properti *Left*, *Top*, *Width*, dan *Height* dengan menggunakan sebuah operasi tunggal. seperti terlihat pada kotak dibawah ini.

```
'Lebar Form menjadi dua kali lipat dan
'posisinya berada pada pojok kiri atas.
'Syntaxnya : Move(Left As Single, [Top], [Width], [Height])

frmCoba.Move 0, 0, frmCoba.Width * 2, frmCoba.Width * 2
```

b. Metode *Refresh*

Metode *Refresh* merupakan metode yang digunakan untuk menggambar kembali suatu kontrol. Pada keadaan normal biasanya kita tidak perlu memanggil metode ini, sebab Visual Basic secara otomatis akan

menyegarkan kembali penampilan suatu kontrol ketika sudah diubah. Metode ini digunakan jika suatu kontrol ingin segera disegarkan kembali tanpa menunggu proses lain yang masih harus dilakukan. Penggunaan metode tersebut dalam kode program dapat dilihat pada kotak dibawah ini.

c. Metode *SetFocus*

Metode SetFocus digunakan untuk memindahkan fokus dari input ke suatu kontrol tertentu. Metode digunakan untuk memodifikasi nilai TabOrder yang sudah ditetapkan sebelumnya.

4. Kejadian (*Event*)

Selain Properti dan Metode, Visual Basic juga menyediakan sejumlah kejadian (*events*). Berikut ini merupakan kejadian-kejadian umum yang disediakan:

- Kejadian *Click* dan *DoubleClick*

  Kejadian *Click* dan *DoubleClick* terjadi pada saat seorang pengguna melakukan *click* atau *double click* pada sebuah kontrol.

- Kejadian *Change*

  Kejadian *Change* dibangkitkan pada saat isi dari suatu kontrol mengalami perubahan.

- Kejadian G*otFocus* dan *LostFocus*

  Kejadian *GotFocus* akan dibangkitkan pada saat sebuah kontrol menerima fokus input, sedangkan *LostFocus* merupakan kebalikan dari kejadian *GotFocus*, dibangkitkan pada saat kontrol memindahkan fokusnya menuju ke kontrol yang lain.

- Kejadian *KeyDown*, *KeyUp*, dan *KeyPress*

  Kejadian *KeyDown*, *KeyUp*, dan *KeyPress* merupakan kejadian yang berhubungan dengan *keyboard*. Masing-masing kejadian dibangkitkan pada saat *keyboard* di tekan ke bawah, dilepas, dan ditekan.

- Kejadian *MouseDown*, *MouseUp*, dan *MouseMove*

  Kejadian-kejadian ini dibangkitkan pada saat mouse di-*click*, dilepas, dan digerakkan di atas sebuah control

- Activate

  Event ini terjadi bila sebuah form menjadi window yang aktif.
- Deactive

  Event ini terjadi ketika anda berpindah dari satu form ke form lain.
- GotFocus

  Event ini terjadi bila sebuah objek menjadi satu-satunya focus. Sebuah form dapat membuat objek focus bila form tersebut menjadi window aktif.
- Lostfocus

  Event ini terjadi bila sebuah objek kehilangan focus karena ada objek lain yang mendapatkan focus.
- Load

  Event ini terjadi bila sebuah form dibuka atau dipanggil.
- Unload

  Event ini terjadi bila sebuah form ditutup.
- Resize

  Event ini terjadi bila sebuah form diubah ukurannya.
- Timer

  Event ini terjadi ketika event lain dijalankan sebelum event timer ini. Event ini hanya berhubungan dengan control timer.

5. Contoh Program

Kita akan mencoba membuat program yang meminta user untuk menekan tombol Yes sampai dapat. Letakkan beberapa object yang ada dalam form, atur beberapa setting property yang ada seperti terlihat pada gambar di bawah ini:

Kontrol	Property	Setting Value
Form	Name	frYes
	Caption	Klik Yes
Label	Caption	Are you handsome??
Image	Picture	Gambar bebas
Command	Name	cmdYes
	Style	1-Graphical
	Picture	Gambar bebas

Ketikkan kode program berikut ini :

```
'Kode program berikut dijalankan pada saat user
'menggerakkan pointer di atas cmdYes
'cmdYes akan bergerak secara random

Private Sub cmdYes_MouseMove(Button As Integer, Shift As
Integer, X As Single, Y As Single)
cmdYes.Left = Int(Rnd * (Me.Width - cmdYes.Width))
cmdYes.Top = Int(Rnd * (Me.Height - cmdYes.Height))
End Sub
```

C. Latihan

Buat program yang bisa merubah warna tombol *(command).* Misalnya, ketika pointer berada di atas tombol maka tombol berwarna kuning, dan ketika pointer tidak berada di atas tombol (meninggalkan tombol) maka warna tombol menjadi biru.

BAB IV
KONTROL INTRISIK

A. Pendahuluan

Dalam Visual Basic yang dimaksud dengan kontrol-kontrol intrinsik adalah kontrol-kontrol yang tampil pada Toolbox saat Visual Basic pertama kali dijalankan. Disamping kontrol-kontrol intrinsik tersebut, Visual Basic juga mempunyai kontrol-kontrol tambahan yang disebut *Microsoft ActiveX Control* (*OCX controls*). Meskipun OCX mempunyai banyak keunggulan dibanding kontrol intrinsik, namun penggunaan kontrol intrinsik tetap memiliki keunggulan dibanding OCX antara lain:

- Kontrol intrinsik disertakan pada file MSVBVM60.DLL yang terdapat pada setiap Visual Basic sehingga tidak memerlukan tambahan file yang lain.
- Kontrol intrinsik ditampilkan lebih cepat dibanding OCX pada saat program dieksekusi (*run time*).

B. Materi

1. Command

Command button dapat digunakan dengan mudah yaitu hanya dengan diletakkan pada form dan memberikan *caption* (properti *caption*) dan nama (properti *name*) yang sesuai. Kode untuk Prosedur kejadian klik (*Click event procedure*) perlu ditulis agar sebuah command button dapat berfungsi (menerima kejadian ketika ditekan).

```
Private Sub cmdTest_Click()
  ' Menyimpan data, menutup form yang aktif.
  Call SaveDataToDisk 'Memanggil prosedur SaveDataToDisk
  Unload Me ‘menutup form
End Sub
```

Selain kejadian klik (*click event*), *command* juga menerima kejadian dari *keyboard* dan *mouse* (seperti *KeyDown*, *KeyPress*, *KeyUp*, *MouseDown*, *MouseMove*, *MouseUp*, tetapi tidak menerima kejadian *DblClick*).

2. Label

Kontrol label digunakan untuk memberikan keterangan untuk kontrol-kontrol lain seperti *TextBox*, *ListBox*, dan *ComboBox*. Biasanya properti *caption* pada kontrol tersebut diisi dengan kalimat yang sesuai dan dilengkapi karakter *ampersand*(&) untuk memberikan *hot key*. Selain properti *caption*, properti lain yang cukup berguna adalah *BorderStyle* (digunakan untuk menampilkan kontrol label secara 3 Dimensi) dan *Alignment* (Jika align dari caption tampil pada pinggir kiri, kanan, atau tengah).

Jika *caption* sebuah label terlalu panjang, maka properti *WordWarp* dapat diberi nilai true sehingga label tersebut memiliki lebih dari satu baris. Atau juga dapat mengubah nilai properti *AutoSize* menjadi *true* agar kontrol secara otomatis mengubah ukurannya sesuai dengan panjang *caption*.

3. TextBox

Kontrol *TextBox* digunakan untuk menerima masukan dari pengguna. Pengguna dapat memasukan data berupa angka, huruf, dan karakter-karakter khusus. Pada umumnya kontrol tersebut diletakkan di sebelah kontrol label yang berguna untuk memberikan keterangan. Setelah kontrol TextBox tersebut diletakkan pada form, umumnya properti *text* pada kontrol tersebut dihapus. Selain properti *text*, juga terdapat properti *multiline* yang digunakan agar *Textbox* dapat menerima masukan karakter yang panjang. Seperti *label*, *TextBox* juga mendukung properti *alignment*.

Jika sebuah kontrol *TextBox* menerima masukan yang terbatas panjangnya, maka properti *maxlength* dapat diberi nilai yang menunjukan panjang maksimum yang diperbolehkan.

Jika kontrol *TextBox* menerima masukan berupa *password*, maka properti *passwordChar* dapat diisi dengan dengan karakter tertentu, biasanya karakter asterik(*).

4. CheckBox

Kontrol *Checkbox* mempunyai banyak kegunaan ketika suatu dialog menawarkan pilihan. Memungkinkan user memilih lebih dari satu pilihan.

Ketika kontrol tersebut diklik maka kontrol tersebut menunjukkan sebuah keadaan benar atau salah (True atau False).

Gambar 4.1 CheckBox

Peletakan kontrol tersebut pada sebuah form biasanya dilakukan dengan mengisi properti Caption untuk menjelaskan keadaan dari kontrol. Sedangkan kejadian (Event) yang penting dari kontrol tersebut adalah kejadian klik (*click event*).

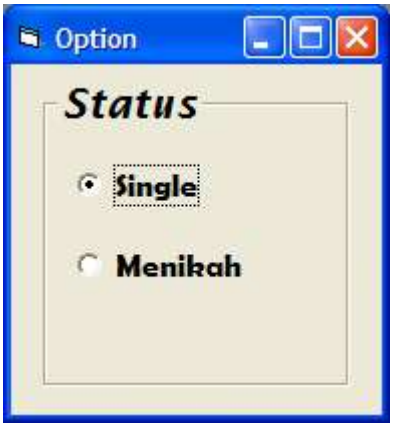
Sebagai contoh ketika sebuah checkbox diklik, maka akan memberikan dampak pada kontrol yang lain. Berikut contoh penggunaan pada suatu kode

```
Private Sub chkSop_Click()
      fraTest.Enabled = (chkSop.Value = 1)
End Sub
```

CheckBox memiliki nilai 1 dan 0 (1 = True, 0 = False). Ketika CheckBox dipilih (dicentang) maka akan bernilai 1 dan sebaliknya.

5. Option

Kontrol Option selalu digunakan dalam suatu group berjumlah 2 atau lebih yang menawarkan pilihan yang bersifat *mutually exclusive* (hanya diperbolehkan untuk memilih salah satu pilihan dari beberapa opsi yang ditawarkan), lihat gambar 4.2.



Gambar 4.2 Option

Apabila sebuah *Option* dipilih, maka *Option* yang lain dalam group yang sama tidak akan terpilih. *Option* yang dipilih bernilai 1, dan yang tidak dipilih bernilai 0. *Option* dalam satu group tersebut biasanya diletakkan pada sebuah *frame* yang sama.

Contoh dalam pemrogramannya seperti berikut ini:

```
' Jika status single dipilih, mendapat tunjangan 300000
' Jika status menikah dipilih, mendapat tunjangan 500000

If optSingle.Value = 1 Then
    tunjangan = 300000
ElseIf optNikah.Value = 1 Then
    tunjangan = 500000
End If
```

6. Frame

Kontrol *Frame* mempunyai fungsi yang mirip dengan kontrol *label* digunakan untuk memberikan keterangan pada kontrol lainnya. Bedanya, kontrol *frame* juga digunakan sebagai kontainer (tempat) dari kontrol-kontrol yang lain. Contoh kontrol frame, lihat gambar 4.2.

Pada umumnya *frame* diletakkan pada *form* dan properti *caption*-nya diberi nilai untuk memberikan keterangan tentang kontrol-kontrol yang menempel. Setelah kontrol *frame* diletakkan, kontrol anak (*child control*) dari *frame* tersebut dapat diletakkan secara langsung diatas frame tersebut. Alternatif yang lain adalah dengan membuat kontrol-kontrol yang sudah ada menjadi kontrol anak dari sebuah frame dengan cara memilih kontrol-kontrol

tersebut dengan melakukan klik disertai dengan menekan tombol *ctrl* kemudian mengambilnya (*cut : ctrl+x*) dan meletakkan (*paste : ctrl+v*) diatas kontrol *frame*.

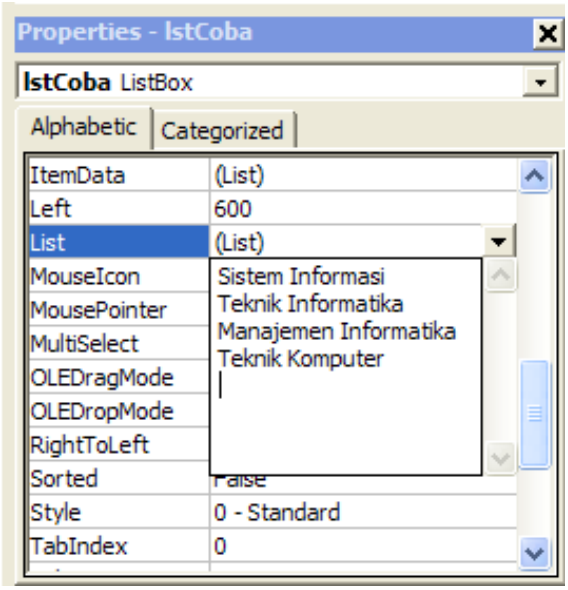
Kontrol *frame* mempunyai 2 karakteristik yang menarik yaitu :

- Jika *Frame* kita gerakkan, maka semua kontrol didalamnya akan mengikuti.
- Properti *enable* dan *visible* dari kontrol *Frame* akan mempengaruhi kontrol didalamnya. Dengan kata lain apabila suatu *Frame* diset *enable : false* (biasa disebut : *disable*), maka semua kontrol yang ada didalamnya secara otomatis akan *disable* pula.

7. List Box

Ketika sebuah Kontrol *ListBox* diletakkan pada sebuah *form*, maka beberapa properti dari kontrol tersebut harus diberikan seperti atribut *sorted* agar secara otomatis item yang ada didalamnya diurutkan berdasarkan alpabetik.

Jika item-item yang harus muncul pada kontrol *listbox* sudah diketahui pada saat *design time*, item-item tersebut dapat dimasukkan langsung melalui properti *list*. Cukup mengetikkan isinya, untuk berpindah ke baris berikutnya tekan Ctrl+Enter.



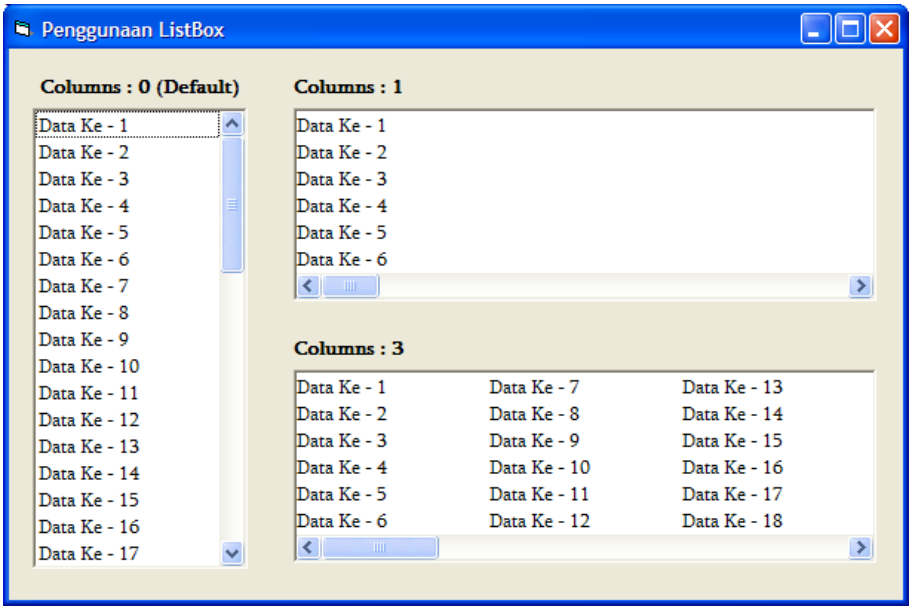
Gambar 4.3 Properti List dari Kontrol ListBox

Namun hal ini tidak menutup kemungkinan untuk memasukkan item data melalui kode program seperti contoh kode dibawah ini :

```
lstCoba.AddItem "Sistem Informasi"
lstCoba.AddItem "Teknik Informatika"
lstCoba.AddItem "Manajemen Informatika"
lstCoba.AddItem "Teknik Komputer"
```

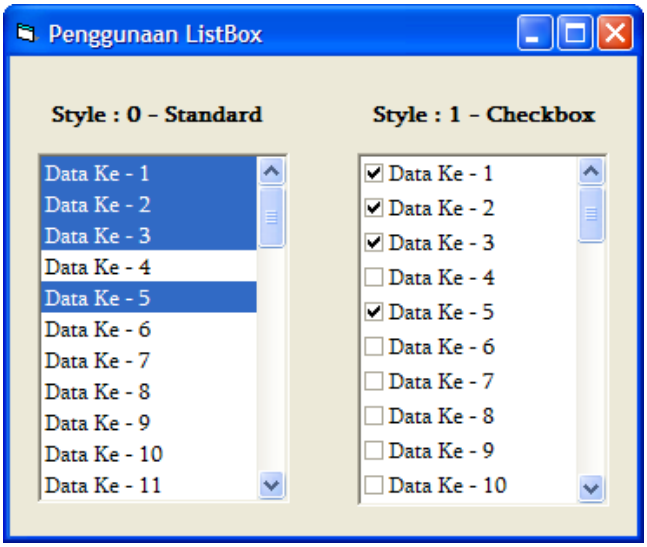
Properti *columns* digunakan untuk menentukan tipe-tipe yang berbeda dari *listbox* dengan beberapa kolom dan sebuah *horizontal scrool bar* pada border sebelah bawah atau sebuah kolom dan sebuah *vertical scroll bar* pada sisi sebelah kanan.

Pada gambar 4.4 ditunjukkan perbedaan penggunaan nilai properti *columns* 0, 1 dan 3



Gambar 4.4 Kontrol ListBox dengan Nilai Properti Columns 0, 1 dan 3

Kontrol *ListBox* juga memperbolehkan seorang pengguna untuk memilih lebih dari satu item pada suatu waktu. Untuk mengaktifkan, properti *MultiSelect* diberi nilai *1-Simple* atau *2-Extended*. Pada mode *1-Simple* pemilihan dapat dilakukan dengan menggunakan menekan spasi (*space bar*) atau *mouse*. Pada mode *2-Extended* pemilihan dilakukan dengan menekan tombol *shift*. Selain properti *Multiselect* , kontrol ListBox mempunyai properti *Style* yang diberi nilai *0-Standard* dan *1-Checkbox* seperti ditunjukkan Gambar 4.5



Gambar 4.5 Kontrol ListBox dengan properti Style 0–Standard dan 1-Checkbox

Contoh : menggunakan ListBox untuk memilih kota yang pernah Anda kunjungi :

Gambar 4.6 Contoh Program Menggunakan ListBox

- Atur form seperti pada gambar
- Ketikkan kode program seperti berikut :

```
Private Sub cmdAdd_Click()
'Menampilkan nama kota di lstKunjung sesuai yang dipilih di lstKota

Dim CurItem As Integer
CurItem = 0
   Do
      'Jika item yang dipilih
    If lstKota.Selected(CurItem) Then
        'Tambahkan ke lstKunjung.Jika Anda menambahkannya ke ComboBox,
        'ganti "lstKunjung" di bawah dengan nama ComboBox yang ada.
        'Contoh: cboKunjung.AddItem lstKota.List(CurItem)
        lstKunjung.AddItem lstKota.List(CurItem)
        'Lalu hapus dari lstKota
        lstKota.RemoveItem (CurItem)
     Else
       CurItem = CurItem + 1
    End If

   Loop Until CurItem = lstKota.ListCount

End Sub

Private Sub cmdAddAll_Click()
'Memindahkan semua kota ke lstKunjung
For i = 0 To lstKota.ListCount - 1
   lstKunjung.AddItem lstKota.List(i)
Next i
lstKota.Clear
End Sub
```

8. Combo Box

Kontrol *ComboBox* merupakan kontrol yang mirip dengan kontrol *Listbox*, jadi apa yang dapat bekerja pada kontrol *Listbox* juga bekerja dengan baik pada kontrol *comboBox*. Seperti pada kontrol *listbox* juga memiliki properti *sorted* yang digunakan untuk mengurutkan item data secara otomatis dan properti *list* yang digunakan untuk menambahkan data pada saat design time. Kebanyakan metode yang digunakan pada *ListBox* juga terdapat pada *ComboBox* seperti *AddItem, RemoveItem, dan Clear*.

Kontrol *ComboBox* sebenarnya merupakan gabungan antara sebuah *TextBox* dan *ListBox*. Kontrol *ComboBox* juga memiliki properti *Style* memberikan 3 pilihan yaitu :

- 0 – Dropdown Combo : pemakai diperbolehkan mengetik tulisan yang tidak ada di *list* (daftar)

- 1 – Simple Combo : pemakai boleh mengetik atau memilih dengan tombol ke atas dan ke bawah pada *keyboard* dan daftar tidak bias terbuka.
- 2 – Dropdown List : pemakai hanya dapat memilih tulisan yang ada di *list* (daftar) dan tidak bisa mengetik di *list*

9. Image

Digunakan untuk menampilkan gambar dalam format bitmaps (BMP), device independent bitmaps (DIB), metafiles (WMF), enhanced metafiles (EMF), GIF dan JPEG compressed files, dan icons (ICO dan CUR).

Properti *Stretch* digunakan untuk menentukan apakah gambar disesuaikan dengan ukuran kontrol (gambar dapat menggalami pengecilan maupun pembesaran)

➢ **Menampilkan gambar kedalam kontrol image**

Pada saat *design* anda dapat mengisikan gambar ke dalam *image box* dengan menggunakan properti *Picture*, sedangkan pada runtime anda dapat menggunakan fungsi *LoadPicture(namafile)* untuk memuat gambar ke properti Picture dari kontrol image, contoh :

```
imgGambar.Picture = LoadPicture("D:\Picture\stmik.jpg")
```

Catatan : Fungsi *LoadPicture(namafile)* digunakan untuk memuat file grafik dengan format grafik bitmap (.bmp), icon (.ico), run-length encoded (.rle), metafile (.wmf), enhanced metafiles (.emf), GIF, JPEG (.jpg).

➢ **Mengosongkan kontrol image**

Untuk mengosongkan kontrol image pada saat runtime, anda dapat menggunakan fungsi LoadPicture, tanpa menggunakan argumen nama file,

```
imgGambar.Picture = LoadPicture
```

➢ **Menyimpan gambar dalam kontrol *Image* ke File**

Anda dapat menggunakan perintah *SavePicture* gambar, namafile untuk menyimpan gambar kedalam file dengan format BMP, contoh :

```
SavePicture imgGambar.Picture,"D:\Picture\stmik.bmp"
```

Catatan : Perintah *SavePicture* akan selalu menyimpan gambar ke format bitmap (.bmp), tanpa memperhatikan format sumber gambar.

10. Picture Box

Kalau *Image* digunakan untuk menampilkan gambar, demikian juga *PictureBox* Selain menampilkan gambar, *Picture Box* mendukung berbagai metoda untuk

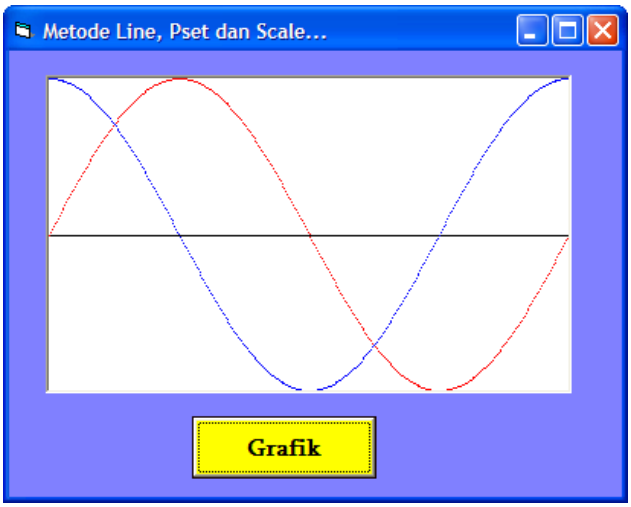
untuk operasi grafik, dan dapat berfungsi sebagai kontainer bagi kontrol-kontrol lain, seperti fungsi *Frame*.

Contoh penggunaan metode *Line, Pset, Scale* pada *PictureBox* untuk menampilkan grafik :

Ketikkan kode progam seperti berikut :

```
Private Sub cmdGrafik_Click()
picGrafik.ScaleMode = 0
picGrafik.ScaleWidth = 360
picGrafik.ScaleHeight = 2
picGrafik.ScaleLeft = 0
picGrafik.ScaleTop = -1
picGrafik.ForeColor = vbBlack ‘Set ForeColor Jadi Hitam
picGrafik.Line (0, 0)-(360, 0) ‘Garis Hitam
For i = 0 To 360
   picGrafik.ForeColor = vbRed   ‘Titik Merah
   picGrafik.PSet (i, -Sin(i * 3.14 / 180))

   picGrafik.ForeColor = vbBlue  ‘Titik Biru
   picGrafik.PSet (i, -Cos(i * 3.14 / 180))
Next i
End Sub
```

Setelah program dijalankan, maka hasilnya seperti terlihat pada gambar 4.7



Gambar 4.7 Membuat Grafik dengan PictureBox

- **Scale Mode**

  Digunakan untuk menentukan unit skala yang digunakan (0 - User, 1 - Twip, 2 - Point, 3 - Pixel, 4 - Character, 5 - Inch, 6 - Milimeter, 7 - Centimeter)

  - 1 inchi = 1440 Twip
  - 1 cm = 567 Twip
  - 1 inchi = 72 point
  - 1 character= (120 Twip untuk lebar, 240 untuk tinggi)
  - 1 cm = 1000 unit (Himetric)

  Anda dapat membuat modus skala sendiri dengan menset properti ini menjadi 0 - User, dan skala anda dapat ditentukan pada ScaleWidth dan ScaleHeight, perhatikan kembali contoh Grafik sinus sebelumnya.

- **Scale Left**

  Digunakan untuk menentukan nilai koodinat horizontal paling kiri, Anda dapat menggunakan properti ini untuk menentukan koordinat paling kiri dari suatu sumbu X.

- **Scale Top**

  Digunakan untuk menentukan nilai koodinat vertikal paling atas, anda dapat menggunakan properti ini untuk menentukan koordinat paling atas dari suatu sumbu Y.

- **Scale Height**

  Digunakan untuk menentukan tinggi sumbu vertikal. Menentukan panjang sumbu Y.

- **ScaleWidth**

  Digunakan untuk menentukan panjang sumbu horizontal. Menentukan panjang sumbu X.

Metode Grafik Pada PictureBox

Salah satu perbedaan antara Image dengan PictureBox adalah tersedianya berbagai metoda penggambaran grafik pada *PictureBox*, antara lain:

- **Circle(x,y),r,warna,awal,akhir,Aspek**

Menggambarkan sebuah lingkaran dengan berpusat pada koordinat **x,y** dan jari-jari **r** dengan **warna** garis, mulai dari sudut **awal**, sampai sudut **akhir** (yang dinyatakan dalam radian), serta **aspek** perbandingan tingi dengan lebar.

- **Cls**

  Membersihkan *PictureBox* dengan warna *BackColor*

- **Line (x1,y1) - (x2, y2),warna**

  Menggambarkan garis tunggal dari koordinat **x1,y1** sampai dengan **x2,y2**

- **Line (x1,y1) - (x2, y2),warna,B**

  Menggambarkan kotak dari koordinat **x1,y1** sampai dengan **x2,y2**

- **Line (x1,y1) - (x2, y2),warna,BF**

  Mengambarkan kotak berisi dari koordinat **x1,y1** sampai dengan **x2,y2**

- **Pset (x,y)**

  Mencetak dot (titik) pada koordinat tertentu pada *Form, Picturebox*, dan *Printer*, pada koordinat yang ditentukan

11. ScrollBar

Ada 2 macam kontrol *ScrollBar* yaitu *HscrollBar* dan *VscrollBar*. Keduanya mempunyai sifat yang sama namun berbeda orientasi. Properti dari kontrol tersebut yang cukup penting adalah properti *Min* dan *Max* yang merepresentasikan nilai jangkauan minimum dan maksimum. Properti lain yang penting pada waktu run-time adalah *Value*, yang selalu mengembalikan nilai posisi indikator yang terdapat pada *scrollbar*.

Ada 2 kejadian (Event) penting dari *scroolbar* yaitu kejadian *Change* yang akan dipanggil pada saat *scrollbar* diklik dan kejadian geser (*Scroll*) yang dipanggil ketika indikator pada *scrollbar* mengalami perubahan.

Contoh : Menggeser posisi *Image* ke atas dan ke bawah

Gambar 4.8 Menggunakan VscrollBar untuk menggeser posisi gambar

Kode programnya seperti berikut ini:

```
Private Sub vsbGambar_Change()
imgGambar.Top = -vsbGambar.Value
End Sub

Private Sub vsbGambar_Scroll()
imgGambar.Top = -vsbGambar.Value
End Sub
```

Event Scroll, akan dibangkitkan ketika pemakai melakukan pergeseran terhadap Bar dengan menggunakan drag pada tombol kiri mouse, jadi Event Scroll akan terjadi ketika pemakai melakukan pergeseran dengan menekan tombol kiri mouse, dan diakhiri dengan Event Cahnge ketika pemakai melepas penekanan mouse. Jadi anda harus memanfaatkan kedua event tersebut untuk mendapatkan hasil yang baik dalam pemakaian ScrollBar.

12. Timer

Komponen Timer sangat baik untuk mengimplementasikan pengaruh waktu terhadap suatu proses seperti proses animasi atau dalam pembuatan game supaya kecepatan dari game bisa diatur. Komponen Timer bersifat non-visual, pada saat program dijalankan, Timer tidak kelihatan. Beberapa properti *Timer* adalah sebagai berikut :

- **Enabled**

  Menentukan apakah kontrol dapat efektif terhadap Event Timer.

- **Index**

  Digunakan untuk menentukan nomor index, jika kontrol tersebut merupakan kontrol array.

- **Interval**

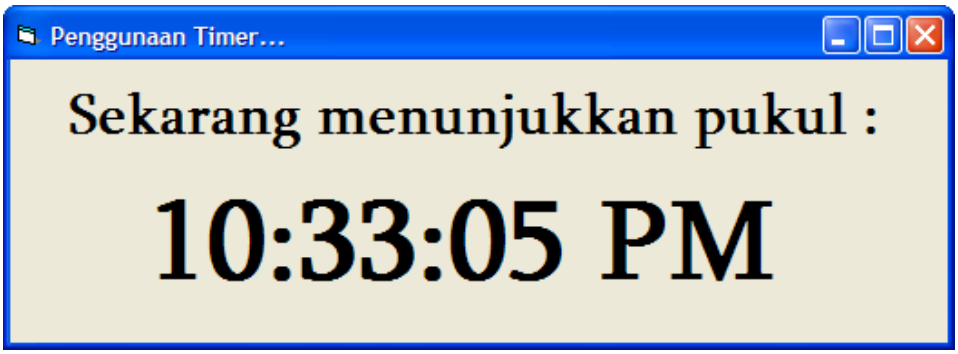
  Menentukan nilai interval dalam mili detik (1/1000) antar pemanggilan Event Timer.

- **Tag**

  Properti ini dapat digunakan sebagai tempat menyimpan data sementara yang berkaitan dengan kontrol label tersebut

  Event Timer : Event yang dibangkitkan oleh kontrol timer berdasarkan interval waktu yang telah ditentukan.

Contoh : Membuat Jam Digital

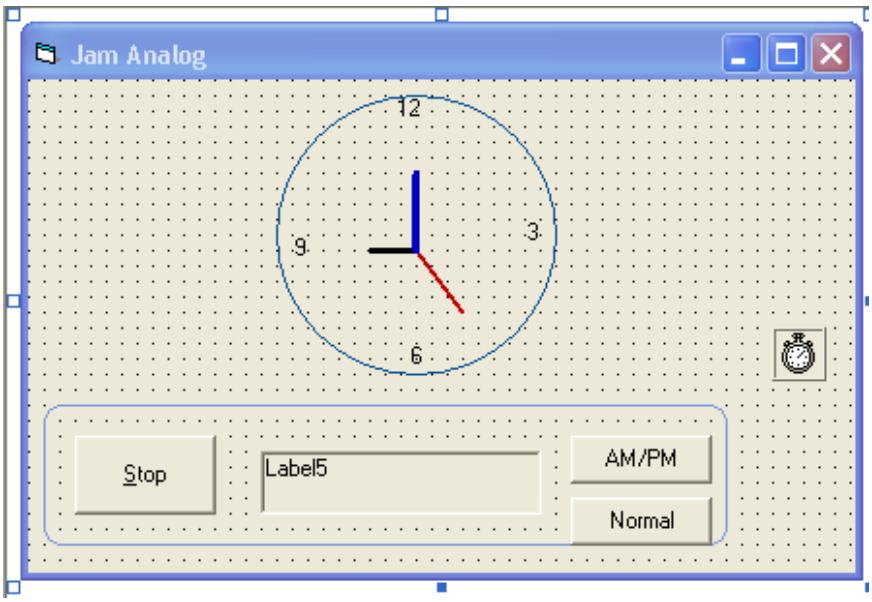


Gambar 4.9 Menggunakan Timer untuk membuat Jam Digital

Letakkan Label dan Timer pada Form seperti pada gambar 4.9. Atur properti Timer, (*Enabled : True*, *Interval* : 1000). Dan ketikkan kode

```
Private Sub tmrJam_Timer()
lblJam.Caption = Format(Now, "hh:mm:ss")
End Sub
```

Contoh : Membuat jam analog

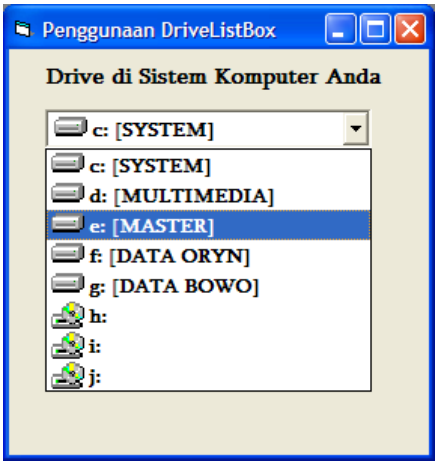


Form/Control	Properties	Setting
Form1	Name Caption	Form1 Jam Analog
Shape 1	Name Shape	Shape1 3-Cirlcle
Shape2	Name Shape	Shape2 4-Rounded Rectangle
Label1	Name BackStyle Caption	Label1 0-Transparent 12
Label2	Name BackStyle Caption	Label2 0-Transparent 3
Label3	Name BackStyle Caption	Label3 0-Transparent 6
Label4	Name BackStyle Caption	Label4 0-Transparent 9
Label5	Name BackStyle Caption	Label5 1-FixedSingle
Command1	Name Caption Index	Cmd_bentuk AM/PM Normal STOP 0,1,2
Timer1	Name Interval	Tmrquartz 1000

```
Const pi = 3.14159
Dim sw As Integer

Private Sub Form_Load()
Call Timer1_Timer
sw = 0
End Sub

Private Sub Cmd_bentuk_Click(Index As Integer)
Select Case Index
Case 0
sw = 0
Label5.Caption = Format(Time, "hh:mm:ss AMPM")
Case 1
sw = 1
Label5.Caption = Format(Time, "hh:mm:ss")
Case 2
End
End Select
End Sub

Private Sub TmrQuartz_Timer()
Dim xh As Single, yh As Single
Dim xm As Single, ym As Single
Dim xs As Single, ys As Single
Dim hours As Single
Dim minutes As Single
Dim seconds As Single
Dim jam As Single
If sw = 0 Then
Label5.Caption = Format(Time, "hh:mm:ss AMPM")
Else
Label5.Caption = Format(Time, "hh:mm:ss")
End If
Beep
hours = Hour(Time)
minutes = Minute(Time)
seconds = Second(Time)
jam = hours + minutes / 60
xh = 500 * Cos(pi / 180 * (30 * jam - 90))
yh = 500 * Sin(pi / 180 * (30 * jam - 90))
linehour.X2 = xh + linehour.X1
linehour.Y2 = yh + linehour.Y1
xm = 750 * Cos(pi / 180 * (6 * minutes - 90))
ym = 750 * Sin(pi / 180 * (6 * minutes - 90))
Lineminute.X2 = xm + linehour.X1
Lineminute.Y2 = ym + linehour.Y1
xs = 770 * Cos(pi / 180 * (6 * seconds - 90))
ys = 770 * Sin(pi / 180 * (6 * seconds - 90))
Linesecond.X2 = xs + linehour.X1
Linesecond.Y2 = ys + linehour.Y1

End Sub
```

13. DriveListBox, DirListBox, FileListBox

- *DriveListBox* digunakan untuk menampilkan drive yang terdapat pada sistem komputer

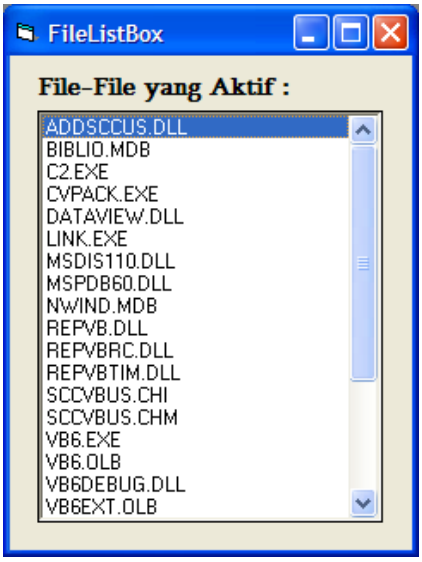


Gambar 4.10 Menggunakan DriveListBox

- DirListBox dapat digunakan untuk menampilkan folder-folder yang terdapat di drive yang aktif.

Gambar 4.11 Menggunakan DirListBox

- *FileListBox* digunakan untuk menampilkan file-file yang terdapat pada folder yang aktif.



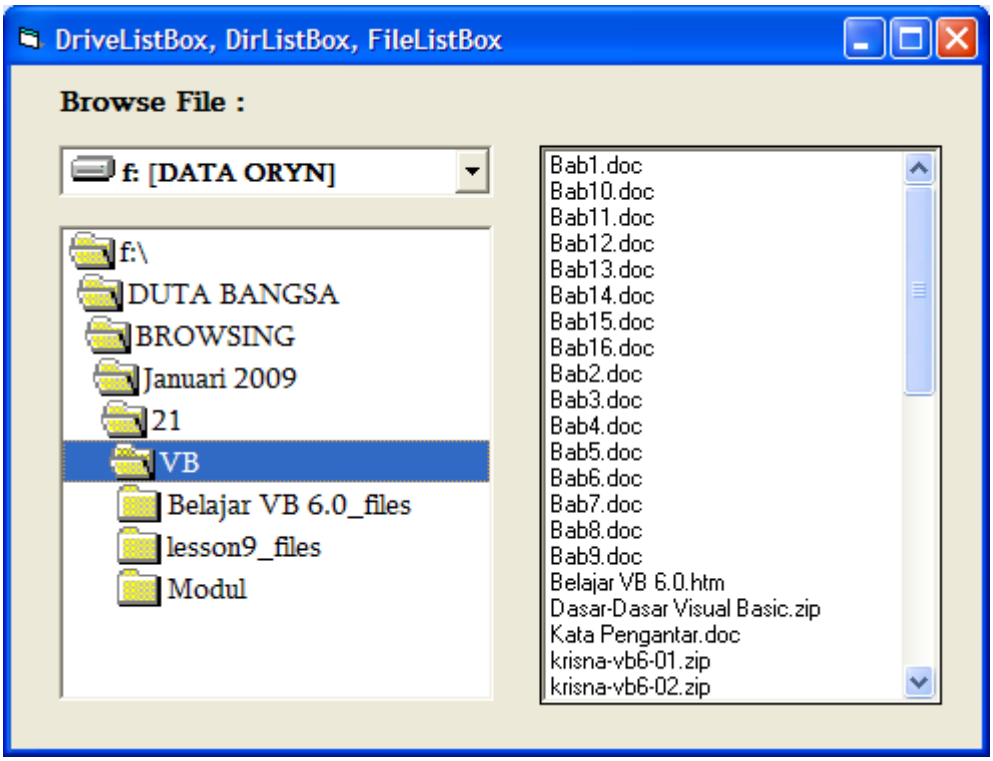
Gambar 4.12 Menggunakan FileListBox

- **Membatasi file pada FileListBox**

  Untuk membatasi file berdasarkan nama file, anda dapat menggunakan properti *Pattern*, misalnya kalau file yang ingin ditampilkan hanya berupa file bitmap, maka kita dapat menggunakan (*.bmp), kalau file yang ingin ditampilkan berupa file grafik kita dapat menggunakan (*.bmp;*.jpg;*.gif;*.wmf;*.ico) Untuk membatasi file berdasarkan atributnya, anda dapat menggunakan properti Archive, Hidden, Normal, ReadOnly dan System, dengan menentukan masing-masing menjadi True atau False.

- **Menghubungkan DriveListBox, DirListBox dan FileListBox**

  Misalnya kita memiliki tiga buah kontrol yaitu DriveListBox, DirListBox, dan FileListBox dimana perubahan pada DriveListBox akan menyebabkan perubahan tampilan pada DirListBox, dan FileListBox.



Gambar 4.13 Menghubungkan DriveListBox, DirListBox, FileListBox

Ketikkan kode program seperti berikut :

```
Private Sub dirTest_Change()
filTest.Path = dirTest.Path
End Sub

Private Sub drvTest_Change()
dirTest.Path = drvTest.Drive
End Sub
```

14. Common Dialog Box

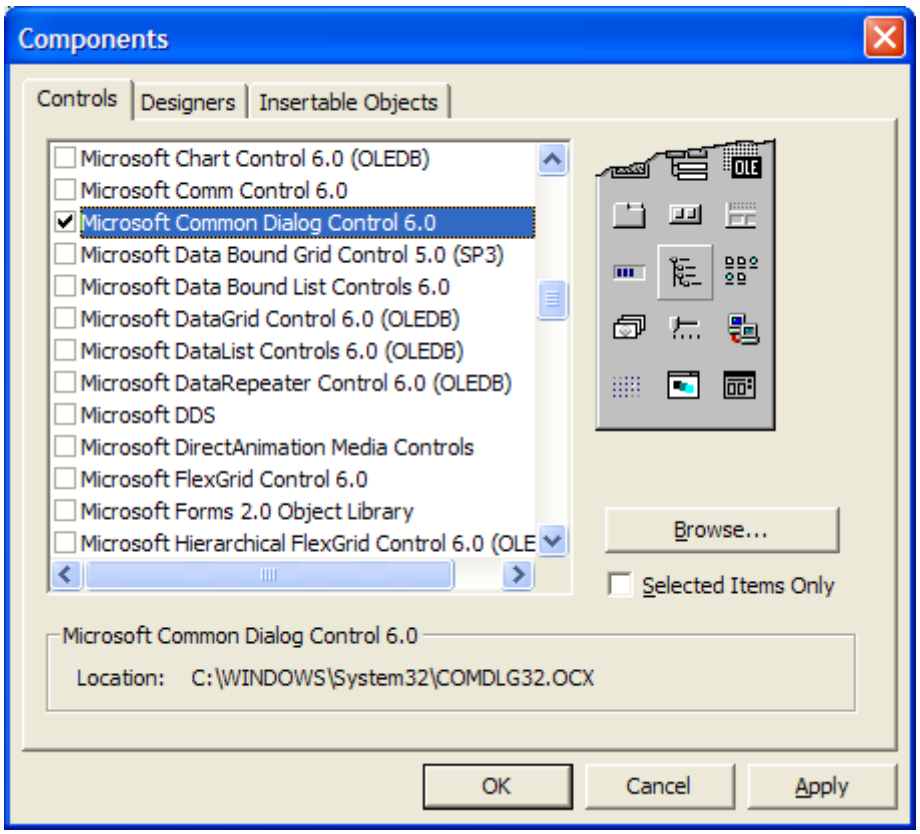
a. Metode Common Dialog

Common Dialog Box merupakan kontrol yang menyediakan fasilitas dialog-dialog umum yang sering digunakan pada lingkungan Windows seperti *File Dialog*, *Font Dialog*, dan *Printer Dialog*.

Pada saat kontrol tersebut diletakkan pada form, kontrol tersebut tidak dapat diubah ukurannya seperti kontrol timer, namun kontrol tersebut dapat dilihat pada saat program dijalankan (*runtime*) dalam bentuk dialog.

Meskipun kontrol tersebut merupakan kontrol yang umum, namun kontrol ini tidak diletakkan pada ToolBox. Untuk menambahkan kontrol tersebut pada Toolbox ikutilah langkah-langkah berikut ini:

1. Pilih menu *Project-Components* atau dengan menekan *Ctrl-T*, maka akan ditampilkan *Components Dialog box*, lihat gambar 4.14.



Gambar 4.14 Window Components-Common Dialog

2. Pilih Microsoft Common Dialog Box Control pada ListBox dan klik OK, maka Kontrol Dialog Box akan ditampilkan pada ToolBox.
3. Untuk menambahkan pada form, double click pada kontrol tersebut

Untuk menampilkan Dialog box dapat digunakan salah satu Method berikut ini:

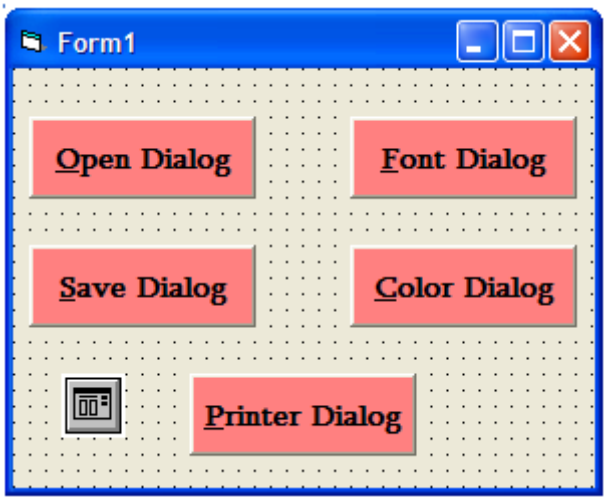
- *ShowColor* untuk menampilkan sebuah color dialog.
- *ShowFont* untuk menampilkan sebuah font dialog.
- *ShowHelp* untuk menampilkan sebuah help dialog.
- *ShowOpen* untuk menampilkan sebuah open file dialog.
- *ShowPrinter* untuk menampilkan sebuah printer dialog.

- *ShowSave* untuk menampilkan sebuah save file dialog

```
cdbDialog.DialogTitle = "File Open"
cdbDialog.Filter = "*.txt" 'Tampilkan file text saja
cdbDialog.FileName = "*.txt"    'Default nama file
cdbDialog.ShowOpen ' menampilkan dialog open file
```

b. Contoh

Tambahkan 1 Common Dialog Control dan 5 Command di Form Anda. Atur seperti berikut :



Gambar 4.15 Contoh Program menggunakan Common Dialog

Ketikkan kode program berikut :

```
Private Sub cmdColor_Click()
   cdbDialog.DialogTitle = "Select a Color"
   cdbDialog.ShowColor     ' Display the dialog box
End Sub

Private Sub cmdFont_Click()
   cdbDialog.DialogTitle = "Font"
   cdbDialog.FontName = "Arial"
   cdbDialog.FontBold = cdlCFBoth
   cdbDialog.ShowFont
End Sub
```

```
Private Sub cmdPrinter_Click()
   cdbDialog.DialogTitle = "Select a Printer"
   cdbDialog.ShowPrinter
End

Private Sub cmdOpen_Click()
   cdbDialog.DialogTitle="File Open"
   cdbDialog.Filter="*.txt"   ' Tampilkan file text saja
   cdbDialog.FileName="*.txt"    ' Default nama file
   cdbDialog.ShowOpen ' menampilkan dialog open file
End Sub

Private Sub cmdSave_Click()
   cdbDialog.DialogTitle="File Save"
   cdbDialog.Filter="*.*"     ' Show all files
   cdbDialog.FileName="test.txt"  ' Default filename
   cdbDialog.ShowSave   ' Trigger the dialog box
End Sub
```

15. Kontrol Animasi

Kontrol Animasi dapat digunakan untuk menjalankan file AVI dan untuk menambahkan animasi pada sebuah program aplikasi. Kontrol ini hanya mendukung file AVI saja yang tidak memiliki suara dan tidak terkompres.

File AVI ini dapat digemukan pada subdirektori \Common\Graphics\Video pada Microsoft Visual Studio 6.0

Kontrol animasi menyediakan 3 properti utama. Dua diantaranya adalah *Center* dan *BackStyle* yang hanya dapat diset pada saat design time dan bersifat *read only* pada saat run time. Jika properti Center diset true, maka file AVI akan ditampilkan tepat ditengah.

Sedangkan properti BackStyle dapat diberi nilai 0-cc2 (*BackStyle Transparent*) atau 1-cc2 (*BackStyle Opaque*). Properti yang ketiga adalah *AutoPlay* yang dapat diberi nilai kapanpun juga. Jika properti ini diberi nilai *True* maka secara otomatis akan dimainkan segera pada saat kontrol tersebut ditampilkan.

Untuk membuka file AVI yang akan digunakan untuk ditampilkan dapat digunakan methods *Open*.

```
Animation1.Open "C:\vb6\Graphics\AVIs\filecopy.avi"
```

Untuk menjalankan file avi dapat digunakan methods Play dengan format sebagai berikut:

```
Play [RepeatCount], [StartFrame], [EndFrame]
```

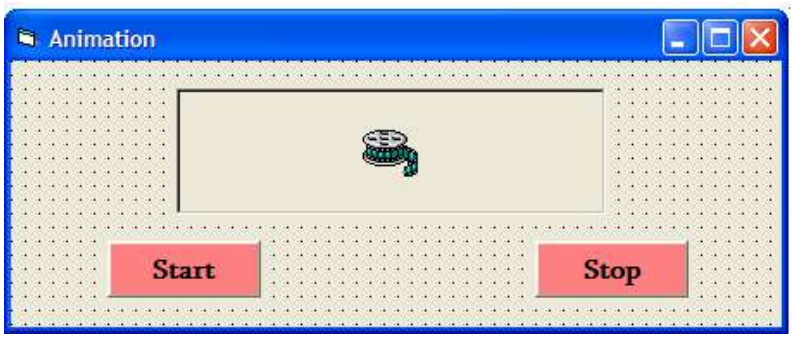
- *RepeatCount* merupakan banyaknya perulangan animasi yang dijalankan (nilai defaultnya adalah 1 yang akan menjalankan file AVI terus menerus).
- *StartFrame* menunjukkan awal frame yang dijalankan.
- *EndFrame* menunujukkan akhir frame.

Untuk menghentikan animasi dapat digunakan salah satu dari 2 cara tergantung pada cara memulai yaitu :

a. Jika animasi dalam mode *AutoPlay*, maka dapat dihentikan dengan memberi nilai properti *AutoPlay* menjadi false.

b. Jika animasi dijalankan dengan menggunakan method *play* maka dapat dihentikan dengan menggunkan method *stop*.

Untuk menghemat memori, kontrol animasi dapat di-*unload* dengan menggunakan metoda *close*



Gambar 4.16 Contoh Program menggunakan Animation 1

16. Contoh Program

Berikut contoh program untuk menghitung durasi waktu pada saat user klik tombol START sampai user klik tombol STOP. Ketika program dijalankan, waktu akan berjalan. Ketika user klik tombol START maka durasi waktu akan berjalan, ketika user klik tombol STOP maka waktu berhenti dan user mendapat informasi durasi waktu. Dan ketika klik tombol EXIT maka program selesai.

Buatlah tampilan form seperti gambar 4.17. Tambahkan sebuah Timer. Ketik kode program seperti yang telah disediakan di kotak *source code*.

Gambar 4.17 Contoh program menghitung durasi waktu

Ketikkan kode program seperti berikut :

```
Dim Awal As Date
Dim Akhir As Date
Dim Lama As Single

Private Sub cmdProses_Click()
'Jika tombol mula-mula adalah Start
If cmdProses.Caption = "START" Then
     Awal = Now
     'Tampung waktu pertama kali dimulai
     lblStart.Caption = Format(Awal, "hh:mm:ss")
     cmdProses.Caption = "STOP"
  'Jika tombol dengan tulisan Stop ditekan
ElseIf cmdProses.Caption = "STOP" Then
    'Matikan Timer
    tmrWaktu.Enabled = False
    Akhir = Now
    Lama = Akhir - Awal
    lblDurasi.Caption = Format(Lama, "hh:mm:ss")
    cmdProses.Caption = "EXIT"
ElseIf cmdProses.Caption = "EXIT" Then
     Unload Me
End If
End Sub
Private Sub tmrWaktu_Timer()
lblJam.Caption = "Sekarang Jam : " & Format(Now, "hh:mm:ss")
If cmdProses.Caption = "STOP" Then
   'Hitung kembali durasi waktu
    lblDurasi.Caption = Format(Now - Awal, "hh:mm:ss")
End If
End Sub
```

C. Latihan

Buka kembali contoh program seperti pada gambar 4.17 di atas. Tambahkan satu textbox/label untuk menginformasikan berapa jumlah yang harus dibayarkan setiap durasi waktu tertentu. Misalnya setiap durasi 30 detik harus membayar Rp. 25,00. Jadi biaya akan otomatis bertambah terus menerus selama program belum diSTOP.

BAB V

FUNGSI-FUNGSI

A. Pendahuluan

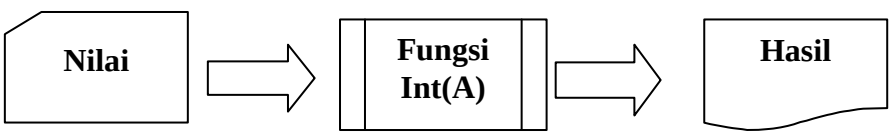
Fungsi adalah suatu program yang dapat menerima berbagai nilai dan memberikan umpan balik tertentu. Nilai yang dimasukkan di dalam suatu fungsi disebut dengan parameter, yang berupa berbagainjenis seperti angka, string, tanggal dan sebagainya.

Manfaat pemakaian fungsi dalam suatu program adalah bisa menghemat waktu dan tenaga. Melihat manfaatnya maka suatu fungsi biasanya digunakan untuk melakukan beberapa seperti berikut ini :

- Memanipulasi teks atau string, yaitu melakukan kegiatan yang berkaitan dengan operasi string. Misalnya fungsi untuk menghitung panjang suatu string, mengambil sebagian string, mengubah string menjadi huruf besar (kapital) dan sebagainya.
- Memanipulasi tanggal, jam, data yaitu melakukan kegiatan yang berkaitan dengan operasi tangga, jam, data dan bahkan juga melakukan operasi input dan output. Misalnya fungsi untuk menampilkan tahun dari suatu tanggal, menit dari suatu jam dan sebagainya.
- Perhitungan matematik, yaitu melakukan proses perhitungan matematika yang berkaitan dengan data nilai/angka. Misalnya menghitung nilai absolut, sinus dan sebagainya.

Cara kerja suatu fungsi dapat diibaratkan sebagai suatu rumus yang sudah disiapkan oleh visual basic. Pemakai tinggal memasukkan nilai yang akan dihitung, lalu hasilnya akan segera didpat.

Gambaran dari cara kerja fungsi dapat dilihat pada gambar berikut :



Gambar 5.1 Alur kerja suatu fungsi

Dari gambaran di atas terlihat bahwa nilai "2.9" diberikan ke variable "A" yang selanjutnya variable tersebut dikenakan suatu fungsi **int().** Hasil dari pengolahan fungsi tersebut adalah dihasilkan suatu angka "2".

Pada visual basic 6.0 terdapat beberapa fungsiintrinsik atau fungsi-fungsi yang sudah built-in, sehingga pemakai tidak perlu lagi untuk membuatnya. Bentuk penulisan dari fungsi adalah nama fungsi ditulis dengan tanda kurung yang dapat berisi parameter dari fungsi tersebut. Perhatikan contoh berikut ini :

Int(3.14)

Left("Komputer",3)

Beberapa fungsi pada visual basic telah dilengkapi dengan bantuan yang berupa tooltip, yang menunjukkan sintaks yang tepat dari fungsi tersebut.

B. Materi

1. Remarks

Remarks digunakan untuk membantu memberikan keterangan seorang programer lain yang nanti akan memodifikasi program aplikasi dikemudian hari. *Remarks* memberikan pesan – pesan yang penting berkaitan dengan program yang dibuat. *Remarks* tidak harus ditulis dengan format tertentu atau bahasa Inggris, melainkan dapat ditulis dalam bahasa Indonesia sekalipun.

Jadi *remarks* merupakan pesan atau keterangan yang ditulis dalam kode program. *Remarks* digunakan untuk membantu menjelaskan tentang kode yang ditulis dan Visual Basic akan mengabaikan semua *Remarks* yang ada pada kode program. Tujuan menambahkan *Remark*s dalam program :

- Memberikan keterangan nama programmer dan tanggal pembuatan program.
- Memberikan keterangan umum mengenai prosedur dan fungsi yang digunakan
- Memberikan keterangan pada perintah – perintah yang sulit dimengerti sehingga jika ada programer lain yang akan memodifikasi akan mengerti maksud dari kode yang ditulis.

Visual Basic mendukung 2 macam penggunaan *Remarks* yaitu *Remarks* yang dimulai dengan menggunakan kata *Rem* dan *Remarks* yang dimulai dengan tanda petik (‘).

Berikut ini format penggunaan statement *Rem*.

```
Rem Programmer: STMIK-DB , Tanggal : 24-Feb-2009
Rem Program untuk menghitung luas segitiga.
Rem Dengan menggunakan masukan berupa
Rem 1. Tinggi segitiga (t)
Rem 2. Alas segitiga (a).
Rem Tombol hitung untuk menghitung luas segitiga
Rem Tombol exit untuk keluar dari aplikasi.
```

Pada contoh tersebut terdapat sejumlah *Remarks* yang menjelaskan nama *programmer* yang membuat dan tanggal pembuatannya serta kegunaan dari program tersebut. Selain menggunakan *Rem* juga dapat digunakan tanda petik satu untuk melakukan remark.

```
‘ Programmer: STMIK-DB , Tanggal : 24-Feb-2009
‘ Program untuk menghitung luas segitiga.
‘ Dengan menggunakan masukan berupa
‘ 1. Tinggi segitiga (t)
‘ 2. Alas segitiga (a).
‘ Tombol hitung untuk menghitung luas segitiga
‘ Tombol exit untuk keluar dari aplikasi.
```

Kedua contoh di atas memberikan pesan dan keterangan yang sama, hanya saja cara yang kedua lebih mudah.

2. Message Box

Adakalanya sebuah program ingin menampilkan pesan kesalahan atau bertanya pada user, sebab kontrol – kontrol yang ada pada *form* kurang jelas. *Message box* tidak seperti kontrol yang melekat pada form. *Message Box* berisi sebuah message akan ditampilkan tepat diatas sebuah form dan akan hilang ketika mendapat respon dari user dengan menklik tombol yang ada pada *message box* tersebut (lihat gambar 5.1).

Visual Basic menyediakan 2 cara untuk menampilkan *message box* yaitu dengan menggunakan statement *MsgBox* dan menggunakan fungsi *MsgBox.*

Gambar 5.2 Contoh pesan dengan MessageBox

a. Statement MsgBox

Pesan yang ditampilkan melalui statement *Msgbox* menampilkan tombol Ok. Ketika user selesai membaca pesan tersebut, maka user akan menekan tombol Ok untuk menutup pesan tersebut. Berikut ini Format statement MsgBox :

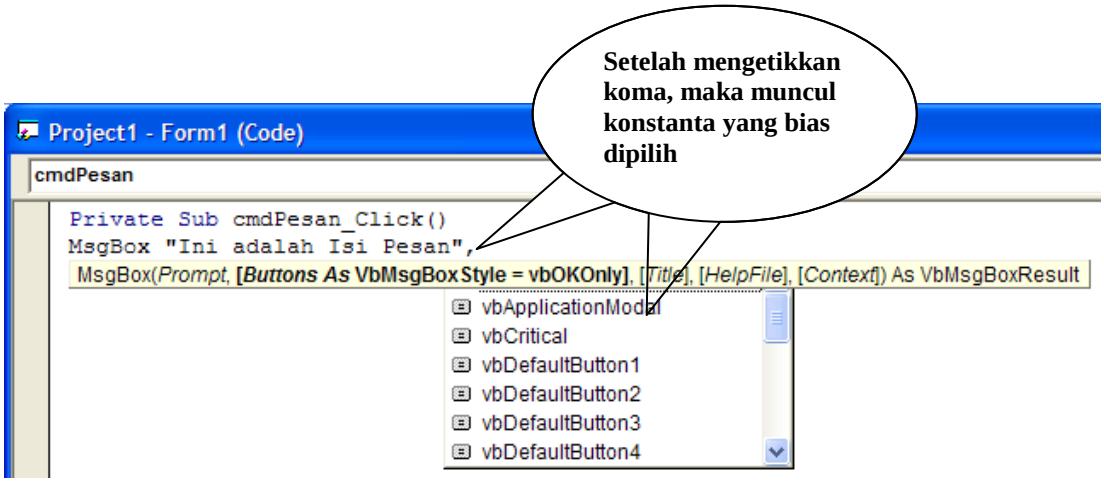
```
MsgBox Prompt, Style Value, Title
```

Prompt merupakan kalimat atau variabel yang berisi string yang akan ditampilkan. *Style* menentukan tipe dari command button yang akan terlihat pada message box dan nilainya dapat dipilih salah satu seperti pada tabel 5.1. Sedangkan *Title* menunujukkan judul dari message box.

Tabel 5.1 Nilai Konstanta Message Box

STYLE VALUE	KONSTANTA	TOMBOL YG DITAMPILKAN
0	vbOkOnly	Ok button
1	vbOkCancel	Ok dan Cancel buttons
2	vbAbortRetryIgnore	Abort, Retry dan Ignore buttons.
3	vbYesNoCancel	Yes, No dan Cancel buttons
4	vbYesNo	Yes dan No buttons
5	vbRetryCancel	Retry dan Cancel buttons

Kita bisa menggunakan nama konstanta atau syle value untuk menggantikan nilai integer pada argumen kedua. Penggunaan nama konstanta akan lebih mudah dibaca dibandingkan menggunakan style value. Visual Basic akan menampilkan list dari nama konstanta tersebut begitu anda mengetikan tanda koma setelah argumen pertama.



Gambar 5.3 Memilih Konstanta pad Message Box

b. Fungsi MsgBox

Untuk menampung tombol mana yang ditekan oleh user, maka digunakan fungsi MsgBox. Format pada fungsi MsgBox() sedikit berbeda dengan statement MsgBox. Fungsi ini menyediakan type yang lebih luas dibanding Statement MsgBox.

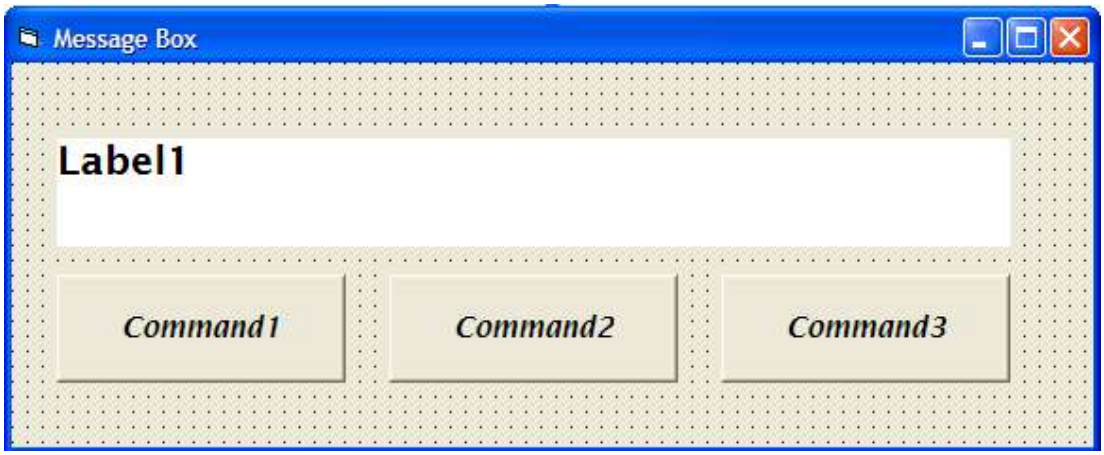
```
PesanAnda=MsgBox(Prompt, Style Value, Title
```

PesanAnda adalah variable yang akan menampung nilai dari fungsi MsgBox(). Nilainya menentukan type dari tombol yang diklik oleh user. Variable tersebut haruslah dideklarasikan sebagai tipe data integer pada general declaration section.

Tabel 5.2 Nilai Konstanta Saat Tombol MessageBox Diklik

NILAI	KONSTANTA	TOMBOL YG DIKLIK
1	vbOk	Ok button
2	vbCancel	Cancel button
3	vbAbort	Abort button
4	vbRetry	Retry button
5	vbIgnore	Ignore button
6	vbYes	Yes button
7	vbNo	No button

Contoh : Buat project baru dengan tiga command button dan label.



Gambar 5.4 Contoh program menggunakan MessageBox

Kemudian klik view code dan ketikan kode berikut ini :

```
Private Sub Command1_Click()
 Dim testMsg As Integer
   testMsg = MsgBox("Silahkan Anda Klik Tombol", 1, "Test")
   If testMsg = 1 Then
      Label1.Caption = "Anda meng-klik tombol OK"
   Else
      Label1.Caption = "Anda meng-klik tombol Cancel"
   End If
End Sub
```

Kode program di atas menggunakan Style Value untuk menampilkan tombol pada message. Kode program berikut menggunakan Konstanta untuk menmpilkan tombol pada message. Coba Anda ketikkan di Command2_Click dan lihat hasilnya!

```
Private Sub Command2_Click()
 Dim testMsg As Integer
   testMsg = MsgBox("Silahkan Anda Klik Tombol",vbOKCancel, "Test")
   If testMsg = vbOK Then
      Label1.Caption = "Anda meng-klik tombol OK"
   Else
      Label1.Caption = "Anda meng-klik tombol Cancel"
   End If
End Sub
```

Gambar 5.5 Tampilan pesan menggunakan MessageBox

Ketika user meng-klik OK pada test button, maka akan muncul pesan "Anda meng-klik tombol OK" sedangkan jika user meng-klik Cancel button maka akan muncul pesan "Anda meng-klik tombol Cancel"

Untuk membuat message box anda lebih menarik, anda bisa menambahkan icon pada message box tersebut. Ada empat tipe icon yang disediakan oleh Visual Basic seperti yang tertera pada tabel berikut ini :

Tabel 5.3 Nilai Konstanta dan Icon pada MessageBox

NILAI	KONSTANTA	ICON
16	vbCritical	
32	vbQuestion	
48	vbExclamation	
64	vbInformation	

Masukan kode berikut ini pada project yang telah anda buat sebelumnya.

```
Private Sub Command3_Click()
Dim testMsg As Integer
   testMsg = MsgBox("Silahkan Anda Klik", vbYesNoCancel _
                    + vbExclamation, "Test")
   If testMsg = 6 Then
      Label1.Caption = "Testing Successful"
   ElseIf testMsg = 7 Then
      Label1.Caption = "Are You Sure?"
   Else
      Label1.Caption = "Testing Fail"
   End If
End Sub
```

3. Input Box

InputBox akan menampilkan message dimana user dapat menginputkan suatu nilai atau message pada form tersebut. Format penggunaannya adalah sebagai berikut:

```
MyMessage=InputBox(Prompt, Title, default_text, x-position, y-position)
```

MyMessage adalah tipe data variant yang dideklarasikan sebagai string. Argumen yang tersedia dapat dijelaskan sebagai berikut:

- Prompt : Pesan yang ditampilkan.
- Title : Judul dari Input Box.
- default-text : Default text yang ditampilkan pada field dimana user dapat menggunakannya atau menggantinya.
- x-position and y-position : menunjukkan posisi dimana input box tersebut akan ditampilkan pada form

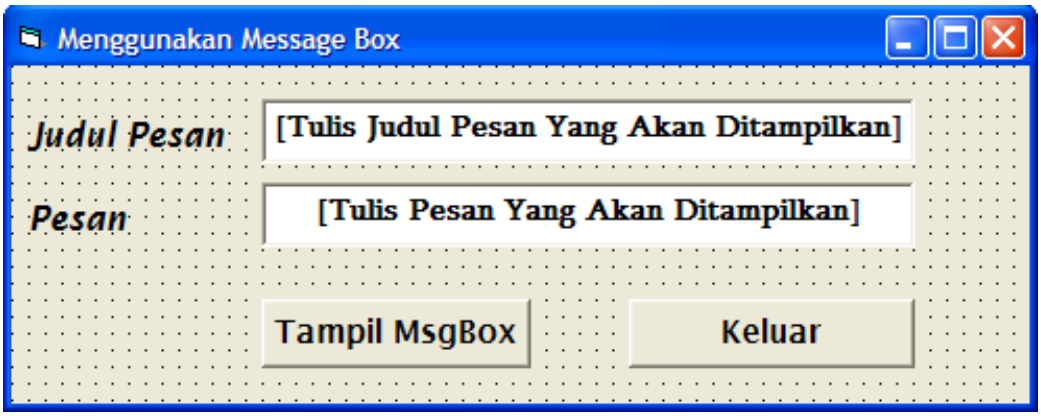
Tambahkan satu Command lagi pada project yang telah anda buat sebelumnya. Masukan kode berikut ini :

```
Private Sub Command4_Click()
Dim userMsg As String
   userMsg = InputBox("Apa pesan anda?", "Message Entry Form", _
   "Masukan message anda disini", 500, 700)
   If userMsg <> "" Then
      Label1.Caption = userMsg
   Else
      Label1.Caption = "No Message"
   End If
End Sub
```

Ketika user meng-klik tombol OK pada Input Box, pada label akan menampilkan message yang telah anda ketik, akan tetapi jika anda meng-klik tombol Cancel maka akan ditampilkan pesan "No Message".

4. Contoh Program

Contoh berikut ini adalah untuk menerapkan penggunaan MessageBox. Tampilan yang akan dibuat sperti pada gambar 5.4.



Gambar 5.6 Contoh MessageBox

Langkah-langkah :

- Jalankan IDE Visual Basic dan pilihlah project Standart Exe.
- Simpanlah project tersebut dengan nama cthPesan.frm untuk form dan cthPesan.vbs untuk projectnya.
- Pada form letakkan kontrol seperti pada gambar 5.5.
- Edit properti dari kontrol-kontrol seperti tabel 5.4.

Tabel 5.4 Nilai Properti Contoh MessageBox

Control	Properti	Setting Value
Form	Name	frmMsgBox
	Caption	Menggunakan Message Box
Command Button	Name	cmdMsgBox
	Caption	Tampil MsgBox
Command Button	Name	cmdKeluar
	Caption	Keluar
Label	Name	lblJudulPesan
	Caption	Judul Pesan
	Font	Size = 12
Label	Name	lblPesan
	Caption	Pesan
	Font	Size = 12
Text	Name	txtJudulPesan
	Font	Size = 12
	Text	[Tulis Judul Pesan Yang Akan Ditampilkan]
Text	Name	txtPesan
	Font	Size = 12
	Text	[Tulis Pesan Yang Akan Ditampilkan]

- Kemudian ketik kode berikut ini :

```
Rem *--------------------------------------*
Rem * Nama Program   = Contoh Message Box   *
Rem * Programmer     = Oryn.Com             *
Rem * Tgl Pembuatan  = 24 Februari 2009     *
Rem * STMIK DUTA BANGSA                    *
Rem *--------------------------------------*

Private Sub cmdKeluar_Click()
  End
End Sub

Private Sub cmdMsgBox_Click()
  Dim JudulPesan As String
  Dim Pesan As String
  JudulPesan = txtJudulPesan.Text
  Pesan = txtPesan.Text
  MsgBox Pesan, vbOKOnly, JudulPesan
End Sub
```

5. Fungsi-Fungsi Numerik

Dengan menggunakan fungsi-fungsi bawaan yang disediakan, waktu pembuatan sebuah program dapat disingkat. Fungsi-fungsi numerik merupakan fungsi-fungsi yang berkaitan dengan tipe data numerik. Fungsi-fungsi ini dapat dikelompokkan sebagai berikut :

a. Fungsi-Fungsi Konversi ke Integer

Ada tiga macam fungsi yang melakukan konversi ke tipe data integer yaitu seperti terlihat pada tabel 5.5.

Tabel 5.5 Fungsi Konversi ke Integer

Fungsi	Keterangan
CInt()	Pembulatan nilai desimal 0.5 dan lebih ke integer yang lebih dekat
Fix()	Pemotongan suatu nilai desimal menjadi integer
Int()	Pembulatan ke bawah suatu nilai desimal

Untuk bilangan positif, fungsi Fix() dan Int() memberikan perlakuan yang sama. Sebagai contoh, kedua pernyataan berikut ini mengembalikan nilai

```
ans1 = Int(14.5)       'mengembalikan 14
ans2 = Fix(14.5)       'mengembalikan 14
```

Namun fungsi Fix() dan Int() memberikan perlakuan berbeda pada bilangan negatif. Kedua pernyataan berikut memberikan pengembalian yang berbeda :

```
ans1 = Int(-14.5)      'mengembalikan -15
ans2 = Fix(-14.5)      'mengembalikan -14
```

Sedangkan fungsi CInt() mengembalikan pembulatan angka yang terdekat. Berikut ni contoh penggunaan CInt() :

```
ans1 = CInt(14.1)      'mengembalikan 14
ans2 = CInt(14.5)      'mengembalikan 14
ans3 = CInt(14.6)      'mengembalikan 15
ans2 = CInt(-14.5)     'mengembalikan -14
ans3 = CInt(-14.6)     'mengembalikan -15
```

b. Fungsi – Fungsi Konversi Tipe Data

Tabel 5.6 merupakan fungsi – fungsi yang melakukan konversi ke tipe data.

Tabel 5.6 Fungsi-Fungsi Konversi ke Tipe Data

Fungsi	Keterangan
CCur()	Mengkonversi argumen menjadi tipe data Currency
CDbl()	Mengkonversi argumen menjadi tipe data double (presisi ganda)
CLng()	Mengkonversi argumen menjadi tipe data Long Integer
CSng()	Mengkonversi argumen menjadi tipe data Single
CStr()	Mengkonversi argumen menjadi tipe data String
CVar()	Mengkonversi argumen menjadi tipe data Variant

Sebagai contoh misalkan sebuah data yang merupakan hasil bagi (1 / 7).

```
lblNilai = CSng(1/7) 'ditampilkan 0.1428571
lblNilai = CDbl(1/7) 'ditampilkan 0.142857142857143
```

Contoh yang lain, ketika Anda akan menghitung 2/0.5 :

```
'Contoh 1:
MsgBox 2 / CInt(0.5) '<-- error Division by zero

'Contoh 2:
MsgBox 2 / Val(0.5) '<-- error Division by zero

'Contoh 3:
MsgBox 2 / CDbl(0.5)  '<-- benar menghasilkan 4
```

Mengapa contoh 1 dan 2 menghasilkan error **Divison by zero**? Karena bilangan pecahan di kedua contoh tersebut dikonversi ke bilangan bulat,

sehingga nilai 0.5 (nol koma lima) dianggap sebagai angka nol saja. Semua angka yang dibagi dengan nol maka akan menghasilkan error . Jadi, solusinya gunakan cara pada contoh 3.

6. Fungsi – Fungsi Matematika

Fungsi-fungsi yang bekaitan dengan matematika seperti terlihat pada tabel 5.7.

Tabel 5.7 Fungsi-Fungsi Matematika

Fungsi	Keterangan
Abs()	Mengembalikan nilai absolut dari argumen
Atn()	Mengembalikan nilai Arc Tangen dari argumen dalam bentuk radian
Cos()	Mengembalikan nilai Cosinus dari argumen dalam radian
Exp()	Mengembalikan nilai Eksponensial dari argumen
Log()	Mengembalikan nilai Logaritma dari argumen
Sin()	Mengembalikan nilai Sinus dari argumen dalam radian
Sqr()	Mengembalikan nilai Akar dari argumen
Tan()	Mengembalikan nilai Tengen dari argumen dalam radian

7. Fungsi – Fungsi String

Selain fungsi-fungsi numerik, Visual Basic juga mempunyai sejumlah fungsi-fungsi String diantaranya seperti pada tabel 5.8.

Tabel 5.8 Fungsi-Fungsi String

Fungsi	Keterangan
LCase()	Mengembalikan argumen string sebagai tipe data string huruf kecil
UCase()	Mengembalikan argumen string sebagai tipe data string huruf besar
Val()	Mengembalikan nilai number dari argumen string
Len()	Mengembalikan nilai number yang menunjukkan panjang string

Substring merupakan bagian dari suatu string. Untuk memperoleh substring dari suatu string dapat digunakan 3 fungsi berikut :

- Left(StringVal, length)

  Fungsi ini mengembalikan nilai string dari sebelah kiri sebanyak *length* karakter .

- Right(StringVal, length)

  Fungsi ini mengembalikan nilai string dari sebelah kanan sebanyak *length* karakter .

- Mid(StringVal, startVal, length)

  Fungsi ini mengembalikan nilai string dari *startVal* sebanyak *length* karakter .

Contoh :

```
Hurufkecil = LCase("Dwi Apri")    'HurufKecil = "stmik"
hurufbesar = UCase("Setyorini")   'HurufBesar = "STMIK"
Title = "STMIK Duta Bangsa"
lTitle = Left(Title, 3)           'lTitle = "STM"
rTitle = Right(Title, 5)          'RTitle = "angsa"
mTitle = Mid(Title, 3, 8)         'MTitle = "MIK Duta"
Panjang = Len(Title)              'length = 17
```

8. Fungsi – Fungsi Tanggal dan Waktu

Jika dalam suatu aplikasi diperlukan pengambilan tanggal atau waktu dari sistem, Visual Basic menyediakan fungsi – fungsi Now(), Date(), dan Time(). Lihat tabel 5.9.

Tabel 5.9 Fungsi-fungsi Tanggal dan Waktu

Fungsi	Keterangan
Now()	Mengembalikan tanggal danwaktu sistem
Date()	Mengembalikan tanggal dari sistem
Time()	Mengembalikan waktu dari sistem
DateDiff	Mendapatkan selisih dari dua buah tanggal

Untuk mencari selisih dari dua buah tanggal tidaklah sulit, karena dalam visual basic telah disediakan fasilitas untuk melakukan hal tersebut dengan menggunakan fungsi "***DateDiff*** ", yaitu sebuah fungsi yang digunakan untuk mendapatkan selisih dari dua buah tanggal. Dari fungsi *DafeDiff* kita bisa mendapatkan selisih hari, bulan dan tahun dari dua buah tanggal.

Cara penggunaan dari fungsi ini adalah sbb:

```
Dim hari, bulan, tahun

'Untuk mencari selisih hari
 hari=DateTime.DateDiff("d",CDate(Text1.Text),CDate(Text2.Text))

'Untuk mencari selisih bulan
 bulan = DateTime.DateDiff("m",CDate(Text1.Text),CDate(Text2.Text))

'Untuk mencari selisih tahun
 tahun = DateTime.DateDiff("yyyy",CDate(Text1.Text),CDate(Text2.Text))
```

Selain fungsi-fungsi diatas terdapat sebuah fungsi yaitu format() yang berkaitan dengan fungsi-fungsi Tanggal dan waktu.

```
Format(Ekpresi[,format[,firstdayofweek[, firstweekofyear]]])
```

Fungsi ini mengembalikan nilai variant atau string sesuai dengan format yang ditentukan.

9. Contoh Program

1. **Membuat Marquee untuk Form Caption**, dimana caption dari form yang kita gunakan akan berjalan dari kanan kekiri secara terus menerus. Kontrol yang digunakan adalah satu buah kontrol timer yang intervalnya kita set menjadi 100 atau bisa kita ganti sesuai keinginan kita.

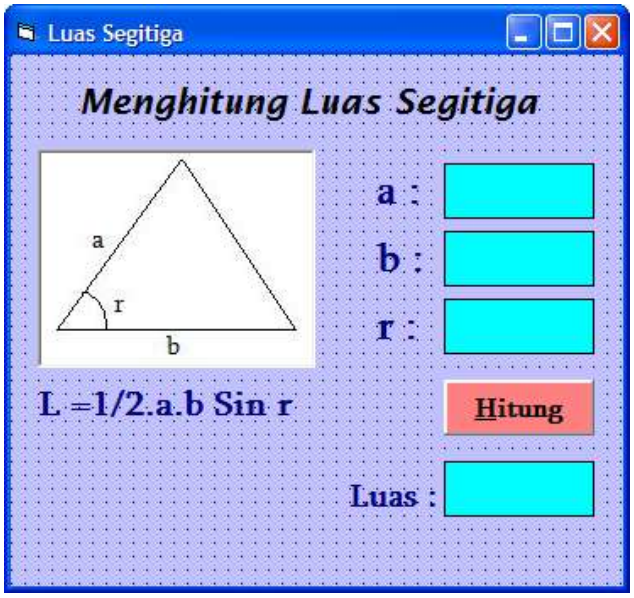
   Ketikkan kode program seperti berikut :

```
Private Sub Form_Load()
Form1.Caption = " [ Belajar VB 6.0 ] "
End Sub

Private Sub Timer1_Timer()
a = Left(Form1.Caption, 1)
b = Len(Form1.Caption)
c = Right(Form1.Caption, b - 1)
Form1.Caption = c + a
End Sub
```

2. **Menghitung Luas Segitiga**, jika diketahui panjang dua sisi segitiga dan besar sudut yang diapit oleh segitiga tersebut. Untuk menyelesaikan masalah tersebut dapat digunakan fungsi berikut : L = ½ .a.b Sin r

   Catatan: sudut r yang Anda masukkan dalam ukuran derajat, untuk itu perlu dirubah terlebih dahulu sebelum dimasukkan dalam fungsi sin yang disediakan oleh VB. (Rad = Phi/180)

Rancang formnya seperti berikut ini :



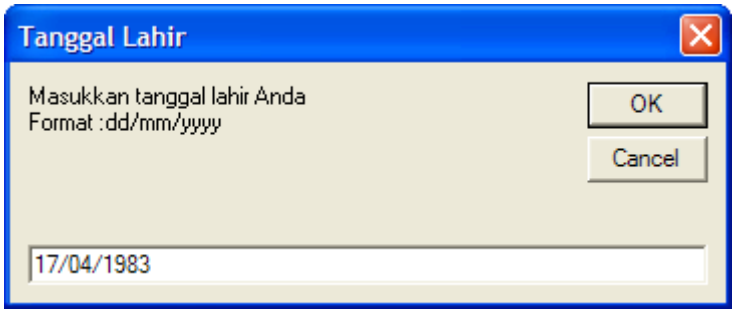
Ketik kode programnya sebagai berikut :

```
Private Sub cmdUsia_Click()
A = Val(txtA.Text)
B = Val(txtB.Text)
R = Val(txtR.Text)

hasil = 0.5 * A * B * Sin(R * (3.14 / 180))
txtLuas.Text = Round(hasil, 2)
End Sub
```

C. Latihan

1. Buatlah sebuah form dengan sebuah label, sebuah Textbox, dan 2 buah command Button. Textbox tersebut digunakan untuk memasukan data berupa angka dengan range antara 0 sampai 100000. Jika seorang user memasukan huruf atau angka yang berada diluar range maka akan ditampilkan pesan kesalahan dan meminta user untuk memasukkan data kembali. Jika masukan seorang user sudah benar maka akan ditampilkan pesan bahwa masukan sudah benar dan keluar dari aplikasi.
2. Buatlah program yang dapat menampilkan hari dan tanggal saat ini :
   Contoh Formnya :

3. Buat program untuk menghitung usia seseorang. Ketika program dijalankan muncul *inputbox* dan user diminta untuk memasukkan tanggal lahir. Tampilannya seperti berikut:



Setelah user memasukkan tanggal lahir, dan mengklik tombol OK, muncul *messagebox* yang menampilkan usia Anda. Tampilannya seperti berikut :

BAB VI
KONTROL PROGRAM

A. Pendahuluan

Penyelesaian kondisi atau pernyataan kondisi (Conditional Statement) merupakan suatu pernyataan yang menganalisa suatu keadaan dan mengambil keputusan berdasarkan pada hasil analisa itu. Hasil dari penyelesaian, jika kondisi benar maka akan dikerjakan instruksi tertentu, sedang jika kondisi salah, maka akan dikerjakan instruksi yang lain. Ini salah satu dari fungsi dalam kontrol program. Dalam visual basic terdapat dua kontrol program yang dapat digunakan yaitu logika percabangan dan logika perulangan.

B. Materi

1. Operator Kondisi

Untuk mengontrol alur program dalam Visual Basic, kita bisa menggunakan bermacam-macam operator kondisi. Pada dasarnya, operator kondisi ini mirip dengan operator aritmatik. Operator kondisi merupakan alat yang sangat hebat, dengan operator tersebut kita dapat melakukan perbandingan untuk memutuskan tindakan selanjutnya. Tabel 6.1 menunjukan operator kondisi yang digunakan dalam Visual Basic.

Tabel 6.1 Operator Kondisi

OPERATOR	MAKSUD
=	Sama dengan
>	Lebih besar
<	lebih kecil
>=	Lebih besar sama dengan
<=	Lebih kecil sama dengan
<>	Tidak sama dengan

2. Operator Logika

Sebagai tambahan untuk operator kondisi, ada beberapa operator logika yang ditambahkan pada Visual Basic sehingga memudahkan pembuatan alur program.

Tabel 6.2 Operator Logika

OPERATOR	MAKSUD
And	Keduanya harus bernilai True
or	Salah satu saja yang bernilai True
Xor	Salah satu boleh bernilai True tetapi tidak boleh keduanya bernilai True
Not	Bernilai False

3. Pernyataan If...Then...Else

a. Syntaks Umum

Syntak umum untuk pernyataan **if...then...else** adalah

```
IF <kondisi> THEN <kode program>
```

Bila <kondisi> bernilai True maka <kode program> akan dikerjakan.

```
IF <kondisi> THEN
     <blok kode program 1>
ELSE
     <blok kode program 2>
END IF
```

Bila <kondisi> bernilai True maka <blok kode program 1> akan dikerjakan, tetapi bila <kondisi> bernilai False maka <blok kode program 2> yang akan

```
IF <kondisi 1> THEN
     <blok kode program 1>
ELSEIF <kondisi 2> THEN
     <blok kode program 2>
ELSE
     <blok kode program 3>
END IF
```

Bila <kondisi 1> bernilai True maka <blok kode program 1> akan dikerjakan, kemudian bila <kondisi 2> bernilai True maka <blok kode program 2> akan dikerjakan, tetapi bila <kondisi 1> dan <kondisi 2> bernilai False maka <blok kode program 3> yang akan dikerjakan.

b. Contoh Program :

1) Membuat Program Input Password

Letakkan kontrol Image, TextBox, Label dan Command. Atur seperti gambar berikut :

Gambar 6.1 Program Input Password dengan Fungsi IF

Pengaturan property setiap object-nya adalah sebagai berikut :

Tabel 6.3 Property Object untuk Program Input Password

Object	Properties	Value
Form1	Caption	Form Password
	StartUpPostion	2-CenterScreen
Image1	Stretch	True
	Picture	Logo.jpg
	Visible	False
Label1	Caption	Input Password
Text1	Name	txtPass
	PasswordChar	*
	Text	<kosong>
Command1	Name	cmdOK
	Caption	&OK

Buka Jendela Code dan pada bagian Code Editor ketikkan kode programnya sebagai berikut :

```
Private Sub cmdOK_Click()
If txtPass.Text = "admin" Then Image1.Visible =
True
```

Simpan program dan jalankan :

1. Ketikkan sembarang teks pada txtPass lalu klik tombol OK, maka tidak terjadi apa-apa.
2. Ketikkan “admin” pada txtPass lalu klik tombol OK, maka gambar logo akan muncul

   Penjelasan kode program :

   If txtPass.Text = "oryn" **Then** Image1.Visible = True

   Kondisi

   Kode program yang dikerjakan jika kondisi *True*

   Modifikasi programnya menjadi seperti berikut :

```
Private Sub cmdOK_Click()
If txtPass.Text = "oryn" Then
    Image1.Visible = True
    MsgBox "Password Benar", vbOKOnly, "Sukses"
Else
    Image1.Visible = False
    MsgBox "Password Salah", vbOKOnly, "Gagal"
    txtPass.Text = ""
    txtPass.SetFocus
End If
```

   Jalankan program :

   1. Ketikkan sembarang teks pada txtPass lalu klik tombol OK, maka gambar tidak muncul dan muncul kotak pesan “Password Salah”. Klik tombol OK pada kotak pesan tersebut. Maka txtPass dikosongkan dan kursor akan aktif di txtPass (txtPass.Setfocus)
   2. Ketikkan “admin” pada txtPass lalu klik tombol OK maka gambar akan muncul dan muncul kotak pesan “Password Benar”.

   Catatan Tambahan :

   - Teks “admin” harus diketik huruf kecil semua. Ingat : data string bersifat *case sensitive* !
   - Agar password-nya tidak bersifat *case sensitive*, modifikasi pernyataan kondisinya menjadi : **If LCase**(txtPass.Text) = "admin" **Then**

- Fungsi **LCase** adalah untuk mengkonversi semua string yang diinput ke txtPass.Text menjadi huruf kecil, walaupun user menginputnya dengan huruf kapital.

TANTANGAN :

Modifikasilah struktur kontrol dan kode program di atas sehingga seorang user hanya mempunyai 3 kali kesempatan untuk mengetikkan password. Setelah 3 kali kesempatan dan password masih salah maka program akan otomatis berhenti.

2) Membuat program menentukan nilai seorang mahasiswa dengan kriteria sebagai berikut :

Tabel 6.4 Interval Nilai untuk Program Konversi Nilai

Interval Nilai	Nilai Huruf	Keterangan
80-100	A	Lulus
68-79	B	Lulus
56-67	C	Lulus
41-55	D	Remidi
0-40	E	Gagal

Atur form beserta kontrol-kontrol yang diperlukan seperti berikut :

Gambar 6.2 Program Konversi Nilai dengan Statement I

Ketikkan kode program sebagai berikut :

```
Private Sub txtNilai_Change()
nilai = Val(txtNilai.Text)
If nilai >= 80 And nilai <= 100 Then
    lblHuruf.Caption = "A"
    lblKet.Caption = "Lulus"
ElseIf nilai >= 68 And nilai <= 79 Then
    lblHuruf.Caption = "B"
    lblKet.Caption = "Lulus"
ElseIf nilai >= 56 And nilai <= 67 Then
    lblHuruf.Caption = "C"
    lblKet.Caption = "Lulus"
ElseIf nilai >= 41 And nilai <= 55 Then
    lblHuruf.Caption = "D"
    lblKet.Caption = "Remidi"
Else
    lblHuruf.Caption = "E"
    lblKet.Caption = "Gagal"
End If
End Sub
```

4. Pernyataan Select...Case

Cara lain untuk menangani pengambilan keputusan dalam sebuah program adalah dengan menggunakan Select…Case Statement, yang mampu menangani sejumlah kondisi dari satu variabel. Select...Case serupa dengan If ... Then ..... ElseIf, tetapi lebih efisien apabila percabangan bergantung kepada satu kondisi saja. Dengan menggunakan Select ... Case sebagai pengganti dari If ... Then ... Else If, akan membuat program menjadi lebih sederhana.

a. Syntaks Umum :

Format penggunaan *Select Case* :

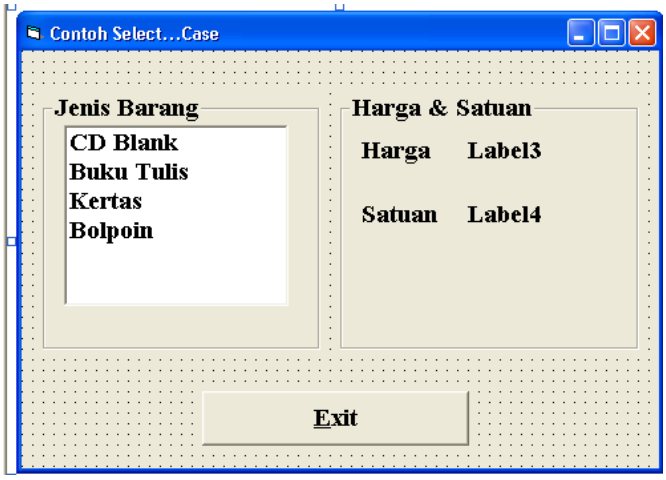
```
Select Case nama_variabel
   Case nilai_1
      Perintah yang akan dieksekusi jika memenuhi nilai_1
   Case nilai_2
      Perintah yang akan dieksekusi jika memenuhi nilai_2
   Case Else
      Perintah yang akan dieksekusi jika tidak memenuhi semua
End Select
```

Type data pada nama_variabel harus sama dengan nilai pada Case.

b. Contoh Program

1. Membuat program untuk mengetahui harga barang berdasarkan jenis barang yang dipilih di ListBox :

Atur Form beserta kontrol lain yang dibutuhkan seperti tampilan berikut:



Gambar 6.3 Program Harga Barang dengan Statement Select Case

Ketikkan kode program berikut :

```
Private Sub lstBarang_Click()
Dim Harga As Long
Dim Satuan As String

Select Case lstBarang.Text
    Case "CD Blank"
        Harga = 85000
        Satuan = "Box"
    Case "Buku Tulis"
        Harga = 36000
        Satuan = "Lusin"
    Case "Kertas"
        Harga = 28000
        Satuan = "Rim"
    Case "Bolpoin"
        Harga = 15000
        Satuan = "Pak"
End Select

Label3.Caption = Harga
Label4.Caption = Satuan
End Sub
```

Catatan : variablel nilai (lstBarang.Text) adalah string sehingga nilai yang ada pada Case seperti "CD Blank" adalah tipe data string juga.

TANTANGAN :

- Modifikasi program Harga Barang di atas (gambar 6.3). Tambahkan input jumlah barang, tambahkan pula output output diskon dan total yang harus dibayar.
- Diskon sebesar 5% dari TotalHarga. Diberikan diskon jika jumlah beli>=5, selain itu tidak mendapat diskon.
- Total Bayar = Total Harga-Diskon

2. Buka kembali contoh program konversi nilai pada gambar 6.2. Anda dapat mengganti statement IF dengan statement Select...Case. Berikut kode programnya:

```
Private Sub txtNilai_Change()
Dim Nilai As Single
Nilai = txtNilai.Text

Select Case Nilai
    Case Is >= 80
        lblHuruf.Caption = "A"
        lblKet.Caption = "Lulus"
    Case Is >= 68
        lblHuruf.Caption = "B"
        lblKet.Caption = "Lulus"
    Case Is >= 56
        lblHuruf.Caption = "C"
        lblKet.Caption = "Lulus"
    Case Is >= 41
        lblHuruf.Caption = "D"
        lblKet.Caption = "Remidi"
    Case Else
        lblHuruf.Caption = "E"
        lblKet.Caption = "Gagal"
End Select
End Sub
```

Keyword **Is** disini untuk menentukan kondisi. Biasa digunakan untuk data numerik.

Atau bisa juga memakai kode program seperti berikut:

```
Private Sub txtNilai_Change()
Dim Nilai As Single
Nilai = txtNilai.Text

Select Case Nilai
    Case 80 To 100
        lblHuruf.Caption = "A"
        lblKet.Caption = "Lulus"
    Case 68 To 79
        lblHuruf.Caption = "B"
        lblKet.Caption = "Lulus"
    Case 56 To 67
        lblHuruf.Caption = "C"
        lblKet.Caption = "Lulus"
    Case 41 To 55
        lblHuruf.Caption = "D"
        lblKet.Caption = "Remidi"
    Case Else
        lblHuruf.Caption = "E"
        lblKet.Caption = "Gagal"
End Select
End Sub
```

5. Pernyataan Do...Loop

Visual Basic mendukung beberapa versi statement *Do*. Looping (perulangan) dengan menggunakan *While* mungkin yang paling populer digunakan dalam pemrograman Visual Basic. Seperti Statement *If…Then*, *do While* juga membutuhkan ekpresi perbandingan untuk keluar dari looping

a. Syntaks Umum

Ada beberapa syntak penggunaan untuk pernyataan Do..Loop

```
a)  Do While <kondisi>
       <VB statement>
    Loop

b)  Do
       <VB statement>
    Loop While <kondisi>
```

Keterangan : VB Statement akan diulang **selama** <kondisi> bernilai TRUE. Pengulangan berhenti bila <kondisi> sudah bernilai FALSE.

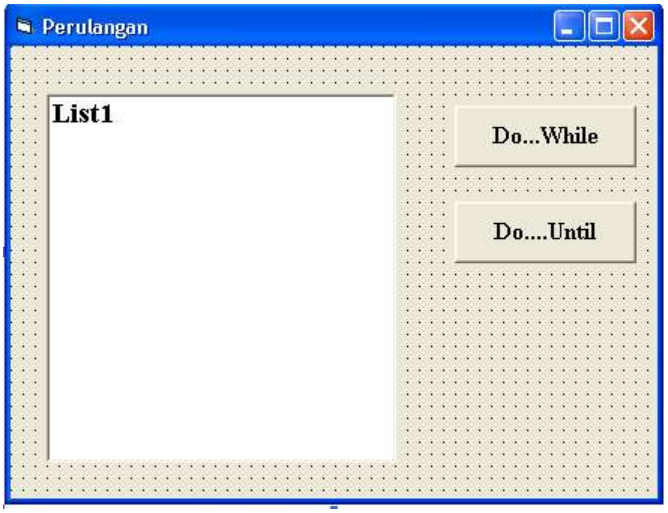
```
c)   Do Until kondisi
        VB statement
     Loop

d)   Do
        VB statement
      Loop Until kondisi
```

Keterangan : VB Statement akan diulang **sampai** <kondisi> bernilai TRUE. Pengulangan berhenti bila <kondisi> sudah bernilai FALSE.

b. Contoh Program

1. Menmpilkan angka 0 sampai 10 di ListBox dengan perintah Do...While dan Do...Until.

Atur Form beserta kontrol yang dibutuhkan seperti pada gambar 6.4



Gambar 6.4 Program Menampilkan Angka dengan Do...Loop

Ketik kode programnya seperti berikut ini :

```
Private Sub cmdUntil_Click()
lstHasil.Clear
i = 0
Do Until i > 10
 lstHasil.AddItem "Angka Ke- " & i
 i = i + 1
Loop
End Sub

Private Sub cmdWhile_Click()
lstHasil.Clear
i = 0
Do While i =< 10
 lstHasil.AddItem "Angka Ke- " & i
 i = i + 1
Loop
End Sub
```

Penjelasan program :

- Kode program tersebut akan menampilkan hasil yang sama.
- Perhatikan pada cmdUntil_Click(). Program akan dijalankan **sampai** i>10
- Perhatikan pada cmdWhile_Click(). Program akan dijalankan **selama** i=<10
- Kedua statement tersebut memiliki arti yang sama

2. Cobalah ganti syntaks statement Do Until dan Do While di atas, dengan kode program berikut ini, lihat hasilnya:

```
Private Sub cmdUntil_Click()
lstHasil.Clear
i = 0
Do
    lstHasil.AddItem "Angka Ke- " & i
    i = i + 1
Loop Until i > 10
End Sub

Private Sub cmdWhile_Click()
lstHasil.Clear
i = 0
Do
    lstHasil.AddItem "Angka Ke- " & i
    i = i + 1
Loop While i =< 10
End Sub
```

6. Pernyataan While...Wend

Pengulangan While…Wend akan mengeksekusi sekumpulan statement-statement perintah selama suatu kondisi itu benar. Looping atau perulangan yang menggunakan While..Wend ini mempunyai syntak sebagai berikut :

a. Syntaks Umum

Ada beberapa syntak penggunaan untuk pernyataan While...Wend

```
While <kondisi>
     VB Statement
Wend
```

Keterangan : Jika kondisi benar, maka semua statement akan dieksekusi dan ketika mencapai baris Wend, control akan kembali lagi ke statement While untuk mengevaluasi kembali nilai dari kondisi, jika nilai dari kondisi masih memenuhi syarat atau benar maka proses loop/perulangan akan terjadi lagi. Jika nilai kondisinya Salah, maka program akan keluar dari loop dan mengeksekusi perintah-perintah yang ada setelah Wend.

b. Contoh Program

1. Contoh Statement While berikut akan mengevaluasi nilai numerik yang dimasukkan user lewat keyboard, dan kondisi yang dievaluasi adalah selama nilainya lebih besar atau sama dengan nol, jika nilainya negatif maka program akan berhenti.

```
Number = 0
While Number >= 0
   Number = InputBox("Silakan masukkan nilai yang lain ?")
Wend
```

2. Contoh 1 di atas bisa dimodifikasi. Letakkan sebuah command button dan sebuah textbox pada form kemudian, ketikkan listing program berikut dalam tombol command button tersebut. (atur posisinya dengan benar).

```
Private Sub Command1_click()
Number = 0
While number >= 0
   Total = Total + Number
   Number = InputBox("Silakan masukkan nilai yang lain ?")
Wend
Text1.Text = Total
End Sub
```

7. Pernyataan For…Next

Perulangan dengan For ... Next merupakan salah satu struktur perulangan yang sering terdapat pada banyak bahasa pemograman, perulangan For ... Next menggunakan suatu variabel yang disebut counter untuk melakukan penambahan secara otomatis, sesuai dengan nilai awal dari variabel

tersebut. Perulangan dengan For digunakan untuk mengulang statement atau satu blok statement berulang kali, sejumlah yang ditentukan.

a. Syntaks Umum

```
FOR <pencacah> = <awal> TO <akhir> [STEP <langkah>]
<blok kode program>
NEXT <pencacah>
```

- <pencacah> adalah variabel (tipe: integer) yang digunakan untuk menyimpan angka pengulangan.
- <awal> adalah nilai awal dari <pencacah>.
- <akhir> adalah nilai akhir dari <pencacah>.
- <langkah> adalah perubahan nilai <pencacah> setiap pengulangan. Sifatnya optional (boleh ditulis ataupun tidak). Bila tidak ditulis maka nilai <langkah> adalah 1.

b. Contoh Program

1. Program menampilkan angka dari 0 sampai 10 dan sebaliknya (10 sampai 0)

   Atur Form dan Kontrol yang dibutuhkan seperti gambar 6.4. Ganti Caption pada Command, masing-masing menjadi ”For Next 1” da For Next2”.

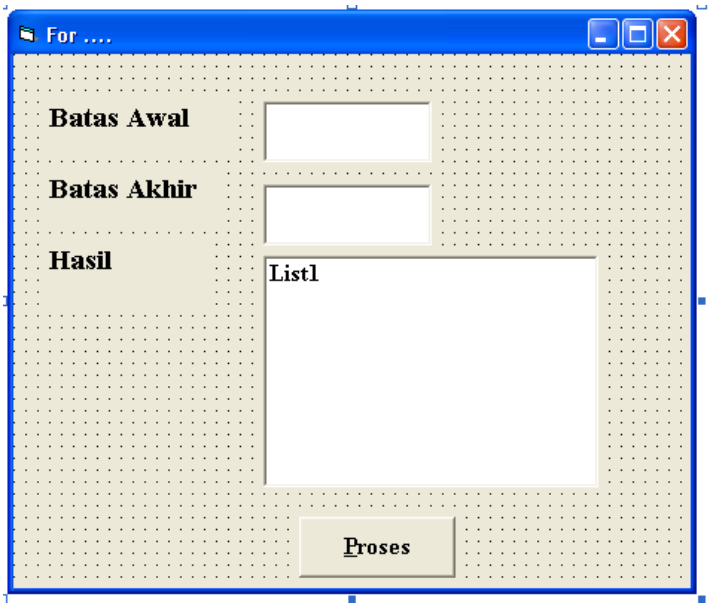
   Ketikkan kode program berikut ini :

```
Private Sub cmdFor1_click()
‘Menampilkan angka 0 sampai 10 di ListBox
lstHasil.Clear
   For i = 0 To 10
      lstHasil.AddItem "Angka " & i
   Next i
End Sub

Private Sub cmdFor2_click()
‘Menampilkan Angka 10 sampai 0 di ListBox
lstHasil.Clear
   For i = 0 To 10 STEP -1
      lstHasil.AddItem "Angka " & i
   Next i
End Sub
```

   Jika Anda menginginkan angka dengan kelipatan 2 dst, maka tambahkanlah perintah STEP,contoh :

For i = 0 To 10 STEP 3, akan menampilkan angka dari 0 sampai 10 dengan kelipatan 3. Hasilnya : 0, 3, 6, 9

2. Menampilkan angka genap pada ListBox. Bilangan genap adalah bilangan yang habis dibagi 2. Memungkinkan user untuk menginputkan batas awal dan akhir. Atur form dengan meletakkan ListBox,2 TextBox , dan Command untuk proses. Buat properti Column = 3 untuk ListBox



Gambar 6.5 Menampilkan bilangan genap dengan For Next

Ketikkan kode program seperti berikut

```
Private Sub cmdProses_Click()
Dim i, Awal, Akhir As Single
lstHasil.Clear

Awal = txtAwal.Text
Akhir = txtAkhir.Text

For i = Awal To Akhir
    If i Mod 2 = 0 Then
        lstHasil.AddItem i
    End If
Next i

End Sub
```

Bagaimana jika Anda diminta untuk menampilkan bilangan ganjil?? Silakan Anda coba sendiri. Okey ☺

C. Latihan

Buat program Registrasi Hotel, atur tampilan form beserta kontrol yang dibutuhkn sebagai berikut :

Pada program di atas user dipersilahkan untuk mengisi namanya, jenis kelaminnya laki-laki atau perempuan, kemudian memilih jenis kamar yang diinginkan, kemudian memilih fasilitas apa saja yang diinginkan.

Tiap kamar memiliki harga yang berbeda-beda. Untuk pilihan jenis kamar gunakan Option, di sini user hanya dapat memilih salah satu saja. Sedangkan untuk fasilitas dapat menggunakan Check, di mana user dapat tidak memilih dan bebas untuk memilih yang mana saja.

Kita asumsikan harga kamar adalah sebagai berikut :

Jenis Kamar	Harga
Standard	150000
Suite	250000
Silver	450000

Jenis Fasilitas	Harga
Sauna	50000
Fitness	75000
Massage	150000

- Kemudian tampilkan harga total pembayarannya dalam bentuk Message Box dengan klik tombol Daftar.
- Minta input pembayaran dengan menggunakan InputBox. Validasilah agar pembayaran mencukupi total pembayaran. Jika pembayaran tidak mencukupi total pembayaran, maka InputBox akan muncul terus samapi pembayaran mencukupi.
- Jika terdapat uang kembali tampilkan jumlah uang kembaliannya dengan MesageBox.
- Dan ucapkan terima kasih. Setelah itu resetlah kembali form ke keadaan semula sebelum diisi.

BAB VII

PROCEDURE DAN FUNCTION

A. Pendahuluan

Dalam kenyataan seringkali program yang harus ditulis cukup panjang, sehingga kesalahan yang mungkin dibuat oleh seorang *programmer* semakin besar. Untuk mengatasi masalah tersebut dapat dilakukan dengan memecah program tersebut menjadi bagian-bagian kecil (rutin) tetapi tetap logis. Rutin-rutin kecil tersebut akan membuat penelusuran dan perawatan program menjadi lebih mudah dan terstruktur. Rutin-rutin kecil tersebut sering disebut dengan nama prosedur.

Dalam Visual Basic terdapat 2 macam prosedur yaitu :

a. Prosedur umum (*general purpose procedure*) merupakan prosedur yang ditemukan di dalam daftar *Drop Down* pada jendela kode.

b. Prosedur Kejadian (*Event procedure*), prosedur yang berisi kode yang dijalankan ketika suatu kejadian dari kontrol di bangkitkan.

Sedangkan prosedur-prosedur yang ditambahkan dalam sebuah program tersebut disebut *subprogram*. Ada 2 jenis subprogram dalam Visual Basic yaitu prosedur subrutin (*subroutine procedures*) dan prosedur fungsi (*function procedures*).

B. Materi

a) Sub Rutin

Subrutin merupakan prosedur umum (*general porpuse procedure*) yang ditulis dan ditambahkan dalam program. Format penulisan sebuah subrutin adalah sebagai berikut:

```
Sub nama_subRutin [(argumen)]
            .
            .
            .
End Sub
```

Setiap kali prosedur dipanggil, maka pernyataan diantara Sub dan End Sub akan dijalankan. Argumen pada prosedur adalah nilai yang akan dilewatkan saat pemanggilan prosedur.

Contoh :

```
Sub TotalPembelian()
    ' Inisialisasi varibel
    Dim Total As Currency
    Dim Disc  As Single

    'mendefinisikan nilai variabel
    Total = txtTotal.Text
    Disc = 0.2

    lblTotal.Caption = Total - Total * Disc
End Sub
```

Sedangkan untuk memanggil sebuah subrutin digunakan format sebagai berikut :

```
[Call] SubName [(Argumen list)]
```

Untuk memanggil subrutin dari contoh diatas dapat digunakan kode sebagai berikut:

```
Call TotalPembelian()
```

atau

```
TotalPembelian
```

Untuk memanggil sebuah subrutin dapat digunakan perintah *Call* diikuti dengan nama subrutin diikuti dan tanda kurung atau langsung memanggil nama subrutin saja.

b) Fungsi

Subrutin dan Fungsi sebenarnya mirip yaitu prosedur umum (*general purpose procedure*) yang ditulis dan ditambahkan dalam program. Bedanya Fungsi mengembalikan nilai sedangkan Subrutin tidak. Format penulisan sebuah Fungsi adalah sebagai berikut.

```
Function nama_Fungsi [(argumen)] as [ReturnValType]
            .
            .
            .
End Function
```

Contoh :

```
Function Total_Pembelian(Total As Currency)
' Inisialisasi varibel
  Dim Disc  As Single

' Mendefinisikan nilai variabel
  Disc = 0.2
  Total_Pembelian = Total - Total * Disc

End Function
```

Pada contoh di atas merupakan fungsi TotalPembelian yang dimodifikasi. Total tidak lagi diambil dari nilai textbox (txtTotal), melainkan dikirim melalui suatu argumen. Argumen yang dikirimkan melalui suatu subrutin atau fungsi dapat digunakan langsung seperti sebuah variabel.

Untuk memanggil fungsi dari contoh diatas dapat digunakan kode sebagai berikut:

```
'Memasukkan nilai Total sebesar 200
Total_Pembelian (200)
```

c) Modul (Module)

Penulisan sebuah modul dapat dilakukan pada sebuah file berekstensi .Bas. File tersebut dapat dibuat melalui menu *Project-Add Module*. Secara default Visual Basic memberikan nama *Module1*.

Pada Module, Anda dapat mendeklarasikan variable dengan awalan DIM, PRIVATE maupun PUBLIC, dimana awalan DIM dan PRIVATE membentuk variable modul level (hanya berlaku didalam pemakaian variable bersangkutan), sedangkan awalan PUBLIC akan membentuk variable global yang akan berfungsi bagi keseluruhan program.

Contoh :

```
Dim A As Integer        'Variabel A adalah modul level
Private B As Integer    'Variabel B adalah modul level
Public C As Integer     'Variabel C dapat digunakan
                        'oleh program keseluruhan
```

Subrutin dapat ditulis pada module. Subrutin tersebut dapat dipanggil kapan saja dari aplikasi yang sama dengan format sebagai berikut:

```
[nama_module.]nama_subrutin
```

Contoh :

```
Module1.TotalPembelian
```

atau

```
TotalPembelian
```

Pemanggilan dapat dilakukan dengan menyebut nama module terlebih dahulu diikuti dengan tanda titik kemudian nama subrutin atau langsung nama subrutin.

d) Contoh Program

Membuat program untuk mengkalkulasi temperatur Celcius ke temperatur Fahrenheit. User diminta untuk memasukkan suhu dalam Fahrenheit dengan fasilitas InputBox, setelah diklik OK muncul *messagebox* yang menampilkan konversi suhu dalam Celcius.

Ketikkan kode program berikut ini :

```
'Fungsi Konversi Ke Celcius
Function Celcius(fDerajat)
Celcius = (fDerajat - 32) * 5 / 9
End Function

Private Sub Form_Load()
'Menampilkan InputBox untuk Input Suhu dalam Fahrenheit
'Celcius(Temp): memanggil fungsi Celcius dengan besar suhu
                'sesuai yang dimasukkan diinputbox
'Perintah Round untuk menampilkan angka desimal dengan
                'mengatur berapa angka di belakang koma

Dim Temp
Temp = InputBox("Silakan masukkan suhu dalam Fahrenheit", "Suhu")
MsgBox "Suhu dalam Celcius adalah : " & _
Round(Celcius(Temp), 2) & " Derajat", vbOKOnly, "Suhu dalam
Celcius"
End Sub
```

C. Latihan

Modifikasi contoh program konversi suhu di atas, buat supaya user bisa memilih menu untuk konversi suhu. Contoh tampilan programnya seperti berikut :

User memasukkan nilai dan memilih satuan suhu yang akan dikonversi. User juga menentukan nilai desimal dari hasil yang akan ditampilkan. Kemudian secara otomatis, hasil akan ditampilkan dalam Celcius, Fahrenheit, Kelvin dan Reamur

BAB VIII
PEMROGRAMAN MULTIMEDIA

A. Pendahuluan

”The American Heritage Dictionary” mendefinisikan bahwa multimedia adalah kombinasi dari penggunaan beberapa media seperti film, slide, music,penerangan dengan text, image, khusunya untuk tujuan pendidikan dan hiburan. Unsur-unsur seperti text, audio (narasi, dialog, sound effect), musik, film, video, fotografi, animasi dan grafik merupakan media pendukung yang tergabung dan terintegrasi menjadi satu-kesatuan karya multimedia. Bentuk interactive multimedia termasuk di dalamnya website, cdrom interactive, program/software, presentasi, tutorial, help sectiondal an bahkan game.

Sekarang yang jadi pertanyaan, bagaimana jika produk-produk multimedia tidak dihasilkan dari aplikasi program multimedia ? Mampukah program-program komputer yang sifatnya umum dapat digunakan untuk mendesain produk multimedia? Jawabannya tentu tergantung pogram itu sendiri, apakah software programnya mendukung konsep multimedia atau tidak. Umumnya perkembangan teknologi multimedia tidak hanya tergantung pada perkembangan software semata, tetapi terkait dengan perangkat elektronik.

B. Materi

1. Dasar Pemrograman Multimedia

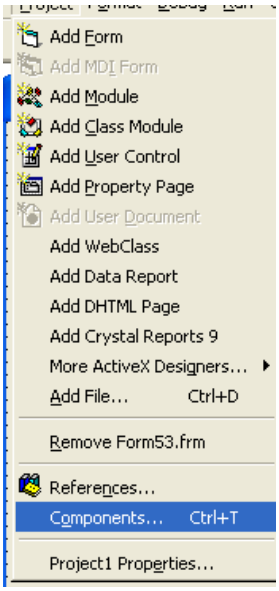
   Visual basic merupakan salah satu bahasa pemrograman yang mendukung konsep multimedia. Sejak diluncurkan pertama kali, visual basic yangzhou bekerja di sistem operasi windows terus menambah fasilitas untuk pemrograman multimedia. Salah satu fasilitas tersebut adalah sebuah kontrol Active X yang disertakan dalam visual basic edition dengan nama kontrol multimedia MCI (Media Control Interface), misalnya CD Player, VCR dan Video Player.

   Sebenarnya dalam visual basic masih terdapat tiga kontrol lain seperti MCIWndx Control, Microsoft ActiveMovie Control, Windows Media Player. Tetapi dari kesemuanya, kontrol multimedia MCI adalah yang terlengkap

serta merupakan kontrol langsung berhubungan dengan visual basic yangzhou melibatkan properti dan event.

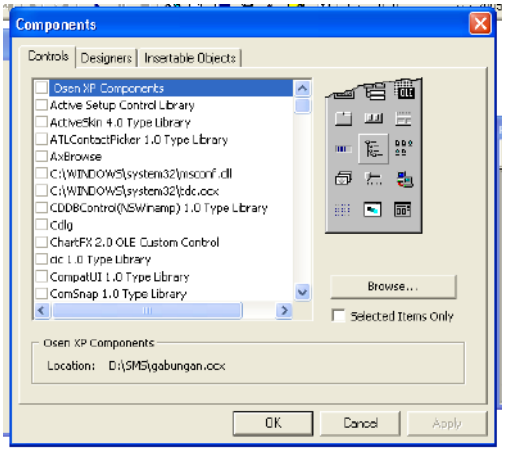
Sebelum mendesain program multimedia dengan visual basic 6.0 terlebih dahulu aktifkan kontrol multimedia yang ingin digunakan. Langkah-langkah untuk mengaktifkan kontrol-kontrol multimedia adalah sebagai berikut :

- Klik menu project dan klik menu popup components atau untukmempersingkat kerja, tekan kombinasi tombol Ctrl+T



Gambar 8.1 Bagian dari menu project

- Selanjutnya visual basic 6.0 menampilkan kotak dialog components, seperti gambar 8.2.
- Klik tab control dan lakukan scrolup (gulungk e atas) untuk menampilkan kontrol-kontrol multimedia



Gambar 8.2 Kotak Dialog Components

2. Menentukan Tipe Peralatan Multimedia

Sebelum menggunakan tombol-tombol pada kontrol multimedia, peralatan multimedia perlu diaktifkan dengan valid yaitu dengan perintah DeviceType. Tujuan dari proses ini adalah agar kontrol tersebut secara otomatis dapat dikonfigurasikan sewaktu program berjalan. Proses penempatan ini biasanya dilakukan lewat kode program dalam suatu procedure, misalnya form_load. Contoh penuliasan dari properti DeviceType adalah sebagai berikut :

MMControl1.deviceType=”WaveAudio”

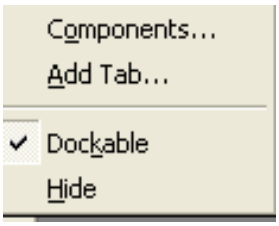
Tabel 8.1 Peralatan Multimedia

No.	Peralatan Multimedia	DevName	Keterangan
1	Video (file .avi)	AVIVideo	Format video Microsoft AVI
2	Audio CD	CDAudio	CD Musik melalui drive CD-ROM
3	Digital Tape	DAT	Peralatan digital tape (DAT)
4	Digital Video	DigitalVideo	Fata Video digital
5	Video	MMMovie	Format film multimedia
6	Scanner	Scanner	Peralatan Scanner
7	MIDI sequencer	Sequencer	Data sequencer MIDI
8	Videotape	VCR	Perekam kaset video
9	Videodisc	Videodisc	Perekam Video Disc
10	Wave (file .wav)	WaveAudio	File Audio microsoft windows
11	Buatan pemakai	other	Tipe multimedia buatan pemakai

3. Kontrol Media Player

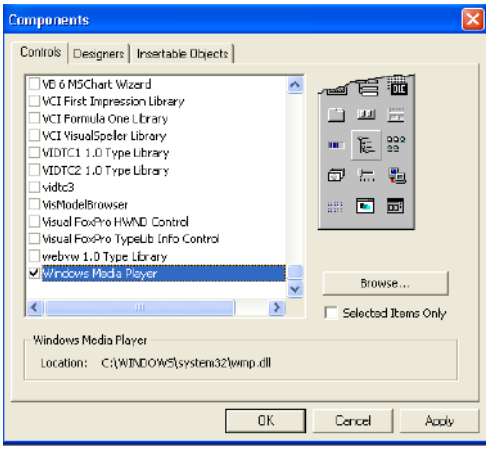
Kontrol media player merupakan salahsa tu komponen pada visual basic 6.0 yang digunakan untuk menjalankan aplikasi video dan audio pada banyak file format multimedia. Untuk mengaktifkan kontrol tersebut, langkah-langkah yang dilakukan yaitu :

- Lakukan klik pada menu project, components, atau dengan cara lain yaitu dengan melakukan klik kanan pada toolbox dank lik menu components.



Gambar 8.3 Menu Shorcut Components

- Selanjutnya sebuah kotak dialog components ditampilkan dan pilih tab controls.
- Berikan tanda cek (√) pada kontrol windows media player. Kontrol ini membutuhkan file yang tersimpan dalam folder system32 yang bernama MSDXM.OCX



Gambar 8.4 memilih kontrol media player

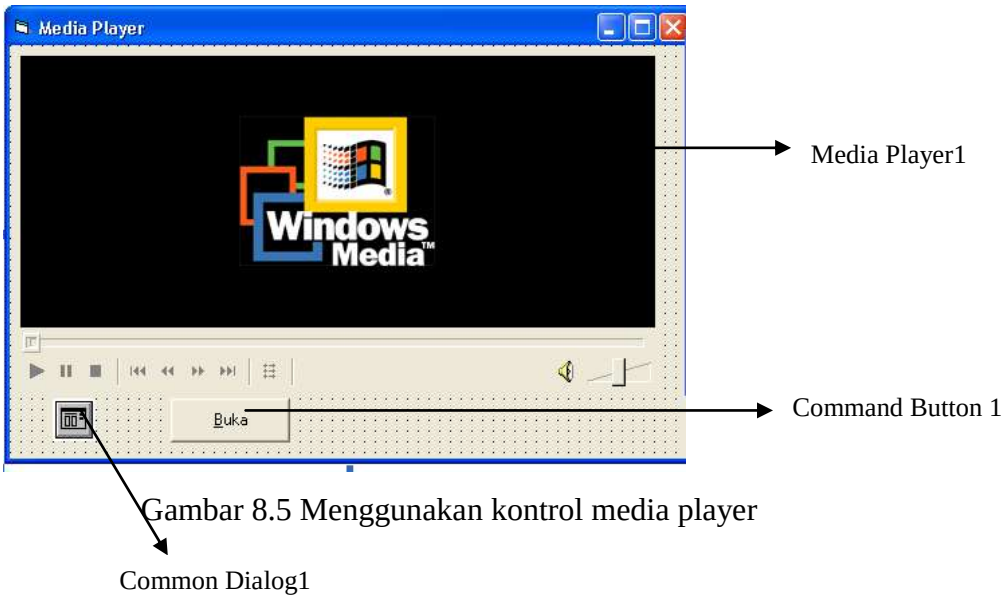
- Klik tombol OK untuk menutup kotak dialog components dan pada toolbox akan terdapat kontrol media player.
- Untuk menempatkan kontrol media player pada form, yaitu klik ikon  pada toolbox dan lakukan drag pada form untuk menentukan luas lokasi yang digunakan.
- Sedangkan untuk menghubungkan kontrol media player denganfile multimedia. Perintah yang digunakan adalah :

  MediaPlayer1.FileName=Path & <nama file>

  Misalnya :

  Media Player1.FileName = ”E:\Film\Coba.mpeg”

Contoh :



Gambar 8.5 Menggunakan kontrol media player

Ketikan Kode Program pada tombol Buka seperti berikut :

```
Private Sub Command1_Click()
    CommonDialog1.ShowOpen
    MediaPlayer1.FileName = _
    CommonDialog1.FileName
End Sub
```

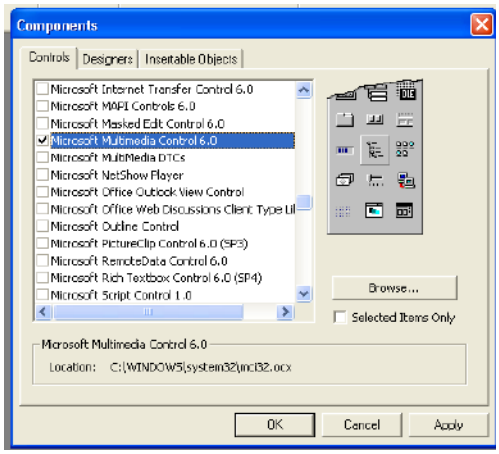
4. Kontrol Multimedia Control 6.0

Kontrol multimedia control 6.0 adalah sebuah kontrol ActiveX yang disertakan dalam Microsoft Visual Basic Profesional Edition. Dengan kontrol ini dapat dibuat program yang mampu menyediakan fungsi-fungsi multimedia pendukung driver MCI (Media Control Interface), misalnya memutar film, memainkan CD Audio, merekam Wave Audio dan sebagainya. Disamping itu kontrol Multimedia MCI juga menyediakan sekumpulan tombol bergaya compact disk untuk memainkan dan merekam media pada program anda.

Kontrol ActiveX Multimedia Control 6.0 dapat ditambahkan ke dalam toolbox dengan langkah-langkah sebagai berikut :

1. Aktifkan kotak dialog components dengan cara klik kanan mouse pada toolbox atau klik menu project dan pilih menu components.

2. Pda tab controls, pilih dan berikan tanda cek pada Microsoft Multimedia Control 6.0 adalah MCI.OCX
3. Klik tombol OK untuk menutup kotak dialog components.



Gambar 8.6 Memilih kontrol multimedia control 6.0

4. Untuk menempatkan kontrol multimedia control 6.0 pada form, yaitu klik icon  pada toolbox danish lakukan drag pada form untuk menentukan luas lokasi yang digunakan.
5. Kontrol multimedia control 6.0 mempunyai serangkaian tombol perintah yang berfungsi secara otomatis pabila terdapat peralatan multimedia yang sedang terbuka dan kontrol dinyalakan. Tombol-tombol tersebut antara lain Prev, Next, Play, Pause, Back, Step, Stop, Record dan eject. Meskipun anda dapat menambahkan fasilitas khusus untuk tombol-tombol ini yaitu dengan menuliskan event procedure, tetapi anda boleh tidak mengkonfigurasikan lagi mengingat setting tombol yang sudah disediakan dapat langsung digunakan dengan baik.

   Sebelum anda mengenal lebih jauh tentang fungsi dari kontrol multimedia control 6.0, perlu kiranya dipahami beberapa property khusus yang diediakan oleh kontrol ini, yaitu :

   - Devive Type

     Property ini digunakan untuk menentukan device yang akan digunakan oleh kontrol multimedia control 6.0, yaitu AVIVideo, DAT, CDAudio, DigitalVideo, MMMovie, Other, Overlay, Scanner, Sequencer, VCR atau WaveAudio. Biasanya kode program diletakkan

dalam event procedure form load agar kontrol secara otomatis dapat dikonfigurasikan sewaktu program berjalan.

Bentuk penulisan dari properti deviceType sebagai berikut :

MMControl.DeviceType=DevName

- FileName

  Properti ini berisi nama file yang ditentukan untuk dimainkan oleh kontrol multimediaMCI

  Bentuk penulisaan :

  MMControl.Filename=<nama file>

- Wait

  Properti ini digunakan untuk menentukan agar kontrol menunggu perintah selanjutnya diselesaikan terlebih dahulu sebelum tugas pengendalian dikembalikan pada aplikasi.

  Bentuk penuliasn :

  MMControl.Wait=<boolean>

- Command

  Properti ini digunakan untuk memberikan perintah pada kontrol yang akan dieksekusi, yaitu Open, Close, Playe, Pause, Stop, Back, Step, Prev, Next, Seek, Record, Eject, Sound atau Save.

  Bentuk penulisan :

  MMControl,Command=<Perintah String>

- Sharable

  Properti ini digunakan untuk menentukanagar program lain mampu menggunakan device MCI yang sama.

  Bentuk penulisan :

  MMConrol.Sharable=<True/False>

- Notify

  Properti ini digunakan agar kontrol multimedia MCI memberitahu atau tidak apabila perintah ”Open” dan ”Play” telah selesai dilaksanakan.

  Bentuk penulisan :

  MMControl.Notify=<True/False>

Contoh Program :

Desain Form seperti berikut :

Gambar 8.7 Menggunakan kontrol MMControl

Ketikan Kode Program Seperti Berikut :

```
Private Sub Command1_Click()
    MMControl1.Notify = False
    MMControl1.Wait = True
    MMControl1.Shareable = False
    CommonDialog1.ShowOpen
    MMControl1.FileName = _
    CommonDialog1.FileName
    Text1.Text = CommonDialog1.FileName
    MMControl1.Command = "open"
    MMControl1.Command = "play"
End Sub

Private Sub Command2_Click()
    End
End Sub
```

5. Memainkan File-File Video

   File video (film) adalah file yang berisi rangkaian gambar dan suara yang diolah sedemikian rupa sehingga terbentuk suatu gerakkan yang terus menerus. Untuk menghasilkan suatu gerakkan, tentu diperlukan suatu

kecepatan yang cukup sehingga dapat mengelabuhi mata orang yang melihat film tersebut.

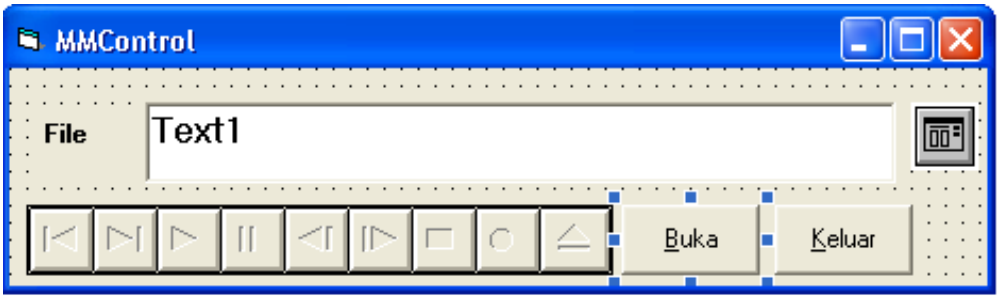
Terdapat dua kecepatan yang harus dipahami dalam mebuat sebuah animasi pada film. Pertama adalah jumlah gambar per detik yang ditampilkan saat proses presentasi (playback rate). Kedua adalah jumlah gambar yang berbeda yang muncul per detik (sampling rate/update rate).

File video merupakan sarana yang ampuh untuk menambahkan animasi, petunjuk langkah demi langkah, informasi bantuan atau hanya untuk membuat agar program yang dibuat lebih memiliki sentuhan pribadi.

a. File AVI

file AVI (Audio Video Interleaving) merupakan file standart yang dimiliki oleh sistem operasi Microsoft Windows untuk memainkan data video dan audio secara digital. File AVI adalah file hasil kopresi gambar-gambar dan atau suara yang terangkai dan membentuk suatu file baru yang dapat dimainkan secara bergerak dan bersuara.

Contoh Program :



Ketikan Kode Proram Berikut :

```
Private Sub Command1_Click()
        MMControl1.Notify = False
        MMControl1.Wait = True
        MMControl1.Shareable = False
        MMControl1.DeviceType=”AVIVideo”
        CommonDialog1.ShowOpen
        MMControl1.FileName = _
        CommonDialog1.FileName
        Text1.Text = CommonDialog1.FileName
        MMControl1.Command = "open"
        MMControl1.Command = "play"

End Sub
```

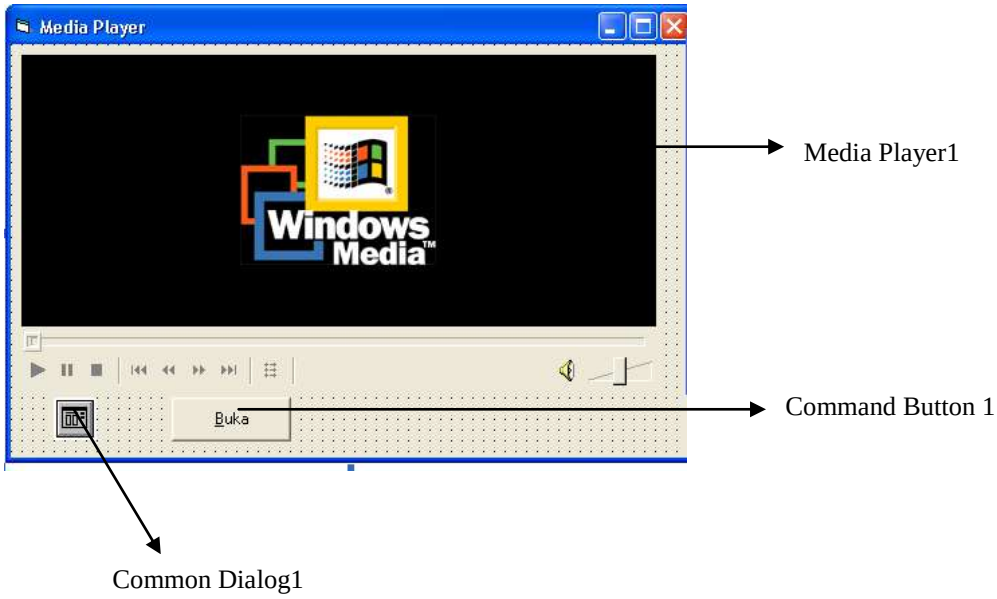
```
Private Sub Command2_Click()
End
End Sub
```

b. File DAT

File DAT (Digital Audio Tape) adalah file yang diproses dengan peralatan digital tape. Ukuran dari file ini cukup besar ibandingkan dengan file standar microsoft yaitu AVI, tetapi gambar yang dihasilkan jauh lebih bagus.

Banyak peralatan (player) yang dapat digunakan untuk menjalankan file DAT ini. Visual Basic 6.0 sendiri mempunyai kontrol program yang mampu untuk mengoperasikan file tersebut. Salah satu kontrol programnya adalah Multimedia MCI. Sedang untuk standar windows dapat digunakan kontrol program windows media player.

Contoh Progam :



Ketikan Kode Program pada tombol Buka seperti berikut :

```
Private Sub Command1_Click()
    CommonDialog1.ShowOpen
    MediaPlayer1.FileName = _
    CommonDialog1.FileName
End Sub
```

6. Memainkan File Audio WAV dan MIDI

a. File WAV

File .WAV (Waveform) adalah file untuk menyimpan suara yang direkam dengan teknologi dari micrisift. Dalam lingkungan windows, format file ini telah diakui sebagai format audio standar. Keuntungan dari file .WAV adalah anda dapat menyimpan suara apa saja dari berbagai sumber, seperti mikrofon, pemutar CD atau bahkan tape recorder.

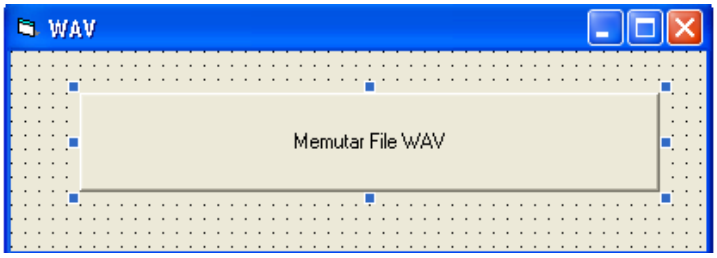
Sebelum disimpan pada file .WAV, suara analog harus diubah ke bentuk suara digital, atau dengan kata lain suara analog harus dicuplic dan dikuantisasi terlebih dahulu. Suara analog dapat dicuplic dengan berbagai frekwensi yaitu 8 KHz, 11 KHz, 22 KHz. Tentu saja makin tinggi frekwensi pencuplikan, kualitas suara semakin tinggi. Sebagai contoh, suara yang dicuplic dengan frekwensi 11 KHz setara dengan suara telepon, sedangkan suara yang dicuplic dengan frekwensi 44 KHz setara dengan suara CD-Audio. Untuk kuantisasi, suara yang disimpan menggunakan 16 bit mempunyai kualitas lebih tinggi dibandingkan suara yang disimpan menggunakan 8 bit saja.

Kelemahan file .WAV yang paling mencolok adalah ukurannya sangat besar. Sebuah lagu yang bila diputar hanya membutuhkan waktu beberapa menit saja, dapat memakan ruangan harddisk puluhan Megabyte.

Dalam memainkan file Wave dengan visual b asic, anda dapat menggunakan kontrol program yang sudah disediakan oleh visual basic atau software multimedia lain dan dapat juga dimainkan dengan memanfaatkan fasilitas library ”Winmm.dll”.

Contoh :

- Desain form seperti berikut :



- Ketik kode program berikut pada general procedure

```
Private Declare Function playsound Lib _
"winmm.dll|" Alias "playsounda" ( _
ByVal lpszname As String, ByVal hmodule _
As Long, ByVal dwflags As Long) As Long
Const snd_async = &H1
Const snd_filename = &H20000
Const snd_sync = &H0
Const snd_nodefault = &H2
Const snd_loop = &H8
Const snd_nonstop = &H10
Const snd_nowait = &H2000

Private Function putarlagu(soundfile As String) As Boolean
Dim pilihlagu As Boolean
pilihlagu = playsound(soundfile, vbNull, sdn_filename + snd_sync
+ snd_nonsto + snd_nodefault)
putarlagu = pilihlagu
End Function
```

- Ketik kode program pada tombol memutar file WAVe

```
Private Sub Command1_Click()
putarlagu ("D:\data\lagu\ok.wav")
End Sub
```

b. File MIDI

sama seperti file .WAV, file MIDI (Musical Instruments Digital Interface) yang berakhiran, .MID juga berhubungan dengan suara. Perbedaan dasar dari file WAV dan file MIDI terletak pada teknik penyimpanannya. Keuntungan file .MID dibandingkan dengan file .WAV adalah ukurannya yang sangat kecil. Kerugiannya adalah tidak semua suara dapat disimpan dalam file .MID, hanya suara dari alat music tertentu saja yang dapat disimpan. Sebagai contoh, anda tidak dapat menyimpan suara anjing menggonggong atau singa mengaum pada file .MID.

Contoh Program :

- Buat desain form seperti berikut ini :

Ketikan kode program berikut :

```
Private Declare Function mcisendstring Lib _
"winmm.dll" Alias "mcisendstringa" ( _
ByVal lpstrcommand As String, _
ByVal lpstrreturnstring As String, _
ByVal ureturnlength As Long, _
ByVal hwndcallback As Long) As Long

Public Function playmidifile( _
midifile As String) As Boolean
Dim lret As Long
If Dir(midifile) = "" Then Exit Function
lret = mcisendstring("stop midi", "", 0, 0)
lret = mcisendstring("close midi", "", 0, 0)
lret = mcisendstring("open sequencer!" & _
midifile & "alias midi", "", 0, 0)
lret = mcisendstring("play midi", "", 0, 0)
playmidifile = (lret = 0)
End Function

Public Function stopmidi() As Boolean
Dim lret As Long
lret = mcisendstring("stop midi", "", 0, 0)
stopmidi = (lret = 0)
lret = mcisendstring("close midi", "", 0, 0)
End Function

Private Sub Command1_Click()
playmidifile ( _
"D:\musuc\coba.mid")
End Sub

Private Sub Command2_Click()
stopmidi
End Sub
```

C. Latihan

Buatlah program untuk menjalankan File Video degan esain seperti berikut ini :

BAB IX
PEMROGRAMAN DATABASE

A. Pendahuluan

Database merupakan bagian dari kehidupan kita sehari-hari meskipun sering tidak disadari. Sebagai contoh di STMIK Duta Bangsa, database digunakan untuk menyimpan data para mahasiswa, dosen, jadwal kuliah, nilai masing-masing mahasiswa, dan lain-lain.

Database merupakan sekumpulan data yang saling berhubungan, didesain untuk menyediakan informasi pada sebuah organisasi. Sedangkan DBMS (The Database Management System) merupakan perangkat lunak yang digunakan untuk mendefinisikan, membuat, mengatur, dan menyediakan akses pada database. Microsoft Access merupakan contoh software RDBMS (Relation DBMS).

B. Materi

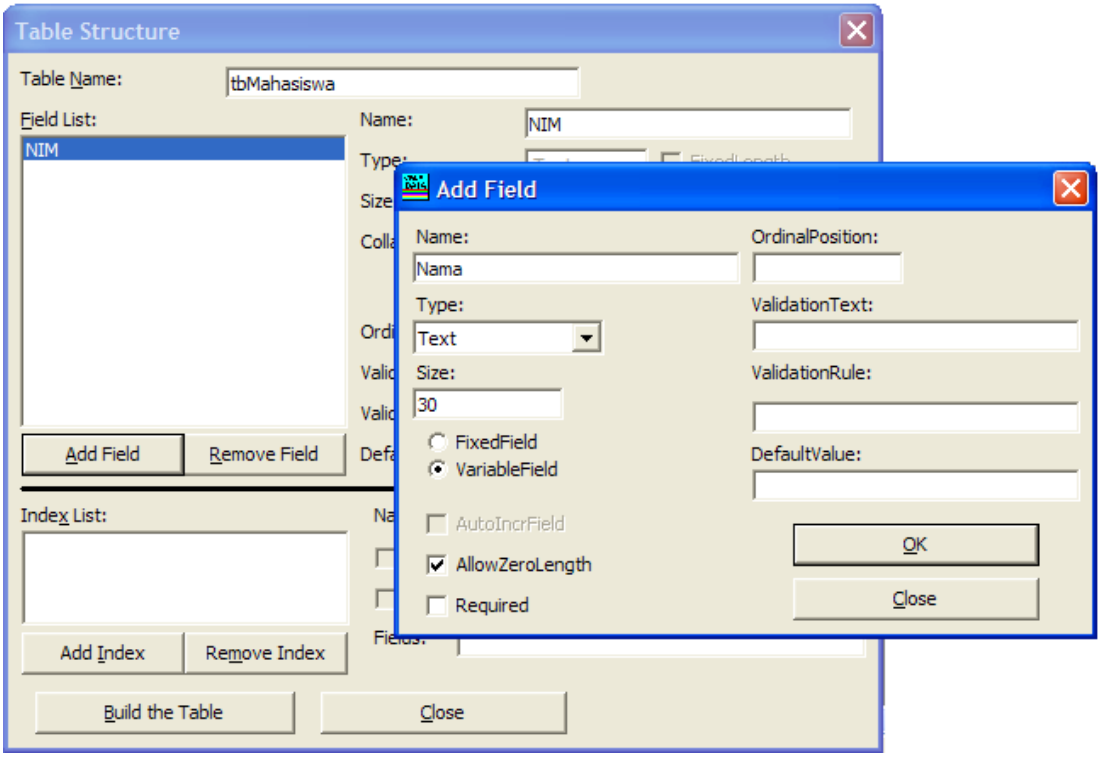
1. Membuat Database dan Tabel dengan Visual Data Manager

Untuk membuat database dengan Visual Data Manager, lakukan langkah-langkah sebagai berikut :

- Pilih menu Add Ins pada menu utama Visual Basic, kemudian pilih Visual Data Manager
- Pilih File – New – Microsoft Access – Version 7.0 MDB
- Berikan nama database(misal dbAkademik), tentukan tempat dimana Anda akan menyimpan database, kemudian klik Save.
- Untuk membuat tabel, klik kanan Properties, pilih New Table

Gambar 9.1 Membuat tabel

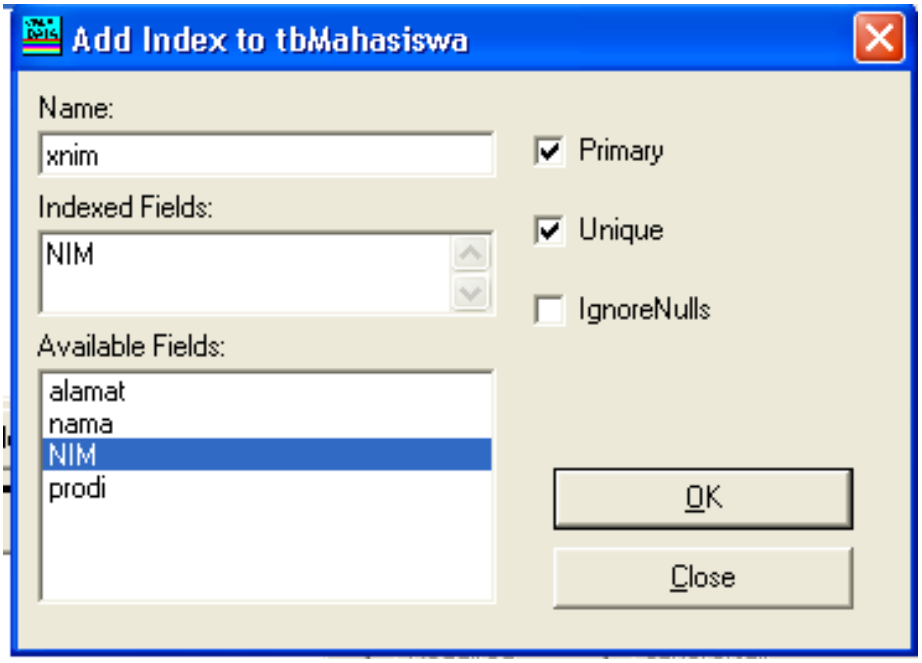
- Isikan nama tabel, klik Add Field, isikan nama field, tipe data dan tentukan Filed Sizenya, Klik OK.



Gambar 9.2 Struktur Tabel

- Untuk membuat Primary Key pada Field, klik Add Index, isikan Name (misal:NIM), pada Available Fields pilih Field yang akan dijadikan Primary Key (misal pilih NIM)
- Tentukan pilihan Index (Primary, Unique, IgnoreNulls). Primary digunakan sebagai pengenal suatu record, jika tabel yang dihubungkan berhubungan

denga tabel yang lain (dalam satu tabel hanya boleh ada satu primary key). Unique dipilih jika ingin indeks bersifat unik, IgnoreNulls digunakan untuk mengabaikan field yang kosong.

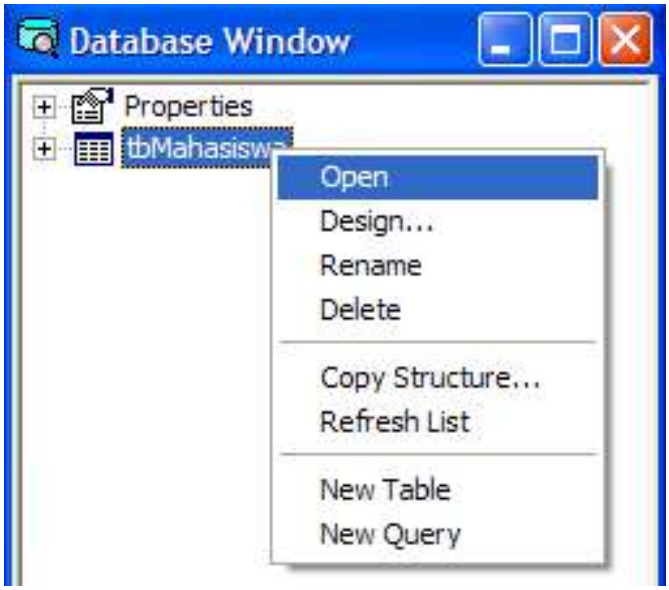


Gambar 9.3 Menentukan Primary Key

- Jika Anda sudah membuat semua Field yang dibutuhkan, maka klik Build the Table

2. Mengisi Data Pada Tabel

- Klik kanan pada nama tabel (tbMahasiswa, pilih Open



Gambar 9.4 Membuka Tabel

➢ Muncul window Dynaset:Mahasiswa

Gambar 9.5 Mengisikan data pada tabel

Fungsi menu yang ada pada tabel tbMahasiswa

- Add : menambah record baru
- Edit : mengoreksi record
- Delete : menghapus record aktif
- Find : mencari record
- Refresh : menampilkan ulang setelah record mengalami perubahan
- Close : menutup tabel

3. Data Acces Object (DAO)

DAO (Data Access Object) merupakan suatu Object Data Access Yang berukuran besar meliputi kemampuan Data Definition, Data Manipulation dan Database Maintenance. Untuk mengakses database menggunakan DAO dibutuhkan object Data. Jika dikoneksikan dengan Microsoft Access DAO hanya bisa dikoneksikan dengan Access 97.

Gambar 9.6 Objek Data

Beberapa properti yang dimiliki oleh objek Data dapat Anda lihat pada tabel 9.1

Tabel 9.1 Properti Objek Data

Properti	Keterangan
Connect	Hubungan file data, bias bersumber dari Access, dBase, Excel, Foxpro, Lotus, Paradox, dan file Text.
DatabaseName	Nama file database
RecordSource	Tabel dalam database
EOF Action	Pengecekan pergerakan kusror. Jika kursor sampai pada akhir file, maka terdapat pilihan property : MoveLast (record terakhir), EOF (akhir file), AddNew (menambah record baru)
Recordset Type	Tipe record : 0-Table, 1-Dynaset, 2-SnapShot

a) Koneksi database dengan DAO

Untuk koneksi database dengan DAO, Anda harus menambahkan objek Data pada Form. Ada dua cara koneksi database menggunakan DAO

1. Dengan mengatur property dari objek Data

Tabel 9.2 Properti Kontrol Data dbAkademik

Property	Setting Value
Name	datAkademik
Caption	dbAkademik
Connect	Access
DatabaseName	dbAkademik
Recordsource	tbMahasiswa

2. Pengaturan dengan mengetikkan kode program seperti berikut :

```
Private Sub Form_Load()
datAkademik.DatabaseName = App.Path & "\dbAkademik.mdb"
datAkademik.RecordSource = "tbMahasiswa"
End Sub
```

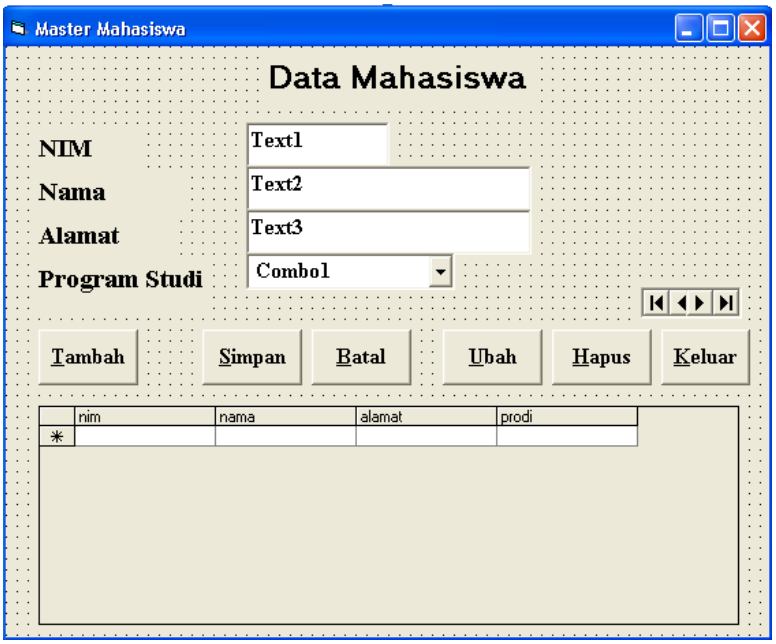
Koneksi database cara pertama masih bersifat statis yang artinya jika program/file dipindah ke folder atau drive lain, maka pengaturan properti DatabaseName harus disesuaikan dengan program/file berada.

Koneksi database cara kedua, sifatnya dinamis. **App.Path** digunakan untuk mendapatkan folder aktif.

b) Contoh Program menggunakan DAO

Berikut adalah contoh program olah data Mahasiswa di STMIK Duta Bangsa. Atur Form seperti pada contoh gambar 9.7.

- Grid menggunakan DBGrid. Untuk menambahkan DBGrid di toolbox adalah dari menu Project – Components, pilih tab Controls- kemudian pilih Microsoft Data Bound Grid Control 5.0. Atur properti Datasource dari DBGrid, pilih datAkademik.



Gambar 9.7 Program olah data mahasiswa dengan DAO

Ketik kode program berikut :

1) Koneksi Database

```
Private Sub Form_Load()
Data1.DatabaseName = App.Path & "\Akademik.mdb"
Data1.RecordSource = "Mahasiswa"
End Sub
```

2) Form Activate

```
Private Sub Form_Activate()
kosong
tdksiap
batal.Enabled = False
simpan.Enabled = False
tambah.SetFocus
End Sub
```

3) Tombol Edit

```
Private Sub cmdEdit_Click()
If cmdEdit.Caption = "Edit" Then
    txtNIM.Enabled = True
    txtNIM.SetFocus
    cmdEdit.Caption = "Update"
    cmdNew.Enabled = False
    cmdDel.Enabled = False
Else
    With datAkademik.Recordset
        .Edit 'Perintah untuk edit data
        !Nama = txtNama.Text
        !Alamat = txtAlamat.Text
        !TglLahir = dtpLahir.Value
        .Update
    End With
    cmdCancel_Click ' memanggil cmdCancel_Click
End If
End Sub
```

4) Sub Rutin

```
Sub kosong()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Combo1.Text = ""
End Sub
Sub siap()
Text1.Enabled = True
Text2.Enabled = True
Text3.Enabled = True
Combo1.Enabled = True
End Sub
Sub tdksiap()
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Combo1.Enabled = False
End Sub
```

5) Tombol Tambah

```
Private Sub tambah_Click()
tombol = "TAMBAH"
siap
kosong
Text1.SetFocus
batal.Enabled = True
simpan.Enabled = True
tambah.Enabled = False
keluar.Enabled = False
ubah.Enabled = False
hapus.Enabled = False
End Sub
```

6) Tombol Simpan

```
Private Sub simpan_Click()
If tombol = "UBAH" Then
Data1.Recordset.Edit
Data1.Recordset!NIM = Text1.Text
Data1.Recordset!nama = Text2.Text
Data1.Recordset!alamat = Text3.Text
Data1.Recordset!prodi = Combo1.Text
Data1.Recordset.Update
Data1.Refresh
End If

If tombol = "TAMBAH" Then
Data1.Recordset.AddNew
Data1.Recordset!NIM = Text1.Text
Data1.Recordset!nama = Text2.Text
Data1.Recordset!alamat = Text3.Text
Data1.Recordset!prodi = Combo1.Text
Data1.Recordset.Update
Data1.Refresh
End If

kosong
tdksiap
simpan.Enabled = False
batal.Enabled = False
tambah.Enabled = True
keluar.Enabled = True
ubah.Enabled = True
hapus.Enabled = True
End Sub
```

7) Tombol Batal

```
Private Sub batal_Click()
kosong
tdksiap
batal.Enabled = False
simpan.Enabled = False
keluar.Enabled = True
tambah.Enabled = True
ubah.Enabled = True
hapus.Enabled = True
tambah.SetFocus
End Sub
```

8) Tombol Ubah

```
Private Sub ubah_Click()
tombol = "UBAH"
siap
Text1.SetFocus
batal.Enabled = True
simpan.Enabled = True
tambah.Enabled = False
keluar.Enabled = False
ubah.Enabled = False
hapus.Enabled = False
End Sub
```

9) Tombol Hapus

```
Private Sub hapus_Click()
jawab = MsgBox("Yakin Data tersebut akan dihapus
...?", vbYesNo + vbQuestion, "Konfirmasi")
If jawab = 6 Then
    Data1.Recordset.Delete
       Data1.Refresh
End If

End Sub
```

10) Tombol Keluar

```
Private Sub keluar_Click()
Unload Me
End Sub
```

11) DbGrid1 RowColChange

```
Private Sub DBGrid1_RowColChange(LastRow As Variant,
ByVal LastCol As Integer)
Text1.Text = Data1.Recordset!NIM
Text2.Text = Data1.Recordset!nama
Text3.Text = Data1.Recordset!alamat
Combo1.Text = Data1.Recordset!prodi
End Sub
```

12) General

```
Dim tombol As String
```

ADO (ActiveX Data Object) merupakan suatu Object Data yang mempunyai kemampuan Data Definition, Data Manipulation dan Database Maintenance serta dapat untuk membangun koneksi dengan beberapa jenis database. Untuk mengakses database menggunakan ADO ada dua fasilitas yang dapat dipakai yaitu

- ADODC

  ADODC adalah sebuah object sehingga dapat dilihat atau ditambahkan di toolbox. Untuk menambahkan objek ini maka lakukan langkah sebagai berikut :

  Klik menu Project, pilih Component (atau tekan Ctrl-T), tandai dengan memberi tanda cek pada Microsoft ADO Data Control 6.0 (OLEDB). Akhiri dengan tekan tombol OK, maka pada Toolbox akan muncul objek ADODC.

- ADODB

  Salah satu cara menghubungkan aplikasi dengan database melalui kode program, tanpa menggunakan objek.

Database yang dapat dikoneksi dengan ADO selain Microsoft Access, dapat juga dengan Foxpro, ODBC, SQLServer atau MySQL.

4. Koneksi database dengan ADODB

Di dalam Visual Basic, Object ADODB yang sering digunakan adalah ADODB.Connection dan ADODB.Recordset. Untuk dapat membuat suatu variable dari ADODB ini kita harus terlebih dahulu memilih references di dalam Visual Baic yaitu Microsoft ActiveX Data Object. Ikuti langkah berikut :

- Pilih menu Project – References
- Pilih Microsoft ActiveX Data Objects 2.1 Library – Klik OK

Module sering digunakan di dalam pendeklarasian Connection dan Recordset untuk penggunaan database , sehingga Object Connection dan Recordset tersebut dapat digunakan pada semua bagian di dalam project yang anda buat.

- **Connection** digunakan untuk melakukan koneksi ke database yang dipilih dengan perintah open ataupun mengeksekusi sintaks-sintaks SQL dengan perintah execute. Untuk membuka koneksi ke database gunakan perintah open yang diikuti oleh connection string yang dapat dihapalkan ataupun menggunakan bantuan seperti control ADODC atau file extension .UDL
- **Recordset** digunakan untuk menampung data (bisa berasal dari 1/ lebih tabel) yang merupakan hasil eksekusi perintah sql select, syntax:

  select *[nama kolom]* ***from*** *[nama tabel]* ***where*** *[kondisi]*

Tambahkan Module pada program VB Anda (Project – Add Module), kemudian ketikkan kode program koneksi database berikut di Module Anda

```
Public Con As ADODB.Connection
Public Sub OpenConnection()
    Set Con = New ADODB.Connection
    Con.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
           "Data Source=" & App.Path & "\dbAkademik.mdb;" & _
           "Persist Security Info=False"
    Con.CursorLocation = adUseClient
    Con.Open
End Sub
```

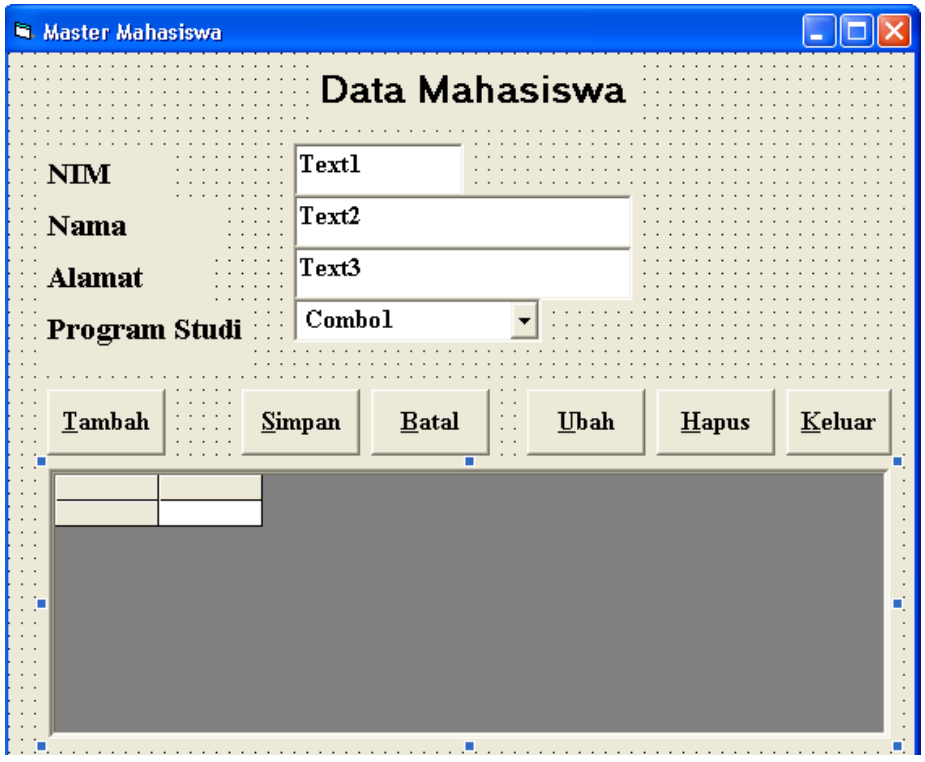
Hasil dari koneksi di atas adalah Con akan membuka koneksi ke database dengan nama dbAkademik.mdb yang terletak di folder aktif dbAkademik.mdb.

Contoh Program menggunakan ADODB

Berikut adalah contoh program olah data Mahasiswa di STMIK Duta Bangsa. Atur Form seperti pada contoh gambar 9.8.

- NIM akan diinput secara otomatis berdasarkan Program Studi, Jenjang, Tahun Masuk dan urutan mahasiswa saat daftar ulang.
- Jenjang Sarjana menawarkan dua program studi yaitu : Sistem Informasi dan teknik Informatika. Jenjang Diploma 3 menawarkan dua program studi yaitu : Manajemen Informatika dan Teknik Komputer.
- Kita akan menggunakan MSFlexGrid untuk menmpilkan data. Cara menambahkan project MSFlexGrid : Pilih menu Project – Components, Pilih Microsoft FlexGrid Control 6.0 – Klik OK

Berikut tampilan Form yang digunakan :



Gambar 9.8 Program olah data mahasiswa dengan ADODB

Ketik kode program berikut :

1. Koneksi Database

   Ketik kode program koneksi database berikut pada Module

```
Public Con As ADODB.Connection
Public rsMahasiswa As ADODB.Recordset
Public Sub OpenConnection()
    Set Con = New ADODB.Connection
    Con.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
           "Data Source=" & App.Path & "\dbAkademik.mdb;" & _
           "Persist Security Info=False"
    Con.CursorLocation = adUseClient
    Con.Open
End Sub
```

2. Kode program di Form_Load

```
Private Sub Form_Load()
OpenConnection 'Memanggil koneksi database dari Module

'Mengkoneksikan rsMahasiswa ke tabel tabel Mahasiswa
Set rsMahasiswa = New ADODB.Recordset
rsMahasiswa.Open "Select * From Mahasiswa", Con, adOpenStatic,
adLockOptimistic

'Memanggil sub TampilData
'Digunakan untuk menampilkan data di Grid
TampilData

End Sub
```

3. Kode program untuk mengatur Grid

```
Sub AturGrid()
    Grid.Rows = 1
    Grid.Cols = 5
    Grid.ColWidth(0) = 500
    Grid.ColWidth(1) = 1000
    Grid.ColWidth(2) = 2000
    Grid.ColWidth(3) = 2000
    Grid.ColWidth(4) = 2000

    Grid.TextMatrix(0, 0) = "No"
    Grid.TextMatrix(0, 1) = "NIM"
    Grid.TextMatrix(0, 2) = "Nama"
    Grid.TextMatrix(0, 3) = "Alamat"
    Grid.TextMatrix(0, 4) = "Program Studi"

End Sub
```

4. Kode program untuk menampilkan data di Grid

```
Sub TulisData()
Dim i As Integer
    i = 0
    Do While Not rsMahasiswa.EOF
        i = i + 1
        Grid.AddItem (i & vbTab & rsMahasiswa(0) & vbTab & _
        rsMahasiswa(1) & vbTab & rsMahasiswa(2) & vbTab & _
        rsMahasiswa(3))
        rsMahasiswa.MoveNext
    Loop
End Sub

Sub TampilData()
Set rsMahasiswa = New ADODB.Recordset
rsMahasiswa.Open "Select * From Mahasiswa " & _
"Order By NIM", Con, adOpenStatic, adLockOptimistic
AturGrid
TulisData
End Sub
```

5. Kode program Sub Rutin

```
Sub kosong()
      Text1.Text = ""
      Text2.Text = ""
      Text3.Text = ""
      Combo1.Text = ""
End Sub

Sub siap()
      Text1.Enabled = True
      Text2.Enabled = True
      Text3.Enabled = True
      Combo1.Enabled = True
End Sub

Sub tdksiap()
      Text1.Enabled = False
      Text2.Enabled = False
      Text3.Enabled = False
      Combo1.Enabled = False
End Sub
```

6. Kode program Form Activate

```
Private Sub Form_Activate()
      kosong
      tdksiap
      batal.Enabled = False
      simpan.Enabled = False
      tambah.SetFocus
End Sub
```

7. Kode program Tombol Tambah

```
Private Sub tambah_Click()
      tombol = "TAMBAH"
      siap
      kosong
      Text1.SetFocus
      batal.Enabled = True
      simpan.Enabled = True
      tambah.Enabled = False
      keluar.Enabled = False
      ubah.Enabled = False
      hapus.Enabled = False
End Sub
```

8. Kode program Tombol Simpan

```
Private Sub simpan_Click()
If tombol = "UBAH" Then
Set rsMahasiswa = New ADODB.Recordset
    rsMahasiswa.Open "Select * From Mahasiswa where NIM =
'" & Text1.Text & "'", Con, adOpenStatic, adLockOptimistic
    With rsMahasiswa
       !NIM = Text1.Text
        !prodi = Combo1.Text
        !nama = Text2.Text
        !alamat = Text3.Text
        .Update
    End With

End If

If tombol = "TAMBAH" Then
Set rsMahasiswa = New ADODB.Recordset
    rsMahasiswa.Open "Select * From Mahasiswa", Con,
adOpenStatic, adLockOptimistic
    With rsMahasiswa
        .AddNew 'Menambahkan data baru
        !NIM = Text1.Text
        !prodi = Combo1.Text
        !nama = Text2.Text
        !alamat = Text3.Text
        .Update
    End With

End If

kosong
tdksiap
simpan.Enabled = False
batal.Enabled = False
tambah.Enabled = True
keluar.Enabled = True
ubah.Enabled = True
hapus.Enabled = True
TampilData
End Sub
```

9. Kode Program Tombol Batal

```
Private Sub batal_Click()
      kosong
      tdksiap
      batal.Enabled = False
      simpan.Enabled = False
      keluar.Enabled = True
      tambah.Enabled = True
      ubah.Enabled = True
      hapus.Enabled = True
      tambah.SetFocus
End Sub
```

10. Kode Program Tombol Ubah

```
Private Sub ubah_Click()
      tombol = "UBAH"
      siap
      Text1.SetFocus
End Sub
```

11. Kode Program Tombol Hapus

```
Private Sub hapus_Click()
      tombol = "HAPUS"
      siap
      Text1.SetFocus
End Sub
```

12. Kode Program Tombol Keluar

```
Private Sub keluar_Click()
      Unload Me
End Sub
```

13. Kode Program General Deklaration

```
Dim tombol As String
```

14. Kode Program Text1 Keypress

```
Private Sub Text1_KeyPress(KeyAscii As Integer)

Set rsMahasiswa = New ADODB.Recordset
rsMahasiswa.Open "Select * From Mahasiswa Where NIM = '" &
Text1.Text & "'", Con, adOpenStatic, adLockOptimistic

If KeyAscii = 13 Then  'Jika ditekan Enter
With rsMahasiswa
```

```
If tombol = "UBAH" Then
        If .EOF Then
         MsgBox "NIM " & Text1.Text & " belum ada"
         Text1.SetFocus
        Else
        Text2.SetFocus
         Text2.Text = !nama
         Text3.Text = !alamat
         Combo1.Text = !prodi
        End If

    ElseIf tombol = "HAPUS" Then
        If .EOF Then
         MsgBox "NIM " & Text1.Text & " belum ada"
         Text1.SetFocus
        Else
         Text2.Text = !nama
         Text3.Text = !alamat
         Combo1.Text = !prodi

         pesan = MsgBox("Apakah Anda yakin menghapus NIM " & _
                Text1.Text & " ?", vbYesNo, "Hapus Data")
        If pesan = vbYes Then
            Set rsMahasiswa = New ADODB.Recordset
            rsMahasiswa.Open "Select * From Mahasiswa where NIM = '"
& Text1.Text & "'", Con, adOpenStatic, adLockOptimistic
            rsMahasiswa.Delete
        End If

        End If
    End If
End With
TampilData
End If

'hanya boleh diisi angka atau backspace
If Not (KeyAscii >= Asc("0") And KeyAscii <= Asc("9") Or KeyAscii =
vbKeyBack) Then
Beep
KeyAscii = 0
End If

End Sub
```

C. Latihan

Buka kembali program yang sudah Anda buat seperti gambar 9.8.

- Tambahkan Tabel Dosen dan Tabel Mata Kuliah pada Database Akademik yang sudah Anda buat. Atur Field yang dibutuhkan untuk masing-masing tabel.
- Tambahkan Form untuk Olah Data Dosen dan Form untuk Olah Data Mata Kuliah dan ketik kode programnya.

BAB X
MENCETAK DATA

A. Pendahuluan

Mencetak data merupakan hasil akhir dari rangkaian proses pengolahan data untuk tujuan pelaporan atau dokumentasi. Misalnya proses cetak yang digunakan untuk pelaporan adalah cetak kwitansi pembayaran dari transaksi pembelian/penjualan yang telah dilakukan.

Microsoft Visual Basic 6.0 mempunyai beberapa cara dalam mencetak data, diataranya lewat kode program yang dirancang sendiri, data report atau memanfaatkan Report designer seperti cristal report dan sebagainya.

Mencetak lewat kode program memang lebih mudah disbanding lewat report designer tetapi kemampuan cetaknya kurang begitu bagus. Sedangkan cristal report sendiri sebenarnya adalah program terpisah dari Microsoft visual basic 6.0, tetapi karena komponen inimendukung kinerjanya maka banyak programmer yang suka membuat laporan dengan bantuan cristal report.

B. Materi

1. Mencetak Data dengan Kode Program

Sebelum membahas bagaimana cara mencetak data ke printer, terlebih dahulu akan dipaparkan bagaimana cara mencetak data ke form. Pada pembahasan sebelumnya sudah dipaparkan tentang mencetak data ke form, dimana proses cetaknya lebih ditekankan pada konsep perulangan (looping). Data yang dicetak dapat ditampung dalam suatu konstanta atau variabel.

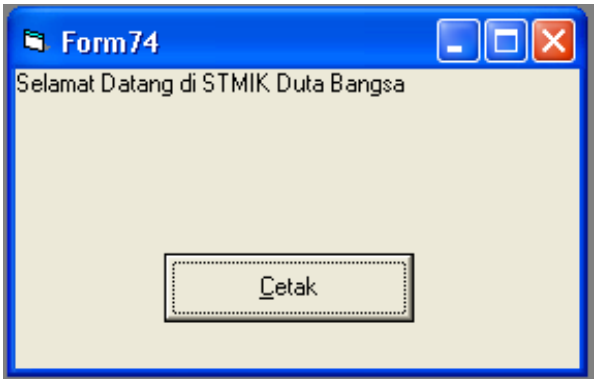
Perintah dasar untuk mencetak data dengan menggunakan kode-kode program adalah :

<objek>.Print <data yang akan dicetak>

Contoh :

a) Mencetak Data ke Form dengan Kode Program

```
DIM Ucapan As String
Ucapan = ”Selamat Datang di STMIK Duta Bangsa”
Print Ucapan
```



Gambar 10.1 Hasil Cetakan ke Form

b) Mencetak data ke Printer dengan Kode Program

```
DIM Ucapan As String
Ucapan = ”Selamat Datang di STMIK Duta Bangsa”
Printer.Print Ucapan
```

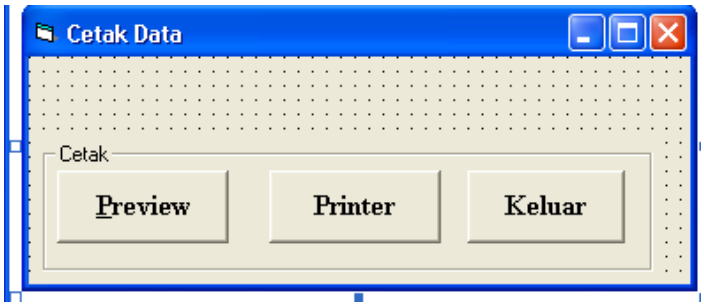
Terdapat beberapa perintah cetak yang sering digunakan pada proses cetak ke printer dari kode-kode program, seperti dijelaskan pada tabel berikut ini :

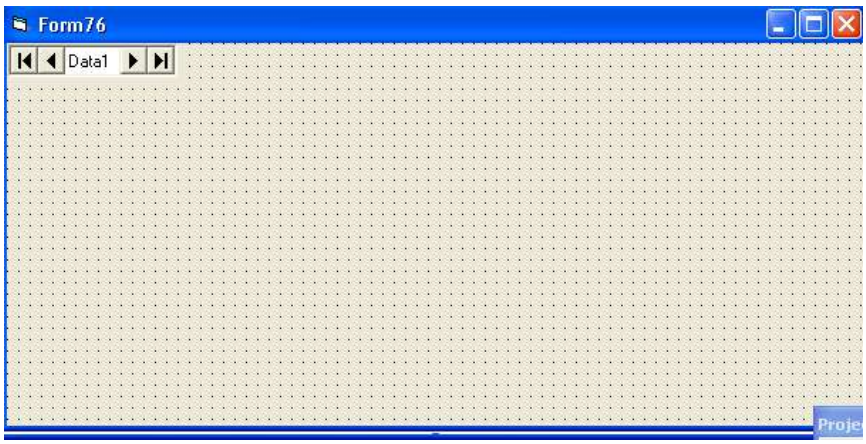
Tabel 10.1 Perintah Mencetak Data

Perintah	Keterangan
Printer.CurrentX = 0 Printer.CurretY = 0	Perintah ini digunakan untuk mengatur posisi head printer pada awal kertas yaitu pojok kiri atas kertas.
Printer.EndDoc	Data-data yang sedang dicetak akan masuk ke dalam Print Spooler yaitu daftar antrian cetak dari printer yang aktif. Sedang untuk melaksanakan pencetakan secara nyata harus ditambahkan perintah Printer.EndDoc
Printer.NewPage	Perintah ini menyababkan head printer berpindah ke awal halaman berikutnya.
Printer.KillDoc	Menghentikan proses cetak secara mendadak (biasanya diikuti oleh kondisi tertentu)
Printer.Orientation = <number>	Menentukan bentuk orientasi cetakan data untuk cetakan portrait (tegak) atau Landscape (mendatar) Nilai 1 Portrait dan 2 Landscape

c) Contoh Program

Desain Form seperti berikut :





Kode program untuk form Cetak

```
Private Sub keluar_Click()
End
End Sub

Private Sub preview_Click()
Form76.Show
End Sub

Private Sub printer_Click()
Dim xno  As String
printer.CurrentX = 0
printer.CurrentY = 0
no = 1
printer.FontSize = 14
printer.FontBold = True
printer.Print Tab(15); "Laporan Data Mahasiswa"
printer.FontSize = 10
printer.Print "-----------------------------------------" & _
 "-------------------------------------------------"
printer.Print Tab(1); "No. ";
printer.Print Tab(5); "NIM";
printer.Print Tab(15); "Nama Mahasiswa";
printer.Print Tab(40); "Alamat";
printer.Print Tab(58); "Program Studi";
Data1.Recordset.MoveFirst
Do While Not Data1.Recordset.EOF
With Data1.Recordset
printer.Print Tab(1); Str(no);
```

```
printer.Print Tab(5); .Fields(0);
printer.Print Tab(15); .Fields(1);
printer.Print Tab(40); .Fields(2);
printer.Print Tab(58); .Fields(3);
End With
no = no + 1
Data1.Recordset.MoveNext
Loop
End Sub
```

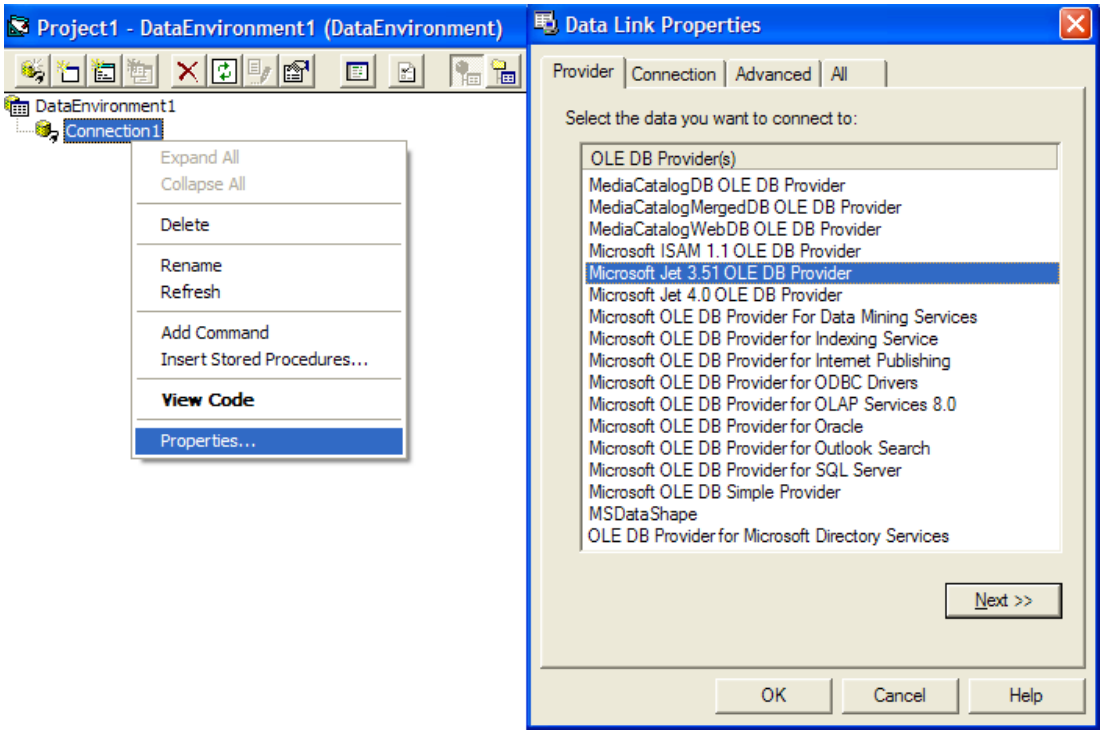
2. Mencetak Data dengan Data Report

Data Report adalah suatu laporan mengenai database yang disediakan oleh Visual Basic. Laporan ini dapat terbentuk jika Data Environment sudah didesain. ***Data Environment*** merupakan penghubung antara database yang telah didesain dengan Data Report yang akan ditampilkan.

a) Menggunakan Data Environment

Buka kembali program yang Anda buat seperti pada gambar 9.7 (Bab 9). Tambahkan Data Environment dengan cara sebagai berikut :

- Pastikan Data Environment dan Data Report sudah diaktifkan dengan cara : pilih Project – Components, pilih tab Designers, Aktifkan (centang) Data Environent dan Data Report- Klik OK.
- Pilih Project – Add Data Environment. Klik kanan Connection1-Pilih Properties. Pada bagian tab Provider pilih Microsoft Jet 3.51 OLE DB Provider (untuk Access 1997), Microsoft Jet 4.0 OLE DB Provider (untuk Access 2000-2003).

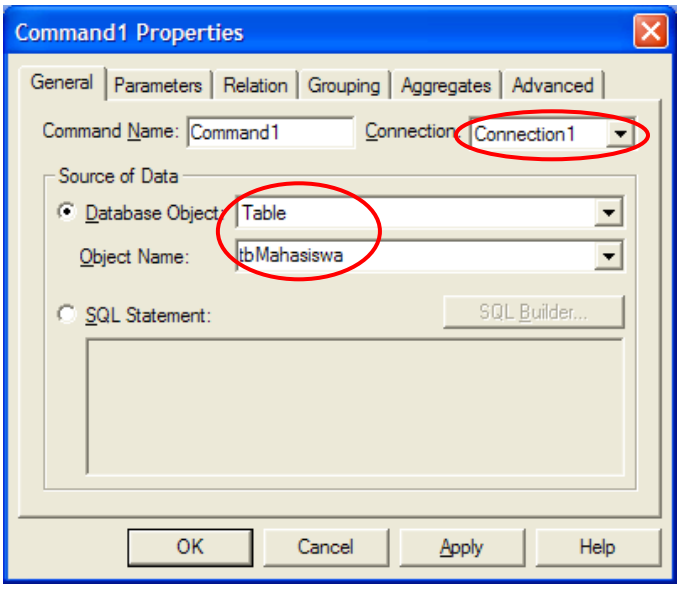


Gambar 10.2 Menggunakan Data Environment

- Klik Next>>, pada bagian *Select or enter a database name*, browse nama database (dbAkademik.mdb)
- Klik tombol Test Connection, jika sukses maka muncul messagebox seperti berikut :

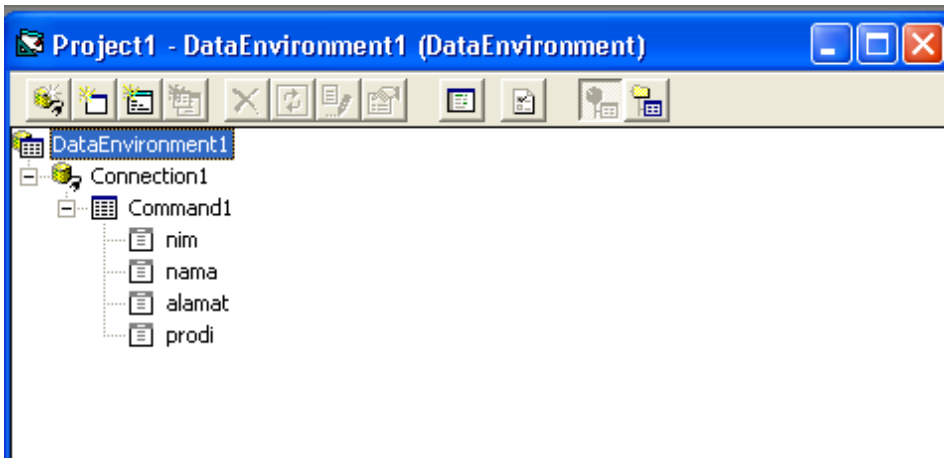
Gambar 10.3 MessageBox Koneksi database sukses

- Tambahkan Command untuk koneksi ke tabel, Klik kanan pada Connection1, pilih Add Command.
- Klik kanan pada Command1 pilih properties. Pada Connection pilih Connection1. Pada Database Object pilih Table. Pada Object Name pilih tbMahasiswa. Klik OK



Gambar 10.4 Properti Command1 untuk koneksi ke tabel

- Klik OK. Klik tanda + pada Command1, maka field-field dari tabel tbMahasiswa akan ditampilkan



Gambar 10.5 Field-field pada Command1 (tbMahasiswa)

b) Menggunakan Data Report

Setelah Data Environment kita atur, kemudian buat laporannya menggunakan Data Report. Ikuti langkah berikut :

- Pilih menu Project – Add Data Report

Tabel 10.1 Bagian Data Report

Bagian	Keterangan
Report Header	Berisi judul laporan
Page Header	Berisi judul kolom data yang akan ditampilkan
Detail	Berisi nama field yang akan ditampilkan pada kolom
Page Footer	Berisi catatan kaki di akhir kolom-kolom report
Report Footer	Berisi catatan kecil/keterangan report

Selain jendela Data Report, pada toolbox akan muncul sebuah tab Data Report yang berisi :

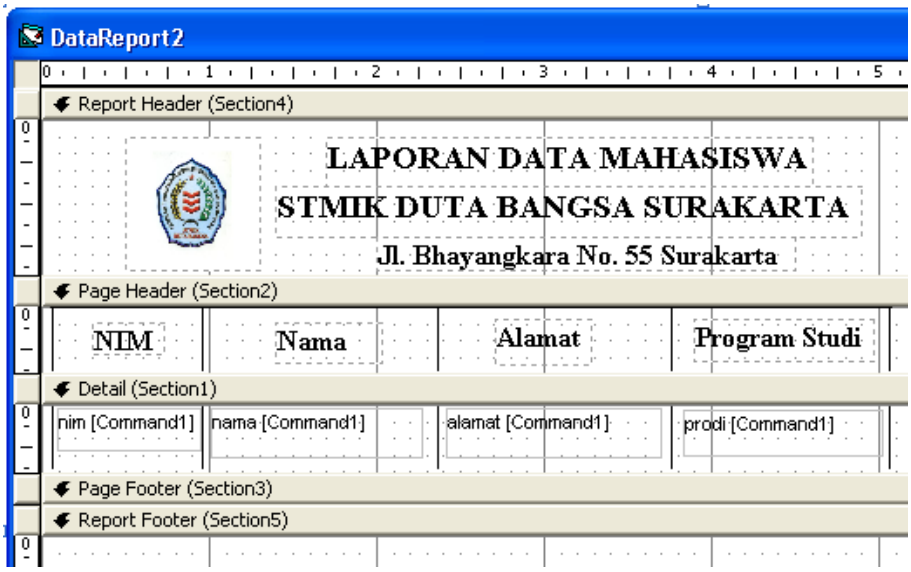
Gambar 10.6 Toolbox pada tab Data Report

Tabel 10.2 Toolbox pada tab Data Report

Nama Kontrol	Keterangan
RptLabel	Kontrol untuk memberikan label pada report
RptTextBox	Kontrol serupa textbox yang hanya menampilkan teks database saat runtime
RptImage	Kontrol untuk menempatkan image pada report
RptLine	Kontrol yang dapat menggambar garis secara horizontal, vertical maupun diagonal
RptShape	Kontrol untuk menggambar shape pada report
RptFunction	Kontrol yang dapat menset hasil perhitungan data

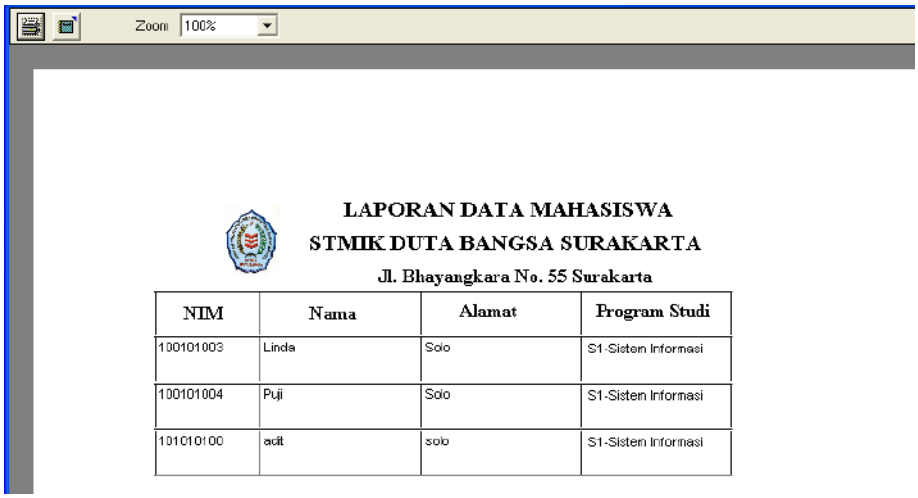
- Letakkan kontrol RptLabel pada bagian Report Header. Ubah Propery Caption menjadi ”Laporan Data ahasiswa”, dan propery Aligment menjadi 2-rptJustifyCenter
- Klik DataReport1, ubah property **DataSource**, pilih DataEnvironment1
- Isi **DataMember** denga nama Command1
- Klik menu bar **Window,** Pilih **Cascade**. Letakkan Jendela **Data Environment1** di atas jendela **Data Report**.
- Drag and Drop field-filed yang ada di Jendela DataEnvironent1 ke Jendela Data Report di bagian **Detail**.

- Pindahkan bagian yang diakhiri tanda ":" ke bagian **Page Header**



Gambar 10.7 Meletakkan Field-Field ke Data Report

- Desain akhir Data Report adalah sebagai berikut :



Gambar 10.8 Desain Akhir Data Report

3. Menctak data dengan Crystal Report

Selain Data Report, kita juga bisa menggunakan Crystal Report dalam pembuatan laporan. Crystal Report adalah suatu form khusus berbentuk seperti lembaran format naskah yang ingin dicetak.

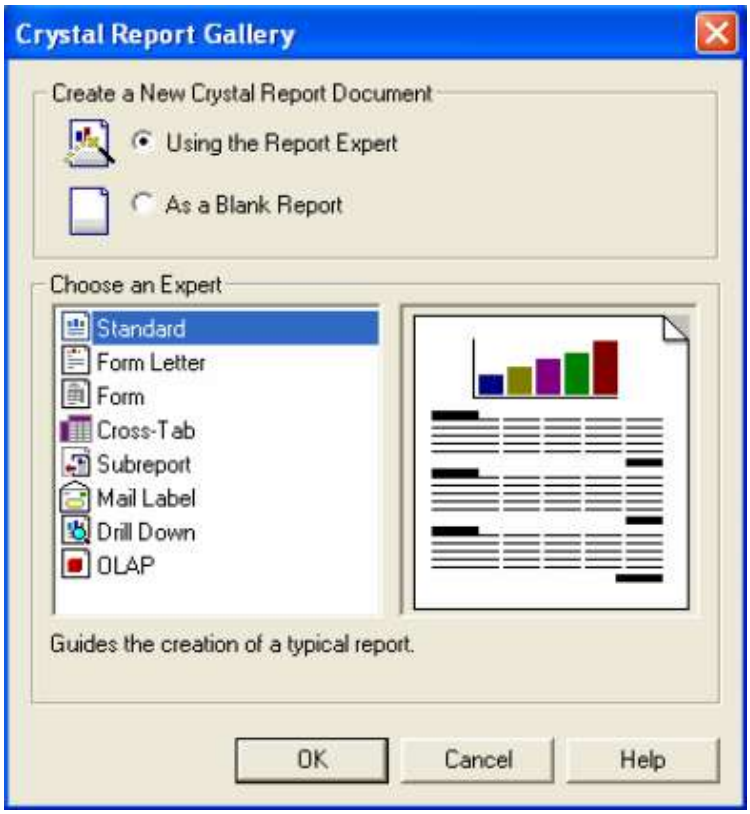
Crystal Report dapat berdiri sendiri dapat pula menjadi satu dengan project Visual

Basic yang anda buat. Bila berdiri sendiri, report tersebut pun dapat dipanggil dari

project Visual Basic dengan Crystal Report Control sehingga report yang telah anda buat dapat digunakan oleh beberapa project sekaligus.

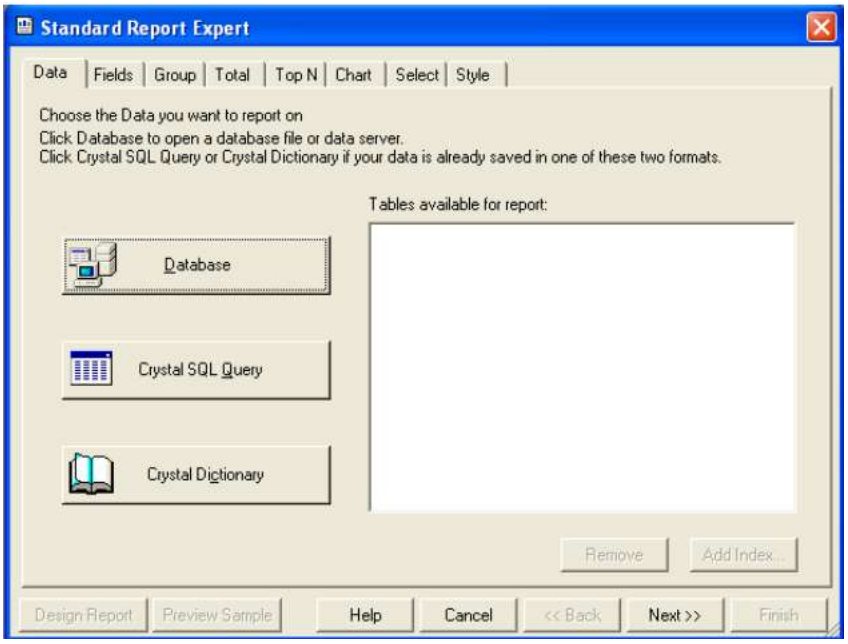
a) Membuat Laporan dengan Crystal Report

- Pertama-tama Anda harus Install dulu Software Crystal Report. Pada kesempatan kali ini menggunakan Crystal Report 8.5.
- Buka program Crystal Report , maka pada tampilan windows akan muncul tampilan seperti gambar di bawah ini



Gambar 10.9 Crystal Report Gallery

- Pilihlah Report Expert dan Standard Expert untuk laporan standard dan pilih OK untuk melanjutkan. Setelah itu akan muncul semua property dari report yang akan kita buat, yang akan ditampilkan seperti gambar :



Gambar 10.10 Standard Report Expert

- Pertama kali kita harus memasukkan datasource dari report yang akan kita buat (darimana data yang mau kita tampilkan) dengan memilih 1 dari ketiga jenis data yang telah tersedia (Database, Crystal SQL Query, dan Crystal Dictionary). Untuk menampilkan data dari database seperti Microsoft Access, kita akan memilih Database.
- Pilih DatabaseFile dan pilih find database file dan click add, cari database yang akan ditampilkan.
- Pilihlah datasource yang diinginkan dan click Add, setelah itu click Close, maka dengan ini, report yang akan kita buat telah ditentukan datasourcenya. Setelah itu pilihlah tombol Next untuk menuju bagian field yang akan kita tampilkan di dalam report.
- Pilihlah field-field yang akan kita tampilkan di report, dan bila sudah selesai, click finish untuk menuju design report kita.
- Buat desainya, simpan Report, misal rptMahasiswa

b) Koneksi Crystal Report dengan Visual Basic

- Pastikan Crystal Report sudah terinstall dan Anda sudah selesai membuat desain reportnya dan sudah Anda simpan dalam satu folder dengan program Visual Basic yang Anda buat.

- Buka kembali program yang telah Anda buat di Bab 9 (Gambar 9.8). Tambahkan tombol untuk cetak data.
- Tambahkan object Crystal Report di toolbox dengan cara pilih Project - Components – Pilih tab Controls - pilih Crystal Report Viewer Control tekan OK. Kemudian tambahkan objek tersebut ke dalam Form
- Berikut kode program untuk mencetak data di VB

```
'Mencetak semua data mahasiswa
Sub CetakAll ()
rptMhs.Reset
rptMhs.Connnect = Con
rptMhs.ReportFileName = App.Path & "\RptMahasiswa.rpt"
rptMhs.WindowState = crptMaximized
rptMhs.RetrieveDataFiles
rptMhs.Action = 1
End Sub

'Mencetak berdasarkan Program Studi
Sub CetakProdi()
rptMhs.Reset
rptMhs.Connnect = Con
rptMhs.ReportFileName = App.Path & "\RptDep.rpt"
rptMhs.SelectionFormula = ("{tbMahasiswa.Prodi}='" & cbProdi.Text & "'")
rptMhs.WindowState = crptMaximized
rptMhs.RetrieveDataFiles
rptMhs.Action = 1
End Sub
```

Beberapa property dari Crystal Report yang biasa digunakan di dalam aplikasi.

- .**Reset** - Digunakan untuk mengembalikan kondisi report ke dalam kondisi semula, biasanya properti ini digunakan pertama kali sebelum properti lainnya .
- **.Connect** - Digunakan untuk menenetukan jenis conection yang digunakan Oleh report tersebut, biasanya diisi oleh variabel ADODB.Connection yang kita buat.
- **.ReportFileName** - Digunakan untuk menentukan report yang akan kita tampilkan Beserta path dari report tersebut.
- **.StoreProcedureParam** - Digunakan bila datasource dari report yang akan kita panggil adalah storeProcedur yang memiliki parameter

- **.Formula** - Digunakan apabila di dalam report yang kita buat, terdapat Formula yang ingin kita isi dari VB.
- **.SelectionFormula** - Digunakan untuk memfilter data dari report yang ingin kita tampilkan dengan menyebutkan nama Datasource.Nama Field
- **.WindowState** -Digunakan untuk menentukan windowstate dari report pada saat pertama kali tampil apakah fullscreen, minimize dsb.
- **.Action** -Digunakan untuk memunculkan report yang akan kita tampilkan dengan memberinya angka 1 (.Action=1)

C. Latihan

Buka kembali program yang sudah Anda buat pada soal Latihan Bab 9. Buatlah Laporan untuk Data Dosen dan Data Mata Kuliah dengan Data Report dan Crystal Report.

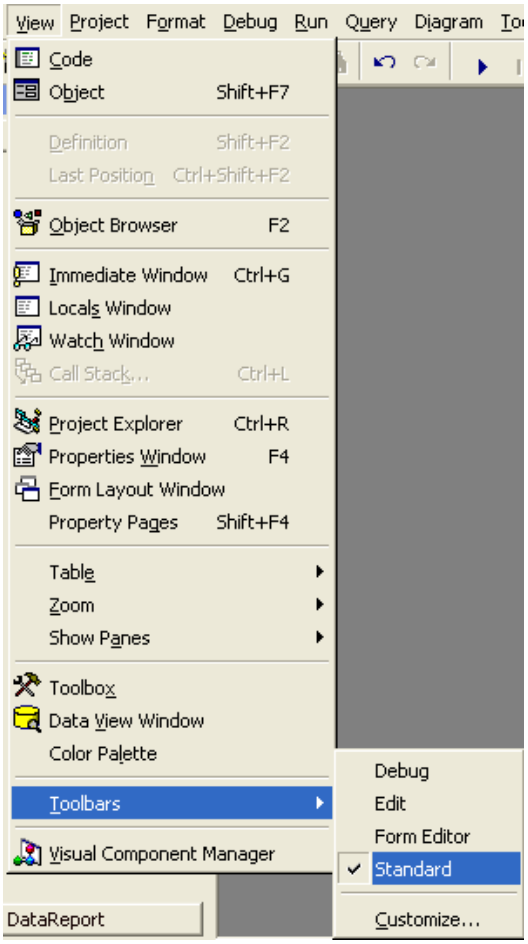
BAB XI
MERANCANG MENU PROGRAM

A. Pendahuluan

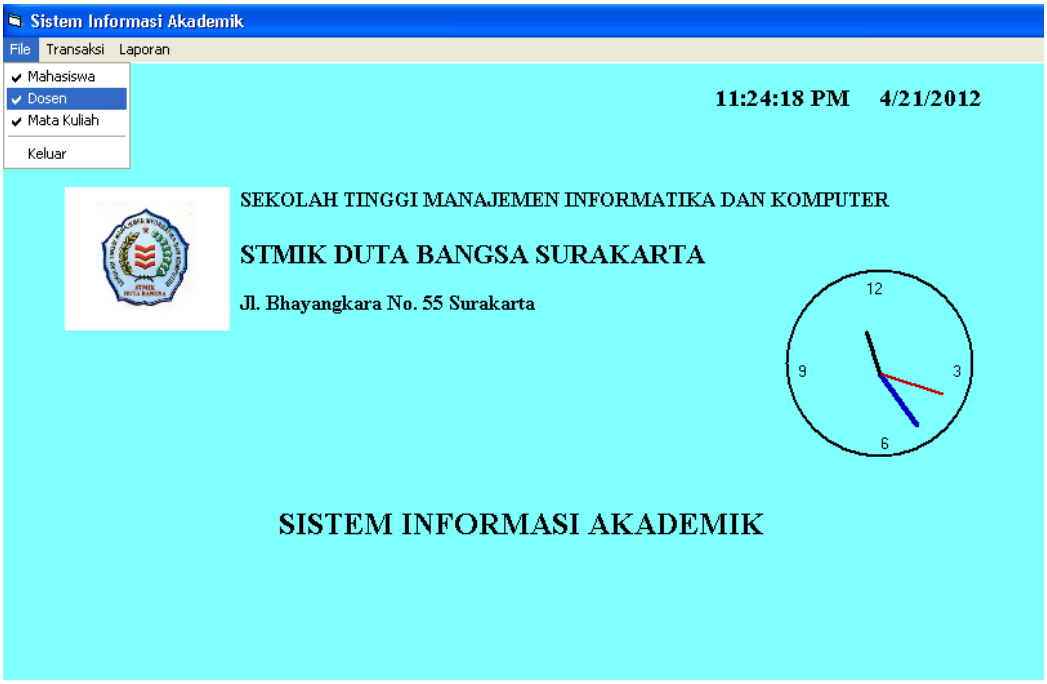
Menu adalah serangkaian pilihan yang dapat dipilih (di-klik atau di-enter) dengan tujuan untuk melakukan tugas-tugas tertentu. Umumnya suatu menu diitempatkan pada bagian atas dari programaplikasi. Tetapipada aplikasi tertentu (misalnya aplikasi pada website), pembuatan menusudah mengacu pada pemakaian gambar-gambar yang menarik. Umumnya gambar-gambar tersebut dibuat oleh software lain seperti photoshop, corel atau bahkan software animasi seperti flash.

Menu utama (menu bar) dalam suatu proyekmerupakan lokomotif utama dalam suatu program karena keberadaann menu bertujuan untuk mengintegrasikan program-program yang digunakan. Dengan demikian pemakai program tidak kesulitan dalam menguunakan program yang sedang dijalankan. Menu bar terletak di bawah title bar adalah tempat keberadaan menu, yang menampilkan Menu Title.

Jika suatu menu di klik (misalnya menu file), sebuah daftar yang berisi perintah-perintah (menu item) terbuka ke arah bawah. Menu item terdiri dari perintah-perintah (seperti New, Open sampai dengan exit), separator bar, dan sub menu title. Sedang masing-masing item mempunyai tugast ersendiri sesuai dengan kode program yang dipasangkan pada menu tersebut. Beberapa item menu menghasilkan proses secara langsung, misalnya menu exit akan langsung menutup suatu program.



Gambar 11.1 Elemen-elemen menu pada visual basic



Gambar 11.2 Contoh menu program aplikasi

B. Materi

1. MDI Form

Pada Visual Basic, anda dapat mengembangkan aplikasi dengan interface sebagai berikut :

- SDI (Single Document Interface)
- MDI (Multiple Document Interface)

Pada aplikasi SDI, setiap form merupakan form-form yang berdiri sendiri, Aplikasi SDI pada windows terdapat pada aplikasi seperti Notepad, WordPad dan Paint.

Sedangkan aplikasi seperti Visual Basic menggunakan MDI, yaitu terdiri dari suatu MDIForm, dan didalamnya merupakan form-form anak (MDIChild).

Ada beberapa hal yang harus diperhatikan dalam penggunaan MDIForm adalah :

- Didalam satu project hanya dapat terdiri dari satu MDIForm
- Anda tidak dapat menempatkan kontrol-kontrol secara langsung pada MDIForm, kecuali kontrol yang memiliki properti Alignment, atau menempatkannya diatas kontainer seperti Picturebox.
- Anda tidak dapat menggunakan metode penggambaran (Print, Line, Circle, dan PSet) seperti pada form umumnya.

a) Membuat MDI Form di Visual Basic

- Kita harus tambahkan dengan klik pada menu Project - Add MDI Form. Klik Open.
- Pada Project Explorer akan muncul dua buah form yaitu Form1 dan MDIForm1. Ubahlah properties MDIChild dari Form1 menjadi true. Hal ini menandakan Form1 adalah sebagai form anak dari MDIForm1. Settinglah pada Project Properties agar Form utamnya adalah MDIForm1.

b) Karakteristik dari MDI Form

- Semua Cchild Form tidak dapat dipindahkan keluar dari MDI Form.
- Ketika suatu Child Form diminimize, akan menjadi icon dibawah MDI Form.

- Anda dapat menentukan apakah Child Form secara otomatis ditampilkan atau tidak dengan menggunakan properti AutoShowChildren pada MDIForm.
- Jika pada Child Form ada menu, maka menu akan ditampilkan pada MDIform menu.

Pada MDIform anda tidak dapat menempatkan kontrol-kontrol visible yang tidak mendukung alignment, kecuali kalau anda menempatkannya kedalam suatu kontainer seperti Picturebox, dan Toolbar. Sedangkan kontrol non-visible seperti Timer dan CommonDialog box dapat ditempatkan diatas MDIForm.

c) Mendapatkan MDI Child yang sedang aktif.

Untuk mendapatkan form-form yang sedang aktif di dalam suatu MDIForm, anda dapat menggunakan properti ActiveForm, contoh :

```
Private Sub MDIForm_QueryUnload(Cancel As Integer, UnloadMode As
Integer)
If Not Me.ActiveForm Is Nothing Then
   MsgBox "Masih ada Form yang aktif"
   Cancel = True
End If
End Sub
```

d) Mengatur MDI Child dalam jendela MDI form

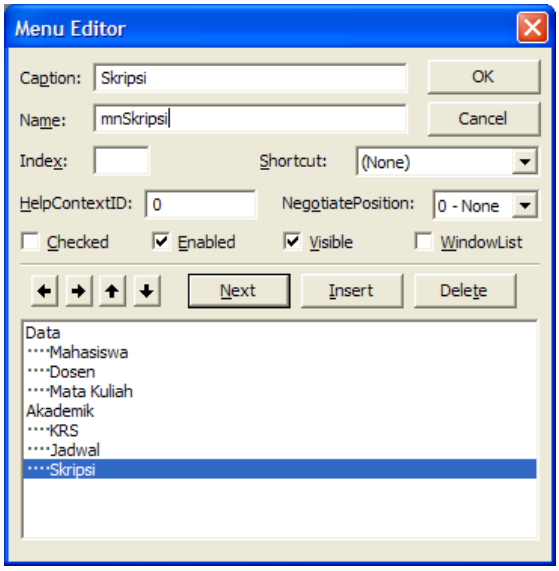
Anda dapat menggunakan metoda Arrange untuk mengatur penyusunan form-form yang sedang aktif didalam suatu MDI Form. Metoda Arrange ini diikuti oleh suatu parameter yang menentukan jenis penyusunan yang akan dilakukan, contoh :

```
Private Sub mnuTileHorizontally_Click()
    Arrange vbTileHorizontal
End Sub
Private Sub mnuTileVertically_Click()
    Arrange vbTileVertical
End Sub
Private Sub mnuCascade_Click()
    Arrange vbCascade
End Sub
Private Sub mnuArrangeIcons_Click()
    Arrange vbArrangeIcons
End Sub
```

2. Menu Editor

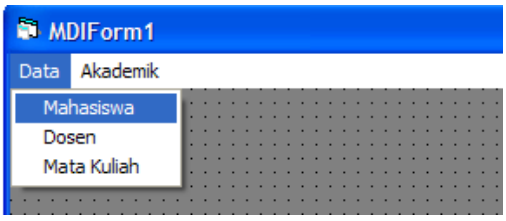
Pembuatan menu pada Visual Basic dapat dilakukan dengan bantuan Menu Editor yang terdapat pada Menu Tools-Menu Editor. Pada dasarnya setiap item menu memiliki Caption dan sebuah Nama. Anda dapat membentuk Kunci Akses dengan menggunakan tanda & (ampersand) pada Caption dari menu tersebut. Untuk membuat menu anda cukup mengetikan Caption dan Name, selanjutnya klik pada Next, dan ketikkan menu yang berikutnya, sampai selesai. Selanjutnya adalah menbuat Sub Menu dengan melakukan klik pada panah kanan dan sebaliknya.

a) Membuat Menu di MDI Form



Gambar 11.3 Window Menu Editor

Setelah menu diatur pada Window Menu Editor, maka berikut hasilnya :



Gambar 11.4 Tampilan menu yang dibuat di MDIForm

Untuk memunculkan frMahasiswa pada saat klik menu Mahasiswa maka ketikkan kode berikut pada jendela MDIForm kode editor.

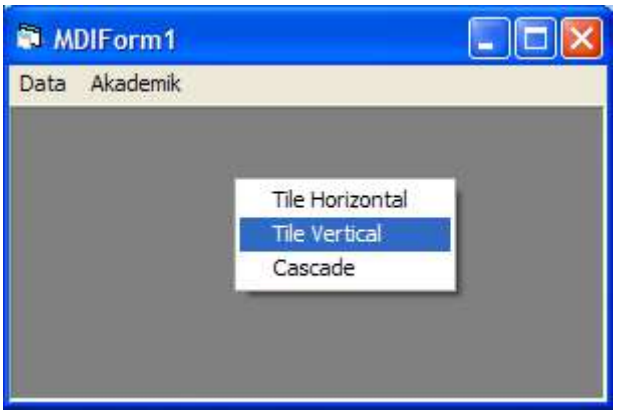
```
Private Sub mnMhs_Click()
frMahasiswa.Show
End Sub
```

b) Membuat Pop Up Menu

Pada Visual Basic, anda dapat membuat menu pop up dengan memanfaatkan menu Editor untuk mendefinisikan nama kelompok menu Pop Up beserta Sub Menunya, dan menonaktifkan option Visible dari

```
Private Sub MDIForm_MouseDown(Button As Integer, Shift As
Integer, X As Single, Y As Single)
If Button And vbRightButton Then
    PopupMenu mnAtur
End If
End Sub
```

Sehingga kalau dilakukan klik kanan pada form akan menanpilkan suatu PopUp menu yang berupa Sub Menu dari mnAtur.



Gambar 11.5 Tampilan Pop Up Menu di MDIForm

C. Latihan

Buka kembali program yang sudah Anda buat pada Bab 9. Koneksikan dengan Crystal Report (rptMahasiswa). Tambahkan MDIForm dan tambahkan menu Laporan untuk menampilkan laporan Mahasiswa, Dosen, Mata Kuliah dan laporan lain yang dibutuhkan.

BAB XII
FILE EXE DAN SETUP EXE

A. Pendahuluan

Suatu program aplikasi yang telah dibuat dengan visual basic 6.0 ingin didistribusikan ke pengguna yang lain. Prosesnya tidak semudah yang kita bayangkan, misalnya denganmengcompile visual basic project yang telah dibuat menjadi file .Exe. selanjutnya file .Exe ini dicopykan ke computer lain. Langkah ini jelas salah karena untuk bias berjalan di computer lain masih dibutuhkan file-file pendukung.

B. Materi

1. File Exe

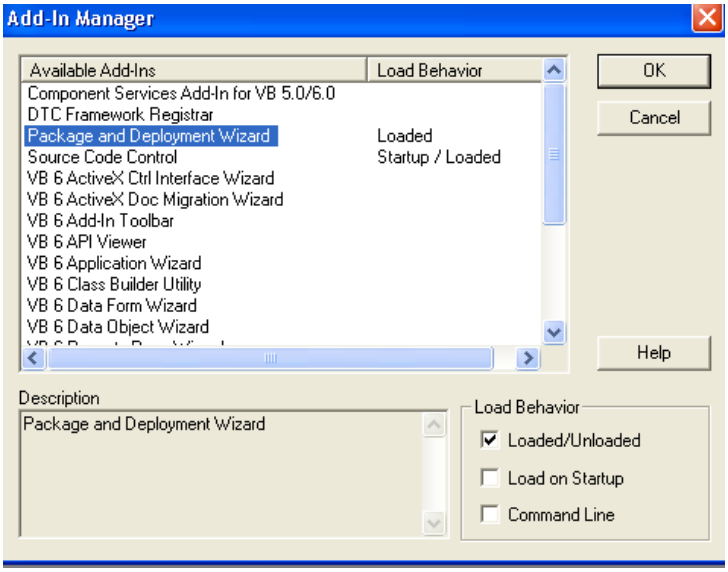
Setelah selesai membuat project anda dapat menyimpannya dan membuat file exe nya. Jika program aplikasi yang sudah dibuat belum dicompile dalam bentuk EXE maka akan dibutuhkan VB setiap kali menjalankannya. Agar user yang lain bisa juga menjalankan program yang telah kita buat, maka program yang telah selesai kita buat harus di compile dulu dalam bentuk exe.

- Untuk membuat File EXE buka kembali projek (*.VBP) yang telah selesai anda buat (misalnya projek mahasiswa yang telah kita buat)
- kemudian klik menu File- Make Project.exe lalu simpan file .exe tersebut dan lanjutkan mengklik OK
- Tunggu sampai proses kompilasi selesai dan coba keluar dari Visual Basic. Lalu jalankan file.exe yang telah anda buat.

2. Setup Exe

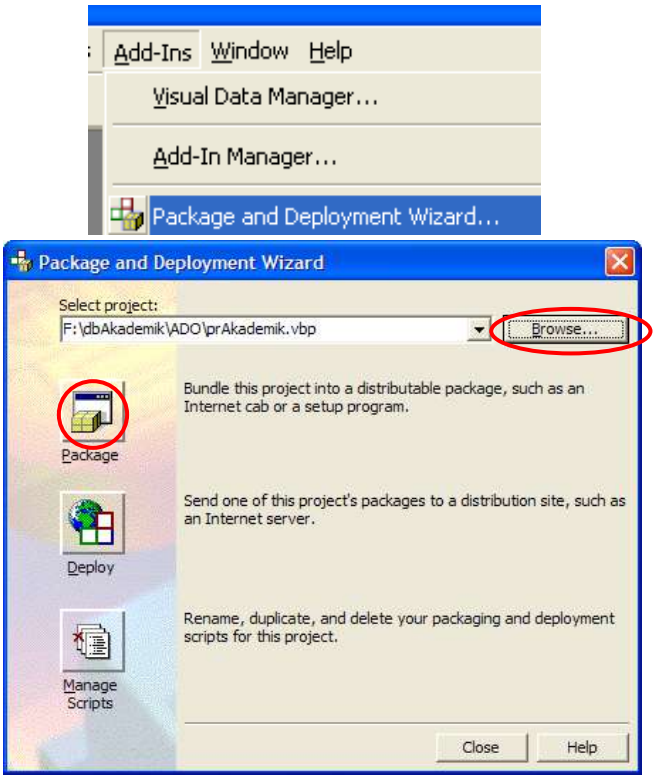
Selain file .exe yang lebih ngetrend lagi yaitu file setup.exe. Membuat file Setup.exe tidaklah sesulit yang dibayangkan hanya dengan beberapa klik anda sudah dapat membuat file Setup.exe.

- Untuk membuat setup.exe kita harus mengaktifkan terlebih dahulu Package & Deployment Wizard. Klik Menu Add Ins – Add Ins Manager, kemudian aktifkan Package and Deployment Wizard.



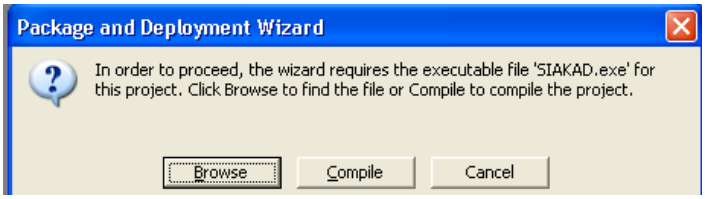
Gambar 12.1 Add-Ins Manager

- Untuk membuat Setup.exe menu Add-Ins - Package & Deployment Wizard | sehingga akan tampil window seperti gambar dibawah ini.



Gambar 12.2 Package and Deployent Wizard

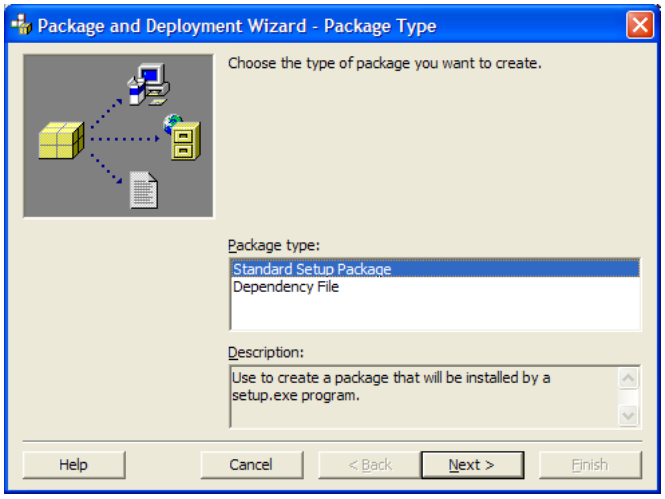
➢ Kemudian klik browse untuk mencari tempat folder projek yang telah kita buat dan simpan. Cari file .vbp yang telah anda buat, pastikan anda juga telah membuat file .exe dalam satu folder yang sama. Kemudian lanjutkan dengan mengklik Package



Gambar 12.3 Kotak Dialog Package and Deployment Wizard

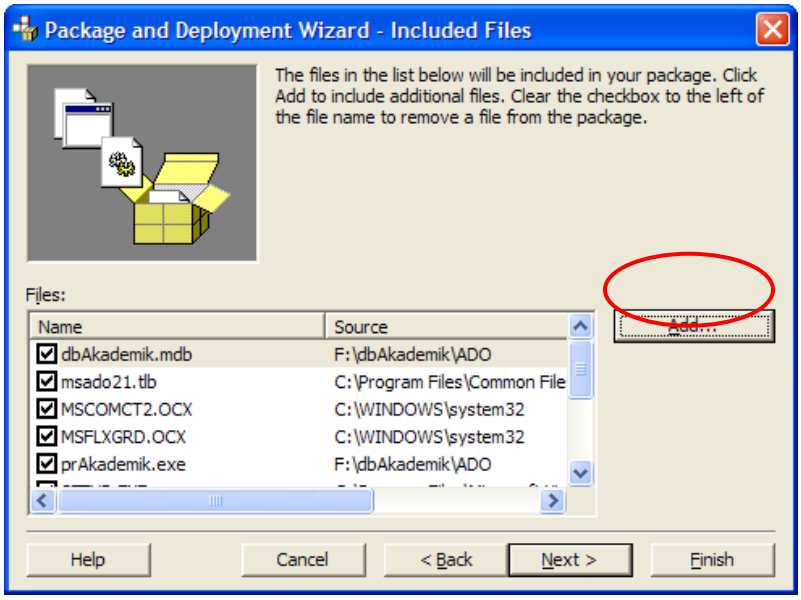
- Browse : digunakan untuk mencari file exe dari project yang sudah dibuat.
- Compile : digunakan untuk melakukan proses kompilasi program dari proyek yang pernah dibuat. Karena masih baru, klik tombol compile ini. Tetapi jika file proyeknya sudah pernah di compile, pilih tombol recompile untuk melakukan proses kompilasi ulang.
- Cancel : pilihan ini akan membatalkan seluruhproses yang sedang anda kerjakan.

➢ Setelah diklik Package akan muncul window seperti gambar 12.2



Gambar 12.4 Package and Deployent Wizard-Package Type

- Pilih Package type = **Standard Setup Package**. Lanjutkan dengan mengklik Next Setelah itu akan muncul window yang meminta anda untuk menyimpan hasil kompilasi Setup.exe. Pilih tempat folder yang anda inginkan Kemudian klik Next. Maka muncul Window yang menampilkan file-file yang akan dipackage. Jika Anda menggunakan database (misal : dbAkademik.mdb) dan file database tersebut belum ada di list, maka tekan tombol Add untuk menambahkan file database tersebut.



Gambar 12.5 Package and Deployent Wizard-Included Files

- Setelah anda klik Next akan muncul Window Cab Options, pilih pada defaultnya yaitu Single Cab, kemudian lanjutkan dengan klik Next.
- Setelah anda klik Next akan muncul Window Installation Title, masukkan sesuai dengan nama project yang kita buat. Kemudian lanjutkan dengan mengklik Next.
- Setelah anda klik Next akan muncul Window Start Menu Items, pilih pada defaultnya, lanjutkan dengan mengklik Next
- Setelah anda klik Next akan muncul Window Install Location yang memberitahu ke kita tempat / lokasi instalasi, biarkan pada defaultnya, lanjutkan dengan mengklik Next
- Setelah anda klik Next akan muncul Window Shared Files, lanjutkan dengan mengklik Next. Sehingga akan muncul window Finished, biarkan pada defaultnya, kemudian klik Finish.

- Tunggu beberapa saat sampai hasil kompilasinya selesai. Install hasil setup file yang sudah Anda buat. Jalankan dari start menu windows.

C. Latihan

Buatlah setup file dari program yang Anda buat. Install setup file tersebut, dan jalankan program yang sudah Anda instaal dari Start Menu

BAB XIII

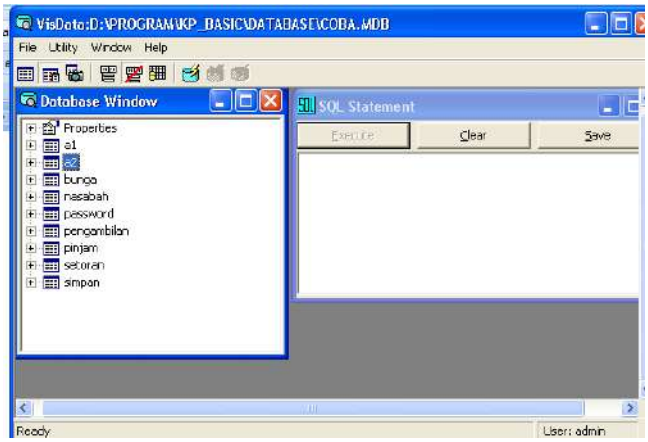
MEMBUAT SISTEM SIMPAN PINJAM KOPERASI

A. Pendahuluan

Sistem simpan pinjam koperasi, terdapat data master data bunga dan data nasabah, sedangkan proses transaksi terdapat transaksi simpan, pinjam, pengembalian dan setoran. Untuk laporan yang dibutuhkan laporan nasabah, laporan pinjam, laporan simpan, laporan pengembalian dan laporan setoran.

B. Materi

1. Buat database koperasi.mdb



2. Buat Tabel yang dibutuhkan dalam sistem simpan pinjam koperasi

   a) Buat Tabel Nasabah dengan struktur tabel seperti berikut :

Field Name	Type	Width	Index
No_rek	text	5	primary
nama	text	25	
J_kel	text	1	
Tempat_lhr	text	20	
Tgl_lahir	date	8	
agama	text	1	
alamat	text	30	
Saldo_spn	currency	8	
Saldo_pjm	currency	8	

b) Buat tabel pengambilan dengan struktur table seperti berikut :

Field Name	Type	Width	Index
No_trans	text	5	primary
no-rek	text	5	
Tgl_trans	date	8	
Jml_ambil	currency	8	

c) Buat tabel pinjam dengan struktur table seperti berikut :

Field Name	Type	Width	Index
No_trans	text	5	primary
no-rek	text	5	
Tgl_trans	date	8	
Jml_pinjam	currency	8	

d) Buat tabel setoran dengan struktur table seperti berikut :

Field Name	Type	Width	Index
No_trans	text	5	primary
no-rek	text	5	
Tgl_trans	date	8	
Jml_setor	currency	8	

e) Buat tabel simpan dengan struktur table seperti berikut :

Field Name	Type	Width	Index
No_trans	text	5	primary
no-rek	text	5	
Tgl_trans	date	8	
Jml_simpan	currency	8	

f) Buat tabel bunga dengan struktur table seperti berikut :

Field Name	Type	Width	Index
Bunga_spn	single	4	Primary
Bunga_pjm	single	4	primary

g) Buat tabel password dengan struktur table seperti berikut :

Field Name	Type	Width	Index
password	Text	10	Primary
nama	text	10	

3. Buat Desain Form untuk memanipulasi data

a) Desain form login

Kode Program :

```
Data1.Recordset.Seek "=", Text2.Text
   If Data1.Recordset.Fields("password") = Text2.Text Then
      ok.SetFocus
   Else
      MsgBox "Password Salah", vbOKOnly, "Pesan:"
      Text2.SetFocus
   End If
salah:
End Sub


Private Sub selesai_Click()
End
End Sub

Private Sub text1_keypress(KeyAscii As Integer)
If KeyAscii = 13 Then
   Text2.SetFocus
End If
End Sub

Private Sub text2_keypress(KeyAscii As Integer)
If KeyAscii = 13 Then
   ok.SetFocus
End If
End Sub
```

b) Desain form bunga

Kode Program :

```
Private Sub batal_Click()
Text1.Visible = True
Text2.Visible = True
Text3.Visible = False
Text4.Visible = False
mati
edit.Enabled = True
keluar.Enabled = True
batal.Enabled = False
ok.Enabled = False
End Sub

Private Sub DBGrid1_RowColChange(LastRow As Variant, ByVal LastCol As Integer)
On Error GoTo x
tampilan
x:
End Sub

Private Sub edit_Click()
On Error GoTo x
Text1.Visible = False
Text2.Visible = False
Text3.Visible = True
Text4.Visible = True
Text3.Enabled = True
Text4.Enabled = True
Text3.BackColor = &H80000005
Text4.BackColor = &H80000005
keluar.Enabled = False
edit.Enabled = False
ok.Enabled = True
batal.Enabled = True
x:
End Sub
```

```
Private Sub Form_Load()
On Error GoTo x
Text3.Visible = False
Text4.Visible = False
mati
batal.Enabled = False
ok.Enabled = False
x:
End Sub

Private Sub keluar_Click()
Unload Me
End Sub

Private Sub ok_Click()
On Error GoTo x
Data1.Recordset.Index = "xb_spn"
Data1.Recordset.Seek "=", Text1.Text
Data1.Recordset.Index = "xb_pjm"
Data1.Recordset.Seek "=", Text2.Text
      If Not Data1.Recordset.NoMatch Then
        Data1.Recordset.edit
        Data1.Recordset!bunga_spn = Text3.Text
        Data1.Recordset!bunga_pjm = Text4.Text
        Pesan = MsgBox("Yakin Simpan Hasil Editan ??", vbYesNo,
"PEMBERITAHUAN")
        If Pesan = vbYes Then
              Data1.Recordset!bunga_spn = Text3.Text
              Data1.Recordset!bunga_pjm = Text4.Text
              Data1.Recordset.Update
              Data1.Refresh
         Else
         End If
         End If
edit.Enabled = True
keluar.Enabled = True
batal.Enabled = False
ok.Enabled = False
mati
x:
End Sub

Private Sub tampilan()
On Error GoTo x
With Data1.Recordset
Text1.Text = !bunga_spn
Text2.Text = !bunga_pjm
End With
```

```
x:
End Sub

Private Sub mati()
On Error GoTo x
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Text1.BackColor = &H80000004
Text2.BackColor = &H80000004
Text3.BackColor = &H80000004
Text4.BackColor = &H80000004
x:
End Sub

Private Sub text3_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   Text4.SetFocus
End If

x:
End Sub

Private Sub text4_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   ok.SetFocus
End If

x:
End Sub
```

c) Desain Form Nasabah

Kode Program :

```
Private Sub b()
On Error GoTo x
Data4.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 390 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 13) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 360 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 12) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 330 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 11) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 300 Then
```

```
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 10) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 270 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 9) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 240 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 8) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 210 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 7) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 180 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 6) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 150 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
```

```
      ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 5) + _
      Data1.Recordset!saldo_pjm
      Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 120 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = _
      ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 4) + _
      Data1.Recordset!saldo_pjm
      Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 90 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = _
      ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 3) + _
      Data1.Recordset!saldo_pjm
      Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 60 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = _
      ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 2) + _
      Data1.Recordset!saldo_pjm
      Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 30 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = _
      ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 1) + _
      Data1.Recordset!saldo_pjm
      Data1.Recordset.Update
Data8.Recordset.MoveLast
Data6.Recordset.MoveLast
If Date - Data8.Recordset!tgl_trans <= 30 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data6.Recordset!spjm
      Data1.Recordset.Update
End If
End If
```

```
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
x:
End Sub

Private Sub a()
On Error GoTo x
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 390 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_spn = _
      ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 13) + _
      Data1.Recordset!saldo_spn
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 360 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_spn = _
      ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 12) + _
      Data1.Recordset!saldo_spn
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 330 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_spn = _
      ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 11) + _
      Data1.Recordset!saldo_spn
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 300 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_spn = _
```

```
      ((((Data3.Recordset!bunga_spn    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_spn) * 10) + _
      Data1.Recordset!saldo_spn
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 270 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_spn = _
      ((((Data3.Recordset!bunga_spn    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_spn) * 9) + _
      Data1.Recordset!saldo_spn
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 240 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_spn = _
      ((((Data3.Recordset!bunga_spn    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_spn) * 8) + _
      Data1.Recordset!saldo_spn
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 210 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_spn = _
      ((((Data3.Recordset!bunga_spn    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_spn) * 7) + _
      Data1.Recordset!saldo_spn
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 180 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_spn = _
      ((((Data3.Recordset!bunga_spn    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_spn) * 6) + _
      Data1.Recordset!saldo_spn
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 150 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_spn = _
      ((((Data3.Recordset!bunga_spn    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_spn) * 5) + _
      Data1.Recordset!saldo_spn
```

```
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 120 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 4) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 90 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 3) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 60 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 2) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 30 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 1) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Data7.Recordset.MoveLast
Data5.Recordset.MoveLast
If Date - Data7.Recordset!tgl_trans <= 30 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = Data5.Recordset!sspn
        Data1.Recordset.Update
End If
End If
End If
End If
End If
```

```
End If
End If
End If
End If
End If
End If
End If
End If
End If
x:
End Sub

Private Sub c()
On Error GoTo x
Data2.Recordset.MoveLast
Data5.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 390 Then
      Data1.Recordset.edit
      On Error Resume Next
      Data1.Recordset!no_rek = Data5.Recordset!no_rek
      Data1.Recordset!saldo_spn = Data5.Recordset!sspn
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data5.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 360 Then
      Data1.Recordset.edit
      On Error Resume Next
      Data1.Recordset!no_rek = Data5.Recordset!no_rek
      Data1.Recordset!saldo_spn = Data5.Recordset!sspn
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data5.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 330 Then
      Data1.Recordset.edit
      On Error Resume Next
      Data1.Recordset!no_rek = Data5.Recordset!no_rek
      Data1.Recordset!saldo_spn = Data5.Recordset!sspn
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data5.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 300 Then
      Data1.Recordset.edit
      On Error Resume Next
      Data1.Recordset!no_rek = Data5.Recordset!no_rek
      Data1.Recordset!saldo_spn = Data5.Recordset!sspn
```

```
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data5.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 270 Then
        Data1.Recordset.edit
        On Error Resume Next
        Data1.Recordset!no_rek = Data5.Recordset!no_rek
        Data1.Recordset!saldo_spn = Data5.Recordset!sspn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data5.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 240 Then
        Data1.Recordset.edit
        On Error Resume Next
        Data1.Recordset!no_rek = Data5.Recordset!no_rek
        Data1.Recordset!saldo_spn = Data5.Recordset!sspn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data5.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 210 Then
        Data1.Recordset.edit
        On Error Resume Next
        Data1.Recordset!no_rek = Data5.Recordset!no_rek
        Data1.Recordset!saldo_spn = Data5.Recordset!sspn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data5.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 180 Then
        Data1.Recordset.edit
        On Error Resume Next
        Data1.Recordset!no_rek = Data5.Recordset!no_rek
        Data1.Recordset!saldo_spn = Data5.Recordset!sspn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data5.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 150 Then
        Data1.Recordset.edit
        On Error Resume Next
        Data1.Recordset!no_rek = Data5.Recordset!no_rek
        Data1.Recordset!saldo_spn = Data5.Recordset!sspn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
```

```
Data5.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 120 Then
	Data1.Recordset.edit
	On Error Resume Next
	Data1.Recordset!no_rek = Data5.Recordset!no_rek
	Data1.Recordset!saldo_spn = Data5.Recordset!sspn
	Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data5.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 90 Then
	Data1.Recordset.edit
	On Error Resume Next
	Data1.Recordset!no_rek = Data5.Recordset!no_rek
	Data1.Recordset!saldo_spn = Data5.Recordset!sspn
	Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data5.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 60 Then
	Data1.Recordset.edit
	On Error Resume Next
	Data1.Recordset!no_rek = Data5.Recordset!no_rek
	Data1.Recordset!saldo_spn = Data5.Recordset!sspn
	Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data5.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 30 Then
	Data1.Recordset.edit
	On Error Resume Next
	Data1.Recordset!no_rek = Data5.Recordset!no_rek
	Data1.Recordset!saldo_spn = Data5.Recordset!sspn
	Data1.Recordset.Update
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
x:
```

```
End Sub

Private Sub d()
On Error GoTo x
Data4.Recordset.MoveLast
Data6.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 390 Then
      Data1.Recordset.edit
      On Error Resume Next
      Data1.Recordset!no_rek = Data6.Recordset!no_rek
      Data1.Recordset!saldo_pjm = Data6.Recordset!spjm
      Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
Data6.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 360 Then
      Data1.Recordset.edit
      On Error Resume Next
      Data1.Recordset!no_rek = Data6.Recordset!no_rek
      Data1.Recordset!saldo_pjm = Data6.Recordset!spjm
      Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
Data6.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 330 Then
      Data1.Recordset.edit
      On Error Resume Next
      Data1.Recordset!no_rek = Data6.Recordset!no_rek
      Data1.Recordset!saldo_pjm = Data6.Recordset!spjm
      Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
Data6.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 300 Then
      Data1.Recordset.edit
      On Error Resume Next
      Data1.Recordset!no_rek = Data6.Recordset!no_rek
      Data1.Recordset!saldo_pjm = Data6.Recordset!spjm
      Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
Data6.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 270 Then
      Data1.Recordset.edit
      On Error Resume Next
      Data1.Recordset!no_rek = Data6.Recordset!no_rek
      Data1.Recordset!saldo_pjm = Data6.Recordset!spjm
      Data1.Recordset.Update
```

```
Else
Data4.Recordset.MoveLast
Data6.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 240 Then
      Data1.Recordset.edit
      On Error Resume Next
      Data1.Recordset!no_rek = Data6.Recordset!no_rek
      Data1.Recordset!saldo_pjm = Data6.Recordset!spjm
      Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
Data6.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 210 Then
      Data1.Recordset.edit
      On Error Resume Next
      Data1.Recordset!no_rek = Data6.Recordset!no_rek
      Data1.Recordset!saldo_pjm = Data6.Recordset!spjm
      Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
Data6.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 180 Then
      Data1.Recordset.edit
      On Error Resume Next
      Data1.Recordset!no_rek = Data6.Recordset!no_rek
      Data1.Recordset!saldo_pjm = Data6.Recordset!spjm
      Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
Data6.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 150 Then
      Data1.Recordset.edit
      On Error Resume Next
      Data1.Recordset!no_rek = Data6.Recordset!no_rek
      Data1.Recordset!saldo_pjm = Data6.Recordset!spjm
      Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
Data6.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 120 Then
      Data1.Recordset.edit
      On Error Resume Next
      Data1.Recordset!no_rek = Data6.Recordset!no_rek
      Data1.Recordset!saldo_pjm = Data6.Recordset!spjm
      Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
Data6.Recordset.MoveLast
```

```
If Date - Data4.Recordset!tgl_trans >= 90 Then
	Data1.Recordset.edit
	On Error Resume Next
	Data1.Recordset!no_rek = Data6.Recordset!no_rek
	Data1.Recordset!saldo_pjm = Data6.Recordset!spjm
	Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
Data6.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 60 Then
	Data1.Recordset.edit
	On Error Resume Next
	Data1.Recordset!no_rek = Data6.Recordset!no_rek
	Data1.Recordset!saldo_pjm = Data6.Recordset!spjm
	Data1.Recordset.Update
Else
Data4.Recordset.MoveLast
Data6.Recordset.MoveLast
If Date - Data4.Recordset!tgl_trans >= 30 Then
	Data1.Recordset.edit
	On Error Resume Next
	Data1.Recordset!no_rek = Data6.Recordset!no_rek
	Data1.Recordset!saldo_pjm = Data6.Recordset!spjm
	Data1.Recordset.Update
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
x:
End Sub

Private Sub Form_Unload(Cancel As Integer)
c
d
End Sub

Private Sub Form_Activate()
On Error GoTo x
Data1.Recordset.Index = "xnasabah"
```

```
On Error Resume Next
Data1.Recordset.MoveLast
a
b
x:
End Sub

Private Sub Form_Load()
On Error GoTo x
warna_mati
mati
simpan2.Visible = False
x:
End Sub

Private Sub DBGrid1_Change()
If Data1.Recordset!j_kel = "W" Then
   DBGrid1.Columns(4) = "WANITA"
Else
   DBGrid1.Columns(4) = "PRIA"
End If
End Sub

Private Sub DBGrid1_RowColChange(LastRow As Variant, ByVal LastCol
As Integer)
On Error GoTo x
tampilan
x:
End Sub

Private Sub hapus_Click()
On Error GoTo x
If Text1.Text = "" Then
   MsgBox "Klik Data yang Akan Dihapus Dulu Di Grid ??", vbOKOnly,
"pesan"
   DBGrid1.Enabled = True
   DBGrid1.AllowDelete = True
Else
    With Data1.Recordset
    If Not .NoMatch Then
      Pesan = MsgBox("Yakin Akan Menghapus nasabah ber no rekening  "
& !no_rek & " ...??", vbYesNo, "pesan")
      If Pesan = vbYes Then
      pan
         .Delete
         Data1.Refresh
         DBGrid1.Refresh
         kosong
```

```
      End If
   End If
   End With
End If
On Error GoTo 0
Tambah.SetFocus
On Error Resume Next
Data1.Recordset.MoveLast
x:
End Sub

Private Sub refres_Click()
On Error GoTo x
Data1.Refresh
DBGrid1.Refresh
FORM_NASABAH.Refresh
On Error Resume Next
Data1.Recordset.MoveLast
x:
End Sub

Private Sub selesai_Click()
Unload Me
End Sub

Private Sub batal_Click()
On Error GoTo x
tombol_batal
warna_mati
simpan2.Visible = False
simpan.Visible = True
tampilan
refres.Enabled = True
Tambah.SetFocus
On Error Resume Next
Data1.Recordset.MoveLast
x:
End Sub

Private Sub simpan2_Click()
pan
On Error GoTo x
Data1.Recordset.Index = "xnasabah"
Data1.Recordset.Seek "=", Text1.Text
      If Not Data1.Recordset.NoMatch Then
         Data1.Recordset.edit
         Data1.Recordset!no_rek = Text1.Text
           Data1.Recordset!nama = Text3.Text
```

```
          If Option1.Value = True Then
             Data1.Recordset!j_kel = Option1.Caption
          ElseIf Option2.Value = True Then
             Data1.Recordset!j_kel = Option2.Caption
          End If
         Data1.Recordset!tempat_lhr = Text4.Text
         Data1.Recordset!tgl_lahir = DTPicker1.Value
         If Combo2.Text = "Islam" Then
             Combo2.Text = "1"
             Data1.Recordset!agama = Combo2.Text
          ElseIf Combo2.Text = "Kristen" Then
             Combo2.Text = "2"
             Data1.Recordset!agama = Combo2.Text
          ElseIf Combo2.Text = "Katolik" Then
             Combo2.Text = "3"
             Data1.Recordset!agama = Combo2.Text
          ElseIf Combo2.Text = "Hindu" Then
             Combo2.Text = "4"
             Data1.Recordset!agama = Combo2.Text
          ElseIf Combo2.Text = "Budha" Then
             Combo2.Text = "5"
             Data1.Recordset!agama = Combo2.Text
          End If
         Data1.Recordset!tlp = Text5.Text
         Data1.Recordset!alamat = Text6.Text
         Pesan = MsgBox("Yakin Simpan Hasil Editan ??", vbYesNo,
"PEMBERITAHUAN")
         If Pesan = vbYes Then
         Data1.Recordset.Update
         Data1.Refresh
         DBGrid1.Refresh
         Else
            simpan2.Enabled = False
         End If
        End If
Data1.Recordset.Index = "xnasabah"
On Error Resume Next
Data1.Recordset.MovePrevious
tombol_simpan
warna_mati
simpan.Visible = True
simpan2.Visible = False
selesai.SetFocus
x:
End Sub

Private Sub Tambah_Click()
On Error GoTo x
```

```
kosong
auto
tombol_tambah
warna_hidup
refres.Enabled = False
Text3.SetFocus
x:
End Sub

Private Sub edit_Click()
On Error GoTo x
warna_hidup
tombol_tambah
simpan.Visible = False
simpan2.Visible = True
Text2.SetFocus
x:
End Sub

Private Sub simpan_Click()
On Error GoTo x
If Text1.Text = "" Or Text3.Text = "" Or _
Text4.Text = "" Or Combo2.Text = "" Or _
Text5.Text = "" Or Text6.Text = "" Then
   MsgBox "Silahkan Lengkapi Datanya!!", vbOKOnly + vbExclamation,
"PERINGATAN"
   If Text1.Text = "" Then
      Text1.SetFocus
   ElseIf Text3.Text = "" Then
      Text3.SetFocus
   ElseIf Text4.Text = "" Then
      Text4.SetFocus
   ElseIf Text5.Text = "" Then
      Text5.SetFocus
   ElseIf Text6.Text = "" Then
      Text6.SetFocus
  ElseIf Combo2.Text = "" Then
      Combo2.SetFocus
   End If
Else
pan
   Data1.Recordset.Index = "xnasabah"
   Data1.Recordset.Seek "=", Text1.Text
   If Data1.Recordset.NoMatch Then
      Data1.Recordset.AddNew
      Data1.Recordset!no_rek = Text1.Text
      Data1.Recordset!nama = Text3.Text
      If Option1.Value = True Then
```

```
            Data1.Recordset!j_kel = Option1.Caption
         ElseIf Option2.Value = True Then
            Data1.Recordset!j_kel = Option2.Caption
         End If
      Data1.Recordset!tempat_lhr = Text4.Text
      Data1.Recordset!tgl_lahir = DTPicker1.Value
         If Combo2.Text = "Islam" Then
            Combo2.Text = "1"
            Data1.Recordset!agama = Combo2.Text
         ElseIf Combo2.Text = "Kristen" Then
            Combo2.Text = "2"
            Data1.Recordset!agama = Combo2.Text
         ElseIf Combo2.Text = "Katolik" Then
            Combo2.Text = "3"
            Data1.Recordset!agama = Combo2.Text
         ElseIf Combo2.Text = "Hindu" Then
            Combo2.Text = "4"
            Data1.Recordset!agama = Combo2.Text
         ElseIf Combo2.Text = "Budha" Then
            Combo2.Text = "5"
            Data1.Recordset!agama = Combo2.Text
         End If
      Data1.Recordset!tlp = Text5.Text
      Data1.Recordset!alamat = Text6.Text
      Data1.Recordset.Update
      DBGrid1.Refresh
   Else
      MsgBox " No Rekening Sudah Ada,Jadi Tidak Boleh sama!",
vbOKOnly + vbCritical, "KESALAHAN"
      simpan.Enabled = False
   End If
Data1.Recordset.Index = "xnasabah"
On Error Resume Next
Data1.Recordset.MoveLast
tombol_simpan
warna_mati
Tambah.SetFocus
End If
x:
End Sub

Private Sub kosong()
On Error GoTo x
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
```

```
Text6.Text = ""
Combo1.Text = ""
Combo2.Text = ""
Option1.Value = False
Option2.Value = False
DTPicker1.Value = Date
x:
End Sub

Private Sub mati()
On Error GoTo x
simpan.Enabled = False
batal.Enabled = False
x:
End Sub
Private Sub pan()
On Error GoTo x
Dim i As Integer, panbar(32000) As String
ProgressBar1.Min = LBound(panbar)
ProgressBar1.Max = UBound(panbar)
ProgressBar1.Visible = True
ProgressBar1.Value = ProgressBar1.Min
For i = LBound(panbar) To UBound(panbar)
   ProgressBar1.Value = i
Next i
ProgressBar1.Visible = False
ProgressBar1.Value = ProgressBar1.Min
x:
End Sub
Private Sub tombol_simpan()
On Error GoTo x
simpan.Enabled = False
batal.Enabled = False
Tambah.Enabled = True
edit.Enabled = True
Hapus.Enabled = True
selesai.Enabled = True
refres.Enabled = True
x:
End Sub

Private Sub tombol_tambah()
On Error GoTo x
Tambah.Enabled = False
edit.Enabled = False
Hapus.Enabled = False
selesai.Enabled = False
simpan.Enabled = True
```

```
batal.Enabled = True
refres.Enabled = True
x:
End Sub

Private Sub tombol_batal()
On Error GoTo x
simpan.Enabled = False
batal.Enabled = False
Tambah.Enabled = True
edit.Enabled = True
Hapus.Enabled = True
selesai.Enabled = True
x:
End Sub

Private Sub warna_hidup()
On Error GoTo x
Text2.Enabled = True
Text3.Enabled = True
Text4.Enabled = True
Text5.Enabled = True
Text6.Enabled = True
Option1.Enabled = True
Option2.Enabled = True
Combo1.Enabled = True
Combo2.Enabled = True
DTPicker1.Enabled = True
Text2.BackColor = &H80000005
Text3.BackColor = &H80000005
Text4.BackColor = &H80000005
Text5.BackColor = &H80000005
Text6.BackColor = &H80000005
Option1.BackColor = &H80000006
Option2.BackColor = &H80000006
Combo1.BackColor = &H80000005
Combo2.BackColor = &H80000005
x:
End Sub

Private Sub warna_mati()
On Error GoTo x
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Text5.Enabled = False
Text6.Enabled = False
```

```
Option1.Enabled = False
Option2.Enabled = False
DTPicker1.Enabled = False
Combo1.Enabled = False
Combo2.Enabled = False
Text1.BackColor = &H80000004
Text2.BackColor = &H80000004
Text3.BackColor = &H80000004
Text4.BackColor = &H80000004
Text5.BackColor = &H80000004
Text6.BackColor = &H80000004
Option1.BackColor = &H80000006
Option2.BackColor = &H80000006
Combo1.BackColor = &H80000004
Combo2.BackColor = &H80000004
x:
End Sub

Private Sub tampilan()
On Error GoTo x
With Data1.Recordset
Text1.Text = !no_rek
Text2.Text = !nis
Text3.Text = !nama
Combo1.Text = !kelas
If Data1.Recordset!j_kel = "P" Then
   Option1.Value = True
Else: Option2.Value = True
End If
Text4.Text = !tempat_lhr
Text5.Text = !tlp
Text6.Text = !alamat
DTPicker1.Value = !tgl_lahir
If Data1.Recordset!agama = "1" Then
   Combo2.Text = "Islam"
ElseIf Data1.Recordset!agama = "2" Then
   Combo2.Text = "Kristen"
ElseIf Data1.Recordset!agama = "3" Then
   Combo2.Text = "Katolik"
ElseIf Data1.Recordset!agama = "4" Then
   Combo2.Text = "Hindu"
ElseIf Data1.Recordset!agama = "5" Then
   Combo2.Text = "Budha"
End If
End With
x:
End Sub
```

```
Private Sub text1_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   Text2.SetFocus
End If
x:
End Sub

Private Sub text2_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   Text3.SetFocus
End If
If Not IsNumeric(Chr(KeyAscii)) Then
   KeyAscii = 0
End If
x:
End Sub

Private Sub text3_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   Combo1.SetFocus
End If
x:
End Sub

Private Sub combo1_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   Option1.SetFocus
End If
x:
End Sub

Private Sub option1_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   Option2.SetFocus
End If
x:
End Sub

Private Sub option2_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   Text4.SetFocus
End If
```

```
x:
End Sub

Private Sub text4_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   Combo2.SetFocus
End If
x:
End Sub

Private Sub combo2_keypress(KeyAscii As Integer)
If KeyAscii = 13 Then
   Text5.SetFocus
End If
End Sub

Private Sub text5_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   Text6.SetFocus
End If
If Not IsNumeric(Chr(KeyAscii)) Then
   KeyAscii = 0
End If
x:
End Sub

Private Sub text6_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   simpan.SetFocus
End If
x:
End Sub

Private Sub auto()
On Error GoTo x
Dim urutan As String * 5
Dim hitung As Byte
With Data1.Recordset
On Error Resume Next
Data1.Recordset.MoveLast
If .RecordCount = 0 Then
   urutan = "00001"
Else
   .MoveLast
   hitung = Val(Right(.Fields("no_rek"), 5)) + 1
```

```
   urutan = Right("00000" & hitung, 5)
End If
Text1 = urutan
End With
x:
End Sub
Private Sub cari_Click()
Data1.Recordset.Index = "xnasabah"
Data1.Recordset.Seek "=", Text9.Text
If Data1.Recordset.NoMatch Then
   MsgBox "Tidak Ada Data Tersebut !!!"
   Text9.SetFocus
Else
   Text9.SetFocus
End If
End Sub
```

d) Desain Form Transaksi Pinjam

Kode Program :

```
Private Sub b()
On Error GoTo x
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 390 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = _
      ((((Data3.Recordset!bunga_pjm / 100) + 0.00000000023) *
Data1.Recordset!saldo_pjm) * 13) + _
      Data1.Recordset!saldo_pjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 360 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = _
```

```
      ((((Data3.Recordset!bunga_pjm / 100) + 0.00000000023) *
Data1.Recordset!saldo_pjm) * 12) + _
      Data1.Recordset!saldo_pjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 330 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = _
      ((((Data3.Recordset!bunga_pjm / 100) + 0.00000000023) *
Data1.Recordset!saldo_pjm) * 11) + _
      Data1.Recordset!saldo_pjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 300 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = _
      ((((Data3.Recordset!bunga_pjm / 100) + 0.00000000023) *
Data1.Recordset!saldo_pjm) * 10) + _
      Data1.Recordset!saldo_pjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 270 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = _
      ((((Data3.Recordset!bunga_pjm / 100) + 0.00000000023) *
Data1.Recordset!saldo_pjm) * 9) + _
      Data1.Recordset!saldo_pjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 240 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = _
      ((((Data3.Recordset!bunga_pjm / 100) + 0.00000000023) *
Data1.Recordset!saldo_pjm) * 8) + _
      Data1.Recordset!saldo_pjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 210 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = _
      ((((Data3.Recordset!bunga_pjm / 100) + 0.00000000023) *
Data1.Recordset!saldo_pjm) * 7) + _
      Data1.Recordset!saldo_pjm
```

```
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 180 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_pjm) * 6) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 150 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_pjm) * 5) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 120 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_pjm) * 4) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 90 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_pjm) * 3) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 60 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_pjm) * 2) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
```

```
If Date - Data2.Recordset!tgl_trans >= 30 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = _
      ((((Data3.Recordset!bunga_pjm / 100) + 0.00000000023) *
Data1.Recordset!saldo_pjm) * 1) + _
      Data1.Recordset!saldo_pjm
      Data1.Recordset.Update
Data5.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans <= 30 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
x:
End Sub

Private Sub c()
On Error GoTo x
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 390 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 360 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
```

```
If Date - Data2.Recordset!tgl_trans >= 330 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 300 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 270 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 240 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 210 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 180 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 150 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
```

```
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 120 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 90 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 60 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 30 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
x:
End Sub

Private Sub DBCombo1_Change()
On Error GoTo x
With Data1.Recordset
Data1.Recordset.Index = "xnasabah"
Data1.Recordset.Seek "=", DBCombo1.Text
If Not .NoMatch Then
```

```
   Text2.Text = !nama
   On Error Resume Next
   Text3.Text = !saldo_pjm
Else
End If
End With
Data1.Recordset.Index = "xnasabah"
x:
End Sub



Private Sub DBGrid1_RowColChange(LastRow As Variant, ByVal LastCol
As Integer)
On Error GoTo x
tampilan
DBGrid1.Refresh
x:
End Sub

Private Sub Form_Activate()
On Error GoTo x
Data1.Recordset.Index = "xnasabah"
Data2.Recordset.Index = "xpinjam"
On Error Resume Next
Data2.Recordset.MoveLast
b
Tambah.SetFocus
x:
End Sub

Private Sub Form_Unload(Cancel As Integer)
c
End Sub

Private Sub hapus_Click()
On Error GoTo x
If Text1.Text = "" Then
   MsgBox "Klik Data yang Akan Dihapus Dulu Di Grid ??", vbOKOnly,
"pesan"
   DBGrid1.Enabled = True
   DBGrid1.AllowDelete = True
Else
    With Data2.Recordset
    If Not .NoMatch Then
      Pesan = MsgBox("Yakin Akan Menghapus data Pinjam tersebut..??",
vbYesNo, "pesan")
      On Error Resume Next
```

```
      Data1.Recordset.Index = "xpjm"
      Data1.Recordset.Seek "=", Text3.Text
      If Pesan = vbYes Then
      pan
         .Delete
         With Data4.Recordset
         If Not .NoMatch Then
         .Delete
         End If
         End With
         On Error Resume Next
         If Not Data1.Recordset.NoMatch Then
         Data1.Recordset.edit
         Data1.Recordset!saldo_pjm = (Val(Text3.Text) - Val(Text4.Text))
         Data1.Recordset.Update
         Data1.Refresh
          bersih
         End If
         Data1.Refresh
         Data2.Refresh
         DBGrid1.Refresh
         kosong
      End If
   End If
   End With
   End If
On Error GoTo 0
On Error Resume Next
Data2.Recordset.MoveLast
x:
End Sub

Private Sub bersih()
On Error GoTo x
      Data4.Recordset.Index = "xspn1"
      Data4.Recordset.Seek "=", Text3.Text
      If Not Data4.Recordset.NoMatch Then
         Data4.Recordset.edit
         Data4.Recordset!sspn = (Val(Text3.Text) - Val(Text4.Text))
         Data4.Recordset.Update
         Data4.Refresh
      End If
         Data4.Refresh
         Data4.Refresh
         DBGrid1.Refresh
         kosong
x:
End Sub
```

```
Private Sub refres_Click()
On Error GoTo x
FORM_PINJAM.Refresh
Data1.Refresh
Data2.Refresh
DBGrid1.Refresh
On Error Resume Next
Data2.Recordset.MoveLast
x:
End Sub

Private Sub selesai_Click()
Unload Me
End Sub

Private Sub Tambah_Click()
On Error GoTo x
kosong
auto
tombol_tambah
warna_hidup
DBCombo1.SetFocus
x:
End Sub

Private Sub batal_Click()
On Error GoTo x
kosong
tombol_batal
warna_mati
simpan2.Visible = False
simpan.Visible = True
tampilan
x:
End Sub

Private Sub Form_Load()
On Error GoTo x
mati
warna_mati
kosong
simpan2.Visible = False
x:
End Sub

Private Sub coba()
On Error GoTo x
Data4.Recordset.Index = "xpjm2"
```

```
Data4.Recordset.Seek "=", Text3.Text
   If Data4.Recordset.NoMatch Then
      Data4.Recordset.AddNew
      Data4.Recordset!no_rek = DBCombo1.Text
      Data4.Recordset!spjm = Val(Text3.Text) + Val(Text4.Text)
      Data4.Recordset.Update
     ElseIf Not Data4.Recordset.NoMatch Then
        Data4.Recordset.edit
        Data4.Recordset!no_rek = DBCombo1.Text
        Data4.Recordset!sspn = Val(Text3.Text) + Val(Text4.Text)
        Data4.Recordset.Update
        Data4.Refresh
      End If

      DBGrid1.Refresh

x:
End Sub

Private Sub pan()
On Error GoTo x
Dim i As Integer, panbar(32000) As String
ProgressBar1.Min = LBound(panbar)
ProgressBar1.Max = UBound(panbar)
ProgressBar1.Visible = True
ProgressBar1.Value = ProgressBar1.Min
For i = LBound(panbar) To UBound(panbar)
   ProgressBar1.Value = i
Next i
ProgressBar1.Visible = False
ProgressBar1.Value = ProgressBar1.Min
x:
End Sub
Private Sub simpan_Click()
On Error GoTo x
If Text1.Text = "" Or DBCombo1.Text = "" Or Text4.Text = "" Then
   MsgBox "Silahkan Lengkapi Datanya!!", vbOKOnly + vbExclamation,
"PERINGATAN"
   If Text1.Text = "" Then
      Text1.SetFocus
   ElseIf DBCombo1.Text = "" Then
     DBCombo1.SetFocus
   ElseIf Text4.Text = "" Then
      Text4.SetFocus
   End If
Else
pan
   Data2.Recordset.Index = "xpinjam"
```

```
   Data2.Recordset.Seek "=", Text1.Text
   With Data1.Recordset
   Data1.Recordset.Index = "xnasabah"
   Data1.Recordset.Seek "=", Text3.Text
   If Not Text3.Text = 0 Then
      MsgBox "Maaf Anda belum melunasi pinjaman" + Chr(13) _
      + Chr(13) + "      Jadi Tidak boleh Meminjam Lagi ", vbOKOnly,
"Peringatan"
      warna_mati
   Else
   If Data2.Recordset.NoMatch Then
      Data2.Recordset.AddNew
      Data2.Recordset!no_trans = Text1.Text
      Data2.Recordset!tgl_trans = DTPicker1.Value
      Data2.Recordset!no_rek = DBCombo1.Text
      Data2.Recordset!jml_pinjam = Text4.Text
         Data1.Recordset.Index = "xpjm"
         Data1.Recordset.Seek "=", Text3.Text
         If Not Data1.Recordset.NoMatch Then
            Data1.Recordset.edit
            Data1.Recordset!saldo_pjm = Val(Text3.Text) + (Text4.Text)
            Data1.Recordset.Update
            Data1.Refresh
         Else
         If Data1.Recordset.NoMatch Then
            Data1.Recordset.AddNew
            Data1.Recordset!saldo_pjm = Text4.Text
            Data1.Recordset.Update
            Data1.Refresh
         End If
         End If
      Data2.Recordset.Update
      DBGrid1.Refresh
      coba
   Else
      MsgBox " Maaf No transaksi sudah Ada !", vbOKOnly + vbCritical,
"KESALAHAN"
      simpan.Enabled = False
   End If
   End If
   End With
Data2.Recordset.Index = "xpinjam"
On Error Resume Next
Data2.Recordset.MoveLast
On Error Resume Next
tombol_simpan
warna_mati
Tambah.SetFocus
```

```
End If
x:
End Sub


Private Sub kosong()
On Error GoTo x
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text6.Text = ""
DBCombo1.Text = ""
DTPicker1.Value = Date
x:
End Sub

Private Sub mati()
On Error GoTo x
simpan.Enabled = False
batal.Enabled = False
x:
End Sub

Private Sub tombol_simpan()
On Error GoTo x
simpan.Enabled = False
batal.Enabled = False
Tambah.Enabled = True
Hapus.Enabled = True
selesai.Enabled = True
refres.Enabled = True
x:
End Sub

Private Sub tombol_tambah()
On Error GoTo x
Tambah.Enabled = False
Hapus.Enabled = False
selesai.Enabled = False
simpan.Enabled = True
batal.Enabled = True
refres.Enabled = False
x:
End Sub

Private Sub tombol_batal()
On Error GoTo x
```

```
simpan.Enabled = False
batal.Enabled = False
Tambah.Enabled = True
Hapus.Enabled = True
selesai.Enabled = True
refres.Enabled = True
x:
End Sub

Private Sub warna_hidup()
On Error GoTo x
Text4.Enabled = True
DBCombo1.Enabled = True
DTPicker1.Enabled = True
Text4.BackColor = &H80000005
DBCombo1.BackColor = &H80000005
x:
End Sub

Private Sub warna_mati()
On Error GoTo x
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Text6.Enabled = False
DTPicker1.Enabled = False
DBCombo1.Enabled = False
Text1.BackColor = &H80000004
Text2.BackColor = &H80000004
Text3.BackColor = &H80000004
Text4.BackColor = &H80000004
Text6.BackColor = &H80000004
DBCombo1.BackColor = &H80000004
x:
End Sub

Private Sub tampilan()
On Error GoTo x
With Data2.Recordset
Text1.Text = !no_trans
DBCombo1.Text = !no_rek
DTPicker1.Value = !tgl_trans
Text4.Text = !jml_pinjam
DBGrid1.Refresh
With Data1.Recordset
Text6.Text = Val(Text3.Text)
Data1.Refresh
```

```
DBGrid1.Refresh
End With
End With
x:
End Sub
Private Sub dbcombo1_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   Text4.SetFocus
End If
x:
End Sub
Private Sub text1_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   DBCombo1.SetFocus
End If
x:
End Sub

Private Sub Text4_Change()
On Error GoTo x
Text6.Text = Val(Text3.Text) + (Text4.Text)
x:
End Sub

Private Sub text4_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   simpan.SetFocus
End If
If Not IsNumeric(Chr(KeyAscii)) Then
   KeyAscii = 0
End If
x:
End Sub

Private Sub auto()
On Error GoTo x
Dim urutan As String * 5
Dim hitung As Byte
With Data2.Recordset
If .RecordCount = 0 Then
   urutan = "00001"
Else
   .MoveLast
   hitung = Val(Right(.Fields("no_trans"), 5)) + 1
   urutan = Right("00000" & hitung, 5)
```

```
End If
Text1 = urutan
End With
x:
End Sub
```

e) Desain Form Setoran

Kode Program :

```
Private Sub b()
On Error GoTo x
Data5.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 390 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = _
      ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 13) + _
      Data1.Recordset!saldo_pjm
      Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 360 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = _
      ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 12) + _
      Data1.Recordset!saldo_pjm
      Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 330 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = _
      ((((Data3.Recordset!bunga_pjm    /    100)    +    0.00000000023)    *
Data1.Recordset!saldo_pjm) * 11) + _
```

```
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 300 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_pjm) * 10) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 270 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_pjm) * 9) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 240 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_pjm) * 8) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 210 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_pjm) * 7) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 180 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = _
        ((((Data3.Recordset!bunga_pjm     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_pjm) * 6) + _
        Data1.Recordset!saldo_pjm
        Data1.Recordset.Update
Else
```

```
Data5.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 150 Then
	Data1.Recordset.edit
	Data1.Recordset!saldo_pjm = _
	((((Data3.Recordset!bunga_pjm / 100) + 0.00000000023) *
Data1.Recordset!saldo_pjm) * 5) + _
	Data1.Recordset!saldo_pjm
	Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 120 Then
	Data1.Recordset.edit
	Data1.Recordset!saldo_pjm = _
	((((Data3.Recordset!bunga_pjm / 100) + 0.00000000023) *
Data1.Recordset!saldo_pjm) * 4) + _
	Data1.Recordset!saldo_pjm
	Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 90 Then
	Data1.Recordset.edit
	Data1.Recordset!saldo_pjm = _
	((((Data3.Recordset!bunga_pjm / 100) + 0.00000000023) *
Data1.Recordset!saldo_pjm) * 3) + _
	Data1.Recordset!saldo_pjm
	Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 60 Then
	Data1.Recordset.edit
	Data1.Recordset!saldo_pjm = _
	((((Data3.Recordset!bunga_pjm / 100) + 0.00000000023) *
Data1.Recordset!saldo_pjm) * 2) + _
	Data1.Recordset!saldo_pjm
	Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 30 Then
	Data1.Recordset.edit
	Data1.Recordset!saldo_pjm = _
	((((Data3.Recordset!bunga_pjm / 100) + 0.00000000023) *
Data1.Recordset!saldo_pjm) * 1) + _
	Data1.Recordset!saldo_pjm
	Data1.Recordset.Update
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans <= 30 Then
	Data1.Recordset.edit
```

```
        Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
        Data1.Recordset.Update
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
x:
End Sub

Private Sub c()
On Error GoTo x
Data5.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 390 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
        Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 360 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
        Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 330 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
        Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 300 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
        Data1.Recordset.Update
```

```
Else
Data5.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 270 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 240 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 210 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 180 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 150 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 120 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
      Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 90 Then
      Data1.Recordset.edit
      Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
```

```
        Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 60 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
        Data1.Recordset.Update
Else
Data5.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans >= 30 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_pjm = Data4.Recordset!spjm
        Data1.Recordset.Update
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
x:
End Sub

Private Sub DBCombo1_Change()
On Error GoTo x
With Data1.Recordset
Data1.Recordset.Index = "xnasabah"
Data1.Recordset.Seek "=", DBCombo1.Text
If Not .NoMatch Then
   Text2.Text = !nama
   On Error Resume Next
   Text3.Text = !saldo_pjm
Else
End If
End With
Data1.Recordset.Index = "xnasabah"
x:
End Sub
```

```
Private Sub DBGrid1_RowColChange(LastRow As Variant, ByVal LastCol As Integer)
On Error GoTo x
tampilan
x:
End Sub
Private Sub Form_Activate()
On Error GoTo x
Data1.Recordset.Index = "xnasabah"
Data2.Recordset.Index = "xsetor"
On Error Resume Next
Data2.Recordset.MoveLast
b
Tambah.SetFocus
x:
End Sub

Private Sub Form_Unload(Cancel As Integer)
c
End Sub

Private Sub hapus_Click()
On Error GoTo x
If Text1.Text = "" Then
   MsgBox "Klik Data yang Akan Dihapus Dulu Di Grid ??", vbOKOnly, "pesan"
   DBGrid1.Enabled = True
   DBGrid1.AllowDelete = True
Else
    With Data2.Recordset
    If Not .NoMatch Then
      Pesan = MsgBox("Yakin Akan Menghapus data Setoran tersebut..??", vbYesNo, "pesan")
      On Error Resume Next
      Data1.Recordset.Index = "xpjm"
      Data1.Recordset.Seek "=", Text3.Text
      If Pesan = vbYes Then
      pan
         .Delete
         On Error Resume Next
         If Not Data1.Recordset.NoMatch Then
         Data1.Recordset.edit
         Data1.Recordset!saldo_pjm = (Val(Text3.Text) + Val(Text4.Text))
         Data1.Recordset.Update
         Data1.Refresh
          bersih
         End If
         Data1.Refresh
```

```
          Data2.Refresh
          DBGrid1.Refresh
          kosong
       End If
   End If
   End With
   End If
On Error GoTo 0
On Error Resume Next
Data2.Recordset.MoveLast
x:
End Sub

Private Sub bersih()
On Error GoTo x
       Data4.Recordset.Index = "xpjm2"
       Data4.Recordset.Seek "=", Text3.Text
       If Not Data4.Recordset.NoMatch Then
          Data4.Recordset.edit
          Data4.Recordset!spjm = (Val(Text3.Text) - Val(Text4.Text))
          Data4.Recordset.Update
          Data4.Refresh
       End If
          Data4.Refresh
          Data4.Refresh
          DBGrid1.Refresh
          kosong
x:
End Sub

Private Sub refres_Click()
On Error GoTo x
FORM_SETORAN.Refresh
Data1.Refresh
Data2.Refresh
DBGrid1.Refresh
On Error Resume Next
Data2.Recordset.MoveLast
x:
End Sub

Private Sub selesai_Click()
Unload Me
End Sub

Private Sub Tambah_Click()
On Error GoTo x
kosong
```

```
tombol_tambah
warna_hidup
auto
DBCombo1.SetFocus
x:
End Sub
Private Sub batal_Click()
On Error GoTo x
kosong
On Error Resume Next
tombol_batal
warna_mati
simpan1.Visible = False
simpan.Visible = True
tampilan
x:
End Sub

Private Sub Form_Load()
On Error GoTo x
mati
warna_mati
kosong
simpan1.Visible = False
x:
End Sub

Private Sub coba()
On Error GoTo x
Data4.Recordset.Index = "xpjm2"
Data4.Recordset.Seek "=", Text3.Text
   If Data4.Recordset.NoMatch Then
      Data4.Recordset.AddNew
      Data4.Recordset!no_rek = DBCombo1.Text
      Data4.Recordset!spjm = Val(Text3.Text) - Val(Text4.Text)
      Data4.Recordset.Update
     ElseIf Not Data4.Recordset.NoMatch Then
         Data4.Recordset.edit
         Data4.Recordset!no_rek = DBCombo1.Text
         Data4.Recordset!spjm = Val(Text3.Text) - Val(Text4.Text)
         Data4.Recordset.Update
         Data4.Refresh
      End If
      DBGrid1.Refresh

x:
End Sub
```

```
Private Sub pan()
On Error GoTo x
Dim i As Integer, panbar(32000) As String
ProgressBar1.Min = LBound(panbar)
ProgressBar1.Max = UBound(panbar)
ProgressBar1.Visible = True
ProgressBar1.Value = ProgressBar1.Min
For i = LBound(panbar) To UBound(panbar)
   ProgressBar1.Value = i
Next i
ProgressBar1.Visible = False
ProgressBar1.Value = ProgressBar1.Min
x:
End Sub
Private Sub simpan_Click()
On Error GoTo x
If Text1.Text = "" Or DBCombo1.Text = "" Or Text4.Text = "" Then
   MsgBox "Silahkan Lengkapi Datanya!!", vbOKOnly + vbExclamation,
"PERINGATAN"
   If Text1.Text = "" Then
      Text1.SetFocus
   ElseIf DBCombo1.Text = "" Then
     DBCombo1.SetFocus
   ElseIf Text4.Text = "" Then
      Text4.SetFocus
   End If
Else
pan
   Data2.Recordset.Index = "xsetor"
   Data2.Recordset.Seek "=", Text1.Text
   With Data1.Recordset
   Data1.Recordset.Index = "xnasabah"
   Data1.Recordset.Seek "=", Text3.Text
   If Text3.Text = 0 Then
      MsgBox "Maaf Anda sudah Tidak Punya Pinjaman" + Chr(13) _
      + Chr(13) + "    Jadi Tidak Perlu Setor Lagi ", vbOKOnly, "Peringatan"
      Text2.Text = ""
      Text3.Text = ""
      DBCombo1.SetFocus
   Else
   If Data2.Recordset.NoMatch Then
      Data2.Recordset.AddNew
      Data2.Recordset!no_trans = Text1.Text
      Data2.Recordset!tgl_trans = DTPicker1.Value
      Data2.Recordset!no_rek = DBCombo1.Text
      Data2.Recordset!jml_setor = Text4.Text
      On Error Resume Next
         Data1.Recordset.Index = "xpjm"
```

```
          Data1.Recordset.Seek "=", Text3.Text
          If Data1.Recordset.NoMatch Then
             Data1.Recordset.AddNew
             Data1.Recordset!saldo_pjm = Text4.Text
             Data1.Recordset.Update
             Data1.Refresh
          Else
          If Not Data1.Recordset.NoMatch Then
             Data1.Recordset.edit
             Data1.Recordset!saldo_pjm = Val(Text3.Text) - (Text4.Text)
             Data1.Recordset.Update
             Data1.Refresh
          End If
          End If
       Data2.Recordset.Update
       DBGrid1.Refresh
       coba
    Else
       MsgBox " Maaf No transaksi sudah Ada !", vbOKOnly + vbCritical,
"KESALAHAN"
       simpan.Enabled = False
    End If
    End If
    End With
Data2.Recordset.Index = "xsetor"
On Error Resume Next
Data2.Recordset.MoveLast
tombol_simpan
warna_mati
Tambah.SetFocus
End If
x:
End Sub

Private Sub kosong()
On Error GoTo x
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text6.Text = ""
DBCombo1.Text = ""
DTPicker1.Value = Date
x:
End Sub

Private Sub mati()
On Error GoTo x
```

```
simpan.Enabled = False
batal.Enabled = False
x:
End Sub

Private Sub tombol_simpan()
On Error GoTo x
simpan.Enabled = False
batal.Enabled = False
Tambah.Enabled = True
Hapus.Enabled = True
selesai.Enabled = True
refres.Enabled = True
x:
End Sub

Private Sub tombol_tambah()
On Error GoTo x
Tambah.Enabled = False
Hapus.Enabled = False
selesai.Enabled = False
simpan.Enabled = True
batal.Enabled = True
refres.Enabled = False
x:
End Sub

Private Sub tombol_batal()
On Error GoTo x
simpan.Enabled = False
batal.Enabled = False
Tambah.Enabled = True
Hapus.Enabled = True
selesai.Enabled = True
refres.Enabled = True
x:
End Sub

Private Sub warna_hidup()
On Error GoTo x
Text4.Enabled = True
DBCombo1.Enabled = True
DTPicker1.Enabled = True
Text4.BackColor = &H80000005
DBCombo1.BackColor = &H80000005
x:
End Sub
```

```
Private Sub warna_mati()
On Error GoTo x
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Text6.Enabled = False
DTPicker1.Enabled = False
DBCombo1.Enabled = False
Text1.BackColor = &H80000004
Text2.BackColor = &H80000004
Text3.BackColor = &H80000004
Text4.BackColor = &H80000004
Text6.BackColor = &H80000004
DBCombo1.BackColor = &H80000004
x:
End Sub

Private Sub tampilan()
On Error GoTo x
With Data2.Recordset
Text1.Text = !no_trans
DBCombo1.Text = !no_rek
DTPicker1.Value = !tgl_trans
Text4.Text = !jml_setor
DBGrid1.Refresh
With Data1.Recordset
Text6.Text = Val(Text3.Text)
Data1.Refresh
DBGrid1.Refresh
End With
End With
x:
End Sub

Private Sub dbcombo1_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   Text4.SetFocus
End If
x:
End Sub


Private Sub text1_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   DBCombo1.SetFocus
```

```
End If
x:
End Sub



Private Sub Text4_Change()
On Error GoTo x
On Error Resume Next
Text6.Text = Val(Text3.Text) - (Text4.Text)
x:
End Sub

Private Sub text4_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   simpan.SetFocus
End If
If Not IsNumeric(Chr(KeyAscii)) Then
   KeyAscii = 0
End If
x:
End Sub

Private Sub auto()
On Error GoTo x
Dim urutan As String * 5
Dim hitung As Byte
With Data2.Recordset
If .RecordCount = 0 Then
   urutan = "00001"
Else
   .MoveLast
   hitung = Val(Right(.Fields("no_trans"), 5)) + 1
   urutan = Right("00000" & hitung, 5)
End If
Text1 = urutan
End With
x:
End Sub
```

f) Desain Form Simpan

Kode Program :

```
Private Sub a()
On Error GoTo x
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 390 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 13) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 360 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 12) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 330 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 11) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 300 Then
        Data1.Recordset.edit
```

```
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 10) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 270 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 9) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 240 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 8) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 210 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 7) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 180 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 6) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 150 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 5) + _
```

```
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 120 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 4) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 90 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 3) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 60 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 2) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 30 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = _
        ((((Data3.Recordset!bunga_spn     /     100)     +     0.00000000023)     *
Data1.Recordset!saldo_spn) * 1) + _
        Data1.Recordset!saldo_spn
        Data1.Recordset.Update
Data5.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data5.Recordset!tgl_trans <= 30 Then
        Data1.Recordset.edit
        Data1.Recordset!saldo_spn = Data4.Recordset!sspn
        Data1.Recordset.Update
End If
End If
End If
End If
```

```
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
x:
End Sub

Private Sub c()
On Error GoTo x
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 390 Then
	Data1.Recordset.edit
	On Error Resume Next
	Data1.Recordset!no_rek = Data4.Recordset!no_rek
	Data1.Recordset!saldo_spn = Data4.Recordset!sspn
	Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 360 Then
	Data1.Recordset.edit
	On Error Resume Next
	Data1.Recordset!no_rek = Data4.Recordset!no_rek
	Data1.Recordset!saldo_spn = Data4.Recordset!sspn
	Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 330 Then
	Data1.Recordset.edit
	On Error Resume Next
	Data1.Recordset!no_rek = Data4.Recordset!no_rek
	Data1.Recordset!saldo_spn = Data4.Recordset!sspn
	Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 300 Then
	Data1.Recordset.edit
	On Error Resume Next
	Data1.Recordset!no_rek = Data4.Recordset!no_rek
```

```
        Data1.Recordset!saldo_spn = Data4.Recordset!sspn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 270 Then
        Data1.Recordset.edit
        On Error Resume Next
        Data1.Recordset!no_rek = Data4.Recordset!no_rek
        Data1.Recordset!saldo_spn = Data4.Recordset!sspn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 240 Then
        Data1.Recordset.edit
        On Error Resume Next
        Data1.Recordset!no_rek = Data4.Recordset!no_rek
        Data1.Recordset!saldo_spn = Data4.Recordset!sspn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 210 Then
        Data1.Recordset.edit
        On Error Resume Next
        Data1.Recordset!no_rek = Data4.Recordset!no_rek
        Data1.Recordset!saldo_spn = Data4.Recordset!sspn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 180 Then
        Data1.Recordset.edit
        On Error Resume Next
        Data1.Recordset!no_rek = Data4.Recordset!no_rek
        Data1.Recordset!saldo_spn = Data4.Recordset!sspn
        Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 150 Then
        Data1.Recordset.edit
        On Error Resume Next
        Data1.Recordset!no_rek = Data4.Recordset!no_rek
        Data1.Recordset!saldo_spn = Data4.Recordset!sspn
        Data1.Recordset.Update
Else
```

```
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 120 Then
	Data1.Recordset.edit
	On Error Resume Next
	Data1.Recordset!no_rek = Data4.Recordset!no_rek
	Data1.Recordset!saldo_spn = Data4.Recordset!sspn
	Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 90 Then
	Data1.Recordset.edit
	On Error Resume Next
	Data1.Recordset!no_rek = Data4.Recordset!no_rek
	Data1.Recordset!saldo_spn = Data4.Recordset!sspn
	Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 60 Then
	Data1.Recordset.edit
	On Error Resume Next
	Data1.Recordset!no_rek = Data4.Recordset!no_rek
	Data1.Recordset!saldo_spn = Data4.Recordset!sspn
	Data1.Recordset.Update
Else
Data2.Recordset.MoveLast
Data4.Recordset.MoveLast
If Date - Data2.Recordset!tgl_trans >= 30 Then
	Data1.Recordset.edit
	On Error Resume Next
	Data1.Recordset!no_rek = Data4.Recordset!no_rek
	Data1.Recordset!saldo_spn = Data4.Recordset!sspn
	Data1.Recordset.Update
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
End If
```

```
x:
End Sub

Private Sub Form_Unload(Cancel As Integer)
c
End Sub

Private Sub Form_Activate()
On Error GoTo x
Data1.Recordset.Index = "xnasabah"
Data2.Recordset.Index = "xsimpan"
On Error Resume Next
Data2.Recordset.MoveLast
a
x:
End Sub

Private Sub DBCombo1_Change()
On Error GoTo x
With Data1.Recordset
Data1.Recordset.Index = "xnasabah"
Data1.Recordset.Seek "=", DBCombo1.Text
If Not .NoMatch Then
   Text2.Text = !nama
   On Error Resume Next
   Text3.Text = !saldo_spn

Else
End If
End With
Data1.Recordset.Index = "xnasabah"
x:
End Sub

Private Sub DBGrid1_RowColChange(LastRow As Variant, ByVal LastCol
As Integer)
On Error GoTo x
tampilan
x:
End Sub

Private Sub hapus_Click()
On Error GoTo x
If Text1.Text = "" Then
   MsgBox "Klik Data yang Akan Dihapus Dulu Di Grid ??", vbOKOnly,
"pesan"
   DBGrid1.Enabled = True
   DBGrid1.AllowDelete = True
```

```
Else
    With Data2.Recordset
    If Not .NoMatch Then
      Pesan = MsgBox("Yakin Akan Menghapus data Simpan tersebut..??",
vbYesNo, "pesan")
       On Error Resume Next
       Data1.Recordset.Index = "xspn"
       Data1.Recordset.Seek "=", Text3.Text
       If Pesan = vbYes Then
       pan
          .Delete
          With Data4.Recordset
          If Not .NoMatch Then
          .Delete
          End If
          End With
          On Error Resume Next
          If Not Data1.Recordset.NoMatch Then
          Data1.Recordset.edit
          Data1.Recordset!saldo_spn = (Val(Text3.Text) - Val(Text4.Text))
          Data1.Recordset.Update
          Data1.Refresh
           bersih
          End If
          Data1.Refresh
          Data2.Refresh
          DBGrid1.Refresh
          kosong
       End If
    End If
    End With
    End If
On Error GoTo 0
On Error Resume Next
Data2.Recordset.MoveLast
x:
End Sub

Private Sub bersih()
On Error GoTo x
       Data4.Recordset.Index = "xspn1"
       Data4.Recordset.Seek "=", Text3.Text
       If Not Data4.Recordset.NoMatch Then
          Data4.Recordset.edit
          Data4.Recordset!sspn = (Val(Text3.Text) - Val(Text4.Text))
          Data4.Recordset.Update
          Data4.Refresh
       End If
```

```
        Data4.Refresh
        Data4.Refresh
        DBGrid1.Refresh
        kosong
x:
End Sub

Private Sub refres_Click()
On Error GoTo x
FORM_SIMPAN.Refresh
Data1.Refresh
Data2.Refresh
DBGrid1.Refresh
On Error Resume Next
Data2.Recordset.MoveLast
x:
End Sub

Private Sub selesai_Click()
Unload Me
End Sub

Private Sub Tambah_Click()
On Error GoTo x
kosong
auto
tombol_tambah
warna_hidup
DBCombo1.SetFocus
x:
End Sub

Private Sub batal_Click()
On Error GoTo x
kosong
tombol_batal
warna_mati
simpan.Visible = True
tampilan
x:
End Sub

Private Sub Form_Load()
On Error GoTo x
mati
warna_mati
kosong
x:
```

```
End Sub
Private Sub coba()
On Error GoTo x
Data4.Recordset.Index = "xspn2"
Data4.Recordset.Seek "=", Text3.Text
   If Data4.Recordset.NoMatch Then
      Data4.Recordset.AddNew
      Data4.Recordset!no_rek = DBCombo1.Text
      Data4.Recordset!sspn = Val(Text3.Text) + Val(Text4.Text)
      Data4.Recordset.Update
     ElseIf Not Data4.Recordset.NoMatch Then
         Data4.Recordset.edit
         Data4.Recordset!no_rek = DBCombo1.Text
         Data4.Recordset!sspn = Val(Text3.Text) + Val(Text4.Text)
         Data4.Recordset.Update
         Data4.Refresh
      End If

      DBGrid1.Refresh
 x:
End Sub

Private Sub pan()
On Error GoTo x
Dim i As Integer, panbar(32000) As String
ProgressBar1.Min = LBound(panbar)
ProgressBar1.Max = UBound(panbar)
ProgressBar1.Visible = True
ProgressBar1.Value = ProgressBar1.Min
For i = LBound(panbar) To UBound(panbar)
   ProgressBar1.Value = i
Next i
ProgressBar1.Visible = False
ProgressBar1.Value = ProgressBar1.Min
x:
End Sub

Private Sub simpan_Click()
On Error GoTo x
If Text1.Text = "" Or DBCombo1.Text = "" Or Text4.Text = "" Then
   MsgBox "Silahkan Lengkapi Datanya!!", vbOKOnly + vbExclamation,
"PERINGATAN"
   If Text1.Text = "" Then
      Text1.SetFocus
   ElseIf DBCombo1.Text = "" Then
     DBCombo1.SetFocus
   ElseIf Text4.Text = "" Then
      Text4.SetFocus
```

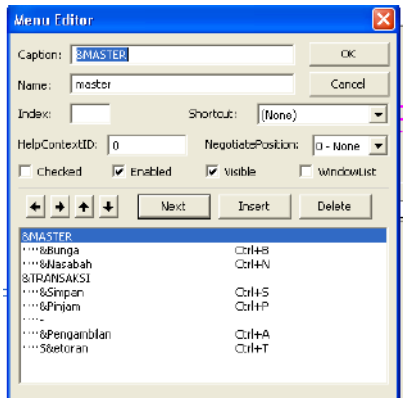
```
   End If
Else
pan
   Data2.Recordset.Index = "xsimpan"
   Data2.Recordset.Seek "=", Text1.Text
   If Data2.Recordset.NoMatch Then
      Data2.Recordset.AddNew
      Data2.Recordset!no_trans = Text1.Text
      Data2.Recordset!tgl_trans = DTPicker1.Value
      Data2.Recordset!no_rek = DBCombo1.Text
      Data2.Recordset!jml_simpan = Text4.Text
      Data1.Recordset.Index = "xspn"
      Data1.Recordset.Seek "=", Text3.Text
      If Data1.Recordset.NoMatch Then
         Data1.Recordset.AddNew
         Data1.Recordset!saldo_spn = Text4.Text
         Data1.Recordset.Update
         Data1.Refresh
      Else
      If Not Data1.Recordset.NoMatch Then
         Data1.Recordset.edit
         Data1.Recordset!saldo_spn = Val(Text3.Text) + Val(Text4.Text)
         Data1.Recordset.Update
         Data1.Refresh
      End If
      End If
      Data2.Recordset.Update
      DBGrid1.Refresh
   Else
      MsgBox " Maaf No transaksi sudah Ada !", vbOKOnly + vbCritical,
"KESALAHAN"
      simpan.Enabled = False
   End If
coba
Data2.Recordset.Index = "xsimpan"
On Error Resume Next
Data2.Recordset.MoveLast
tombol_simpan
warna_mati
Tambah.SetFocus
End If
x:
End Sub

Private Sub kosong()
On Error GoTo x
Text1.Text = ""
Text2.Text = ""
```

```
Text3.Text = ""
Text4.Text = ""
Text6.Text = ""
DBCombo1.Text = ""
DTPicker1.Value = Date
x:
End Sub

Private Sub mati()
On Error GoTo x
simpan.Enabled = False
batal.Enabled = False
x:
End Sub

Private Sub tombol_simpan()
On Error GoTo x
simpan.Enabled = False
batal.Enabled = False
Tambah.Enabled = True
Hapus.Enabled = True
selesai.Enabled = True
refres.Enabled = True
x:
End Sub

Private Sub tombol_tambah()
On Error GoTo x
Tambah.Enabled = False
Hapus.Enabled = False
selesai.Enabled = False
simpan.Enabled = True
batal.Enabled = True
refres.Enabled = False
x:
End Sub

Private Sub tombol_batal()
On Error GoTo x
simpan.Enabled = False
batal.Enabled = False
Tambah.Enabled = True
Hapus.Enabled = True
selesai.Enabled = True
refres.Enabled = True
x:
End Sub
```

```
Private Sub warna_hidup()
On Error GoTo x
Text4.Enabled = True
DBCombo1.Enabled = True
DTPicker1.Enabled = True
Text4.BackColor = &H80000005
DBCombo1.BackColor = &H80000005
x:
End Sub

Private Sub warna_mati()
On Error GoTo x
Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Text6.Enabled = False
DTPicker1.Enabled = False
DBCombo1.Enabled = False
Text1.BackColor = &H80000004
Text2.BackColor = &H80000004
Text3.BackColor = &H80000004
Text4.BackColor = &H80000004
Text6.BackColor = &H80000004
DBCombo1.BackColor = &H80000004
x:
End Sub

Private Sub tampilan()
On Error GoTo x
With Data2.Recordset
Text1.Text = !no_trans
DBCombo1.Text = !no_rek
DTPicker1.Value = !tgl_trans
Text4.Text = !jml_simpan
DBGrid1.Refresh
With Data1.Recordset
Text6.Text = Val(Text3.Text)
Data1.Refresh
DBGrid1.Refresh
End With
End With
x:
End Sub
```

```
Private Sub dbcombo1_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   Text4.SetFocus
End If
x:
End Sub
Private Sub text1_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   DBCombo1.SetFocus
End If
x:
End Sub

Private Sub text4_keypress(KeyAscii As Integer)
On Error GoTo x
If KeyAscii = 13 Then
   simpan.SetFocus
End If
If Not IsNumeric(Chr(KeyAscii)) Then
   KeyAscii = 0
End If
x:
End Sub

Private Sub Text4_Change()
On Error GoTo x
Text6.Text = Val(Text3.Text) + Val(Text4.Text)
x:
End Sub

Private Sub auto()
On Error GoTo x
Dim urutan As String * 5
Dim hitung As Byte
With Data2.Recordset
If .RecordCount = 0 Then
   urutan = "00001"
Else
   .MoveLast
   hitung = Val(Right(.Fields("no_trans"), 5)) + 1
   urutan = Right("00000" & hitung, 5)
End If
Text1 = urutan
End With
x:
End Sub
```

g) Desain MDI Form menu utama



Kode Program :

```
Private Sub backup_Click()
FORM_BACK_UP.Show
End Sub

Private Sub bunga_Click()
FORM_BUNGA.Show
End Sub

Private Sub exit_Click()
tanya = MsgBox(" Yakin Anda akan Keluar dari Program Ini...?", 4 + 32 +
256, "KONFIRMASI")
If tanya = 6 Then
End
End If
End Sub

Private Sub gantipassword_Click()
GANTI_PASSWORD.Show
End Sub
```

```
    Private Sub lapnasabah_Click()
    LAP_NASABAH.Show
    End Sub

    Private Sub lappengambilan_Click()
    LAP_PENGAMBILAN.Show
    End Sub

    Private Sub lappinjam_Click()
    LAP_PINJAM.Show
    End Sub
    Private Sub lapsetoran_Click()
    LAP_SETORAN.Show
    End Sub

    Private Sub lapsimpan_Click()
    LAP_SIMPAN.Show
    End Sub

    Private Sub nasabah_Click()
    FORM_NASABAH.Show
    End Sub

    Private Sub pengambilan_Click()
    FORM_PENGAMBILAN.Show
    End Sub

    Private Sub pinjam_Click()
    FORM_PINJAM.Show
    End Sub

    Private Sub setoran_Click()
    FORM_SETORAN.Show
    End Sub

    Private Sub simpan_Click()
    FORM_SIMPAN.Show
    End Sub

    Private Sub Timer1_Timer()
    Label4.Left = Label4.Left - 180
    If Label4.Left <= -29000 Then
       Timer1.Enabled = True
       Label4.Left = "14000"
    End If
    End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
```

```
Select Case Button.Key
   Case "a"
      FORM_BUNGA.Show
   Case "b"
      FORM_NASABAH.Show
   Case "c"
      FORM_SIMPAN.Show
   Case "d"
      FORM_PINJAM.Show
   Case "e"
      FORM_PENGAMBILAN.Show
   Case "f"
     FORM_SETORAN.Show
      LAP_SETORAN.Show
End Select
End Sub
```

Daftar Pustaka

Subari & Yuswanto, 2008. ”Panduan Lengkap Pemrograman Visual Basic 6.0”, Jakarta: Cerdas Pustaka Publisher.

Tim Penyusun. 2006. Modul Praktek Laboratorium Komputer Visual Basic 6.0. Jakarta. AMIK Bina Sarana Informatika.

Tim Penyusun. 2002. Modul Praktikum Visual Basic. Yogyakarta. FMIPA Universitas Gadjah Mada.

http://dwi.its-sby.edu/PIKTI/Visual%20BAsic/last/Bab1.doc

http://ikc.vip.net.id/berseri/krisna-vb6/index.php

http://ns1.cic.ac.id/~marsani.asfi/tulisan/Modul_1.htm

http://leo.apeaje.info/vb/tutor.html

http://ns1.cic.ac.id/~marsani.asfi/tulisan/Modul_1.htm

http://www.ajibsusanto.site88.net/aplikasi_bisnis/MODUL_VB_6.pdf

http://www.masinosinaga.com/index.php?name=News&file=article&sid=234

http://absanka.wordpress.com/2008/07/05/pdf-program-visual-basic/

http://www.kuliahit.com/kuliahit/article/16/Fungsi-Date-and-Time-pada-VB

http://ikc.cbn.net.id/berseri/krisna-vb6/krisna-vb6-06.zip

http://leo.apeaje.info/vb/lesson8.html

http://mercusian.com/visual-basic/praktek-visual-basic-database-dao.html

http://mugi.or.id/blogs/elang/archive/2008/08/15/penggunakan-dao-data-access-objects.aspx

http://lecturer.eepis-its.edu/~tessy/tutorial/bab5.pdf

Profil Penulis

Eko Purwanto, Lahir di Klaten 27 Juni 1984, menyelesaikan pendidikan SDN 2 Sidoluas (1997), SLTPN 2 Tulung (2000), SMUN 1 Karanfganom (2003) dan menyelesaikan S1 di STIMIK Duta Bangsa Surakarta (2008).

Menikah dengan Sutatik dan telah di karuniai Seorang putra bernama Fawzi Muhammad Afdhal. Saai ini sebagai Dosen di Almamaternya STIMIK Duta Bangsa Surakarta sejak tahun 2008. Sedang menempuh Strata Dua (S2) di STIMIK AMIKOM Yogyakarta.

Penulis telah menerbitkan buku yang sudah ia tulis yaitu Bahasa Pemograman Pascal Tahun 2011.

Pemrograman Visual

Microsoft Visual Basic 6.0

Buku Pemrograman Visual dengan Visual Basic 6.0 untuk memperkenalkan Lingkungan Visual Basic, pembuatan interface dengan menggunakan tools yang ada serta mengimplementasikan program sederhana dengan bahasa Visual Basic. Disamping itu juga memahami struktur dan perintah-perintah program bahasa Visual Basic.

Buku pemrograman Visual dengan Visual Basic 6.0 ini terdiri dari 13 Bab sesuai dengan sistem perkuliahan yang ada di Perguruan Tinggi untuk satu semester. Pada setiap pertemuan diberikan contoh-contoh program dan latihannya, diharapkan dengan mencoba contoh program yang ada dan mengerjakan latihannya, mahasiswa lebih mudah untuk memahami materi yang diberikan. Buku ini diharapkan bisa membantu mahasiswa dan pembaca dalam belajar komputer, khususnya untuk pemrograman Visual Basic.

DPI
DUTA PUBLISHING INDONESIA
DUTA PUBLISHING INDONESIA
JL. PULANGGENI NO. 8 SOLO TELP. (0271) - 712432

ISBN 602934815-6
9 786029 348156