

Pertemuan 1

Pengenalan Bahasa Pemrograman

Visual Basic

Tujuan Praktikum :

1. Mengenalkan konsep bahasa pemrograman
2. Mengenalkan konsep bahasa pemrograman Visual Basic
3. Menjelaskan fungsi-fungsi komponen dalam Visual Basic
4. menjelaskan proses perancangan program pada Visual Basic.

Target Praktikum :

1. Praktikan memahami konsep bahasa pemrograman
2. praktikan dapat merancang program sederhana dengan Visual Basic.

Bahasa Pemrograman

Bahasa pemrograman adalah instruksi-instruksi yang dengan aturan tata bahasa tertentu yang dicompile kemudian dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Sampai sekarang terdapat puluhan bahasa pemrograman. Ada bahasa rakitan (assembly), Fortran, Cobol, Ada, PL/I, Algol, Pascal, Basic, Prolog, LISP, PRG, kemudian ada juga bahasa simulasi seperti CSMP, Simscript, GPSS, Dinamo, dan masih banyak lagi. Berdasarkan terapannya, bahasa pemrograman dapat digolongkan atas dua kelompok besar :

1. Bahasa pemrograman bertujuan khusus
Yang termasuk dalam kelompok ini adalah Cobol (untuk terapan bisnis dan administrasi), Fortran(untuk komputasi ilmiah), bahasa rakitan (untuk pemrograman mesin), prolog(terapan kecerdasan buatan), bahasa-bahasa simulasi, dan sebagainya.
2. Bahasa pemrograman bertujuan umum
Yang termasuk dalam kelompok ini adalah bahasa Pascal, Basic, dan C. Tentu saja pembagian ini tidak kaku. Bahasa-bahasa bertujuan khusus tidak berarti tidak bisa digunakan untuk aplikasi lain. Cobol misalnya, dapat juga digunakan untuk terapan ilmiah, hanya saja kemampuannya terbatas. Yang jelas, bahasa-bahasa pemrograman yang berbeda dikembangkan untuk bermacam-macam terapan yang berbeda pula.

Berdasarkan pada apakah notasi bahasa pemrograman lebih “dekat” ke mesin atau ke bahasa manusia, maka bahasa pemrograman dikelompokkan atas dua macam :

1. *Bahasa tingkat rendah*

Bahasa jenis ini dirancang agar setiap instruksinya langsung dikerjakan komputer tanpa harus melalui penerjemah (translator). Contohnya adalah bahasa mesin

2. *Bahasa tingkat tinggi*

Bahasa tingkat tinggi membuat pemrograman lebih mudah dipahami, lebih “manusiawi”, dan berorientasi ke bahasa manusia (bahasa Inggris). Yang termasuk dalam bahasa ini adalah Pascal, PL/I, Ada, Cobol, Basic, Fortran, C dan sebagainya.

Bahasa pemrograman Visual Basic

Bahasa Visual Basic adalah salah satu di antara sekian banyak bahasa pemrograman yang cukup banyak digunakan oleh para programmer. Visual Basic adalah bahasa pemrograman visual dengan dasar pemrograman bahasa Basic. Bahasa BASIC (Beginner's All-purpose Symbolic Instruction Code) dikembangkan pertama kali pada awal 1950-an. Sementara bahasa Visual Basic, yang merupakan pengembangan bahasa basic, dikembangkan pertama kali oleh Microsoft pada tahun 1991.

Visual Basic ini merupakan salah satu Development Tool yaitu alat bantu yang dapat digunakan untuk membuat berbagai macam program komputer, khususnya yang berbasis Windows.

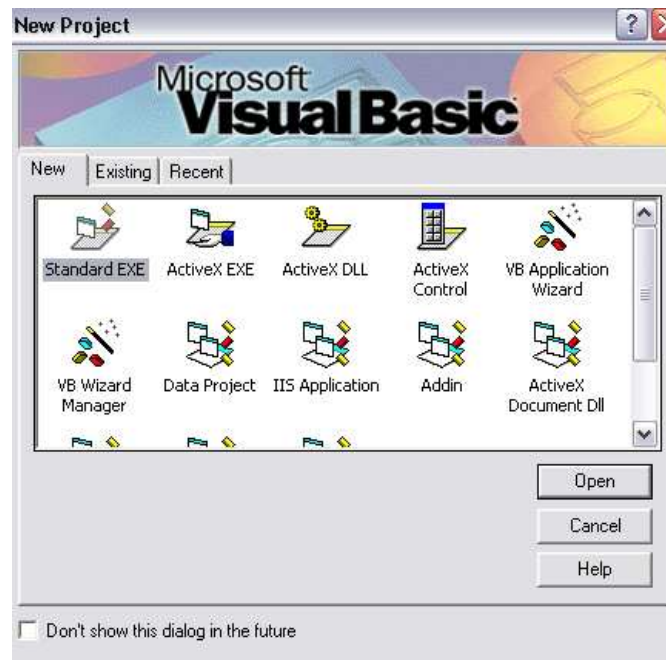
Pembuatan program dalam bahasa Visual Basic berbeda dengan pembuatan program-program DOS umumnya atau yang sering disebut sebagai pemrograman konvensional. Namun jika dilihat dari visualnya, visual Basic sedikit mirip dengan Delphi. Hanya saja bahasa Visual Basic merupakan salah satu bahasa pemrograman komputer yang mendukung object (*Object Oriented Programming = OOP*), sementara bahasa Delphi memiliki basic bahasa Pascal yang terstruktur.

Pembuatan aplikasi dengan menggunakan bahasa Visual Basic dimulai dengan memperkirakan kebutuhan, kemudian merancang tampilan program yang diikuti dengan pembuatan kode program tersebut.

Mengenal Integrated Development Environment (IDE) VB 6

Untuk memulai Visual Basic, klik tombol **Start > Start > Programs > Microsoft Visual Studio 6.0 > Microsoft Visual Studio 6.0**

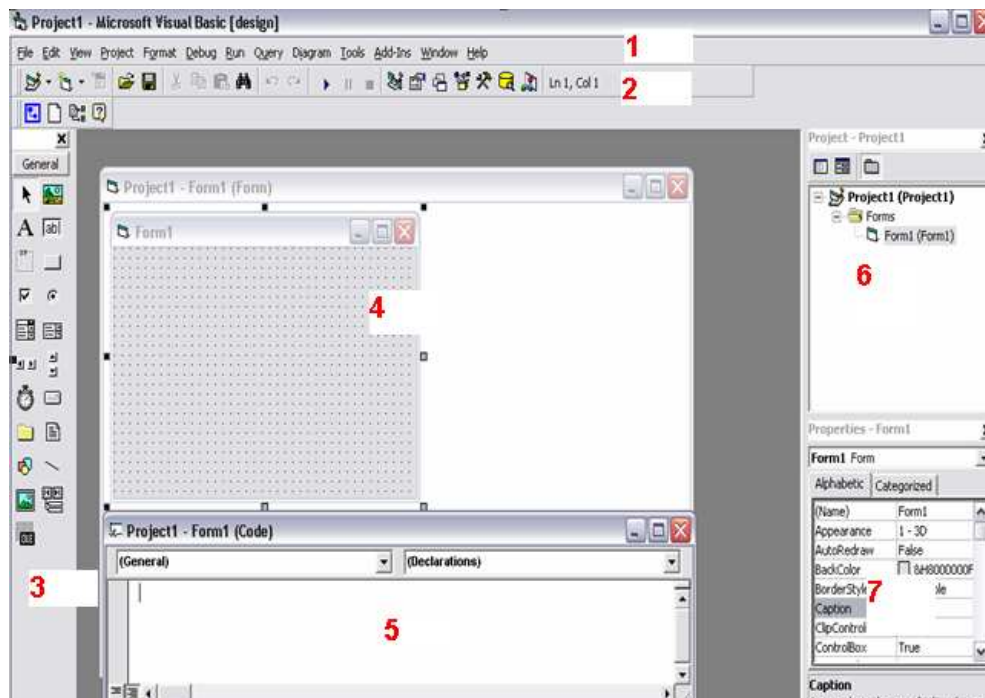
Setelah diaktifkan, akan muncul tampilan seperti ditunjukkan gambar 1.1 .



Gambar 1.1 Tampilan New project

Pilih Standar **EXE** , klik 2 kali atau tekan tombol **Open**.

Maka akan ditampilkan jendela IDE arau are kerja MS VB 6.0 seperti gambar 1.2 :



Gambar 1.2 Tampilan jendela Microsoft Visual Basid 6.0

Keterangan :

1. Menubar

Pada bagian ini terdapat menu-menu dasar yang akan digunakan selama perancangan program. Ada Menu File (untuk membuka program, menyimpan program, dll), menu View (untuk menampilkan tool-tool yang belum ada), Menu Run (untuk kompilasi), dsb.

2. Toolbar

Pada bagian ini ditampilkan icon-icon dari menu utama yang ada pada menubar.

3. Toolbox

Digunakan untuk pemilihan kontrol-kontrol yang akan digunakan oleh program yang akan dirancang. Setiap kontrol diwakili oleh sebuah icon toolbox.

4. Jendela Form

Form adalah area tampilan yang berhubungan dengan sebuah jendela yang dapat dilihat ketika aplikasi berjalan. Pada form ini dapat diletakkan kontrol dan kode untuk pembuatan program. Form akan banyak digunakan saat perancangan.

5. Jendela Code

Jendela Code digunakan untuk menuliskan source code dari program yang kita buat.

6. Project Explorer

Project Explorer akan menampilkan form-form dan modul yang ada di dalam program yang kita buat. Dengan project explorer kita dapat berpindah-pindah dari satu form ke form yang lain.

7. Jendela Properties

Digunakan untuk mengatur sifat(properti) dari form atau kontrol-kontrol. Isi dari window properties ini dapat berubah-ubah sesuai dengan form atau kontrol yang dipilih.

Komponen MS-Visual Basic 6.0

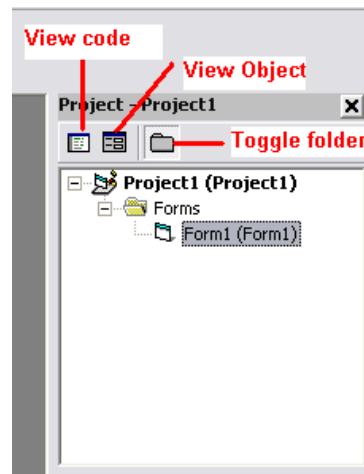
Agar bisa menguasai bahasa pemrograman Visual Basic, kita juga harus mempelajari dasarnya dengan lebih baik dan kuat. Beberapa komponen utama dalam bahasa visual yang harus kita ketahui diantaranya :

Project

Project adalah sekumpulan modul. Jadi project (proyek) adalah program aplikasi itu sendiri. Di dalamnya terdapat form beserta code nya. Project ini disimpan dalam file berakhiran .vbp. File ini akan menyimpan seluruh komponen program, termasuk pilihan proyek, pilihan environment, pilihan file EXE dan segala sesuatu yang berhubungan dengan proyek.

Pada jendela proyek terdapat tiga icon, yaitu icon **View Code**, icon **View Object** dan icon **Toggle Folder**. Icon **View Code** dipakai untuk menampilkan jendela editor kode program. Icon **View Object**

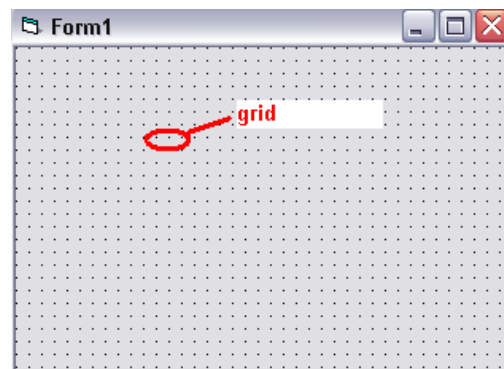
dipakai untuk menampilkan bentuk formulir (form) dan icon **Toggle Folders** berguna untuk menampilkan folder (tempat penyimpanan folder).



Gambar 1.3 jendela project

Form

Form adalah object yang dipakai sebagai tempat bekerja program aplikasi. Jadi bisa dikatakan di form inilah kita melakukan aktifitas perancangan program. Biasanya pada saat mendesain Form, terdapat garis titik-titik yang disebut grid. Grid sangat berguna untuk membantu pengaturan tata letak objek yang dimasukkan dalam form, karena gerakan penunjuk mouse akan sesuai (tepat) pada titik-titik grid.

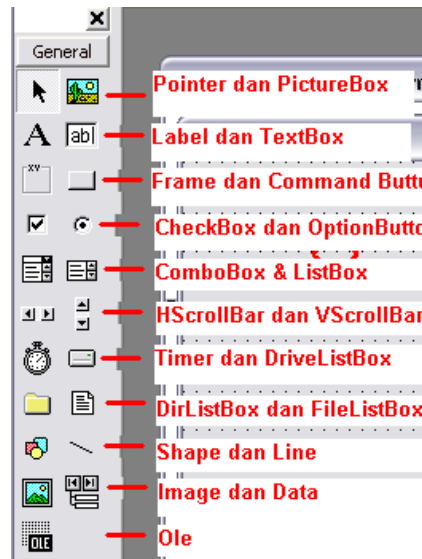


Gambar 1.4 Form

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 mengklik kanan pada toolbox, lalu memilih Components atau Add Tab.

Icon-icon pada toolbox ini seringkali disebut sebagai control. Setiap kontrol memiliki properti seperti warna, tulisan yang diletakkan pada control dan propeerti lainnya. Control-control diletakkan pada toolbox untuk dipilih dan digambarkan pada form. Beberapa control yang ada pada toolbox ditunjukkan oleh gambar 1.5 .



Gambar 1.5 Toolbox

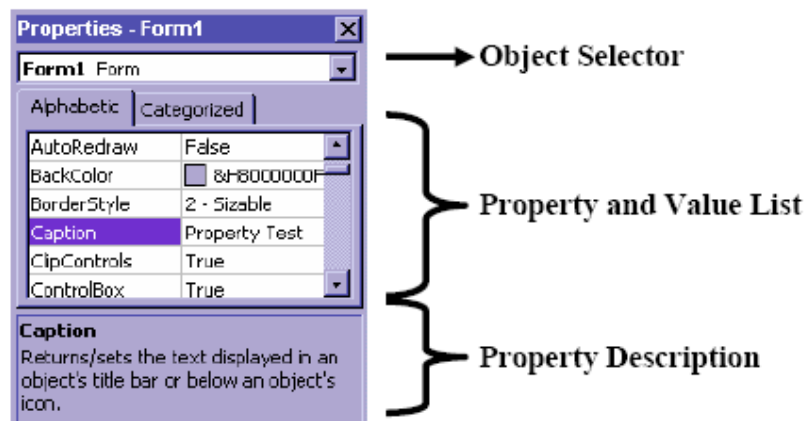
Keterangan :

Nama Control	Penjelasan
Label	Untuk menampilkan tulisan pada form. Pemakai tidak dapat mengubah tulisan tersebut secara langsung
TextBox	Digunakan sebagai tempat input/juga dapat digunakan untuk menampilkan teks, dan pemakai dapat mengubah-ubah tulisan yang terdapat pada control ini.
ListBox	Menampilkan beberapa item, dan dari control ini item-item tersebut dapat dipilih. Scrollbar dapat digunakan untuk enggulung pilihan yangtidak dapat ditampilkan seluruhnya.
ComboBox	Merupakan kombinasi antara TextBox dan ListBox. Sehingga pemilihan item dapat dilakukan dari ListBox, atau dengan mengetik langsung pada TextBox.
CommandButton	Menjalankan suatu tindakan jika pemakai telah melakukan pilihan dengan menekan control ini.
Frame	Mengelompokkan control-control secara visual (tergambar) atau

	secara fungsional(tindakan).
CheckBox	Menampilkan keadaan True/False atau pilihan Yes/No. beberapa control ini dapat memiliki keadaan yang sama pada suatu saat.
OptionButton	Control ini sama dengan control CheckBox. Perbedaannya hanya satu control dari beberapa control ini dapat di-on-kan. Jika control dengan tipe yang sama lainnya di-on-kan, maka control tipe ini yang sebelumnya on akan di-off-kan.
PictureBox	Digunakan untuk menampilkan gambar

Properties

Propoerti digunakan untuk menentukan setting suatu objek. Suatu objek biasanya mempunyai beberapa properti yang dapat diatur langsung dari jendela Properties atau lewat kode program. Setting properti akan menentukan cara kerja dari objek yang bersangkutan saat program aplikasi dijalankan, misalnya menentukan warna objek, bingkai obejk, pengambilan data dan lain-lain.



Gambar 1.6 Jendela properties

Kode program

Kode program adalah serangkaian tulisan perintah yang akan dilaksanakan jika suatu Objek dijalankan. Kode program ini akan mengontrol dan menentukan jalannya suatu objek.

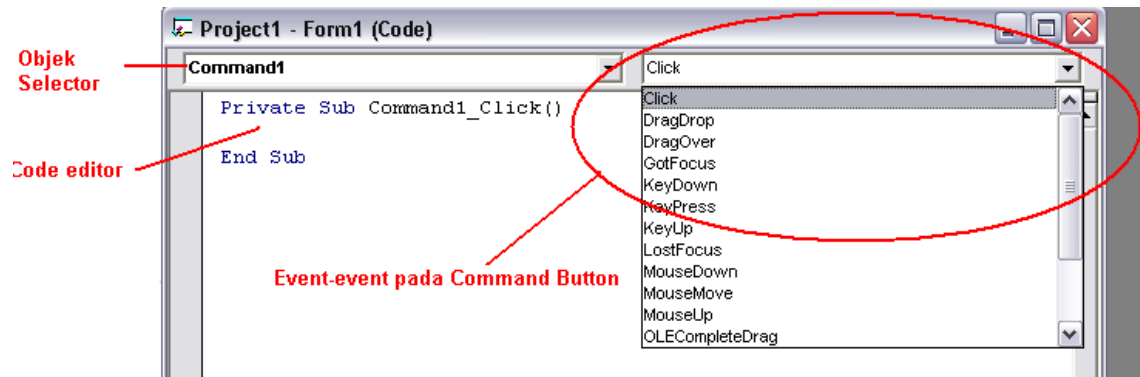
Event

Event adalah peristiwa atau kejadian yang diterima oleh suatu objek, misalnya klik, seret, tunjuk, dan lain-lain. Event yang diterima objek akan memicu MS VB menjalankan kode program yang ada di dalamnya.

Contoh:

```
Private Sub Command1_Click()
```

Baris diatas menunjukkan penggunaan event Click pada object Command1, yang berarti jika kita mengklik objek Command1, maka baris-baris kode program dibawahnya akan dilaksanakan. Tampilan Jendela code dengan event ditunjukkan oleh Gambar 1.7



Gambar 1.7 Jendela Kode

Penjelasan Masing-masing Event

Berikut beberapa event yang paling sering digunakan :

- ✓ Event untuk **mouse**

Click

Event ini akan terjadi apabila seorang pemakai program menekan tombol kiri mouse lalu melepaskannya dengan cepat saat pointer mouse ada diatas objek yang mempunyai event tersebut

MouseDown

Event ini akan terjadi jika pemakai menekan dan menahan tombol mouse

MouseUp

Event ini terjadi bila pemakai melepaskan tombol mouse

MouseMove

Event ini terjadi bila pemakai menggeser mouse.

DragDrop

Event ini terjadi pada objek apabila program menekan dan menahan tombol kiri mouse lalu menyeret objek dari satu tempat ke tempat lain, baru kemudian melepaskan tombol kiri mouse

DragOver

Event ini akan terjadi pada objek apabila pemakai menekan dan menahan tombol kiri mouse lalu menyeret objek melewati objek tersebut.

✓ Event untuk **keyboard**

KeyPress

Event ini akan terjadi pada objek saat pemakai menekan sebuah tombol keyboard lalu melepaskannya.

KeyDown

Event ini akan terjadi pada objek saat pemakai menekan dan menahan sebuah tombol keyboard

KeyUp

Event ini akan terjadi pada objek saat pemakai melepaskan sebuah tombol keyboard

Metoda(method)

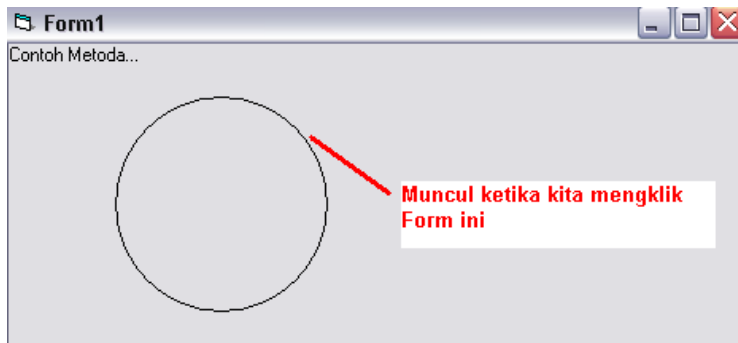
Metoda adalah suatu set perintah seperti halnya fungsi dan prosedur, tetapi sudah tersedia di dalam suatu objek. Seperti halnya properti (yang juga terdapat pada suatu objek), suatu metoda dapat dipanggil dengan menyebut nama objek diikuti tanda titik dan nam metodenya. Metoda biasanya akan mengerjakan suatu tugas khusus pada suatu objek tertentu, sedangkan properti biasanya memberi definisi nilai atau setting pada objek

Contoh :

Buatlah proyek baru dengan memilih **File** lalu **New** dilanjutkan dengan memilih icon **VB Enterprise Control**. Setelah itu lakukan langkah-langkah berikut :

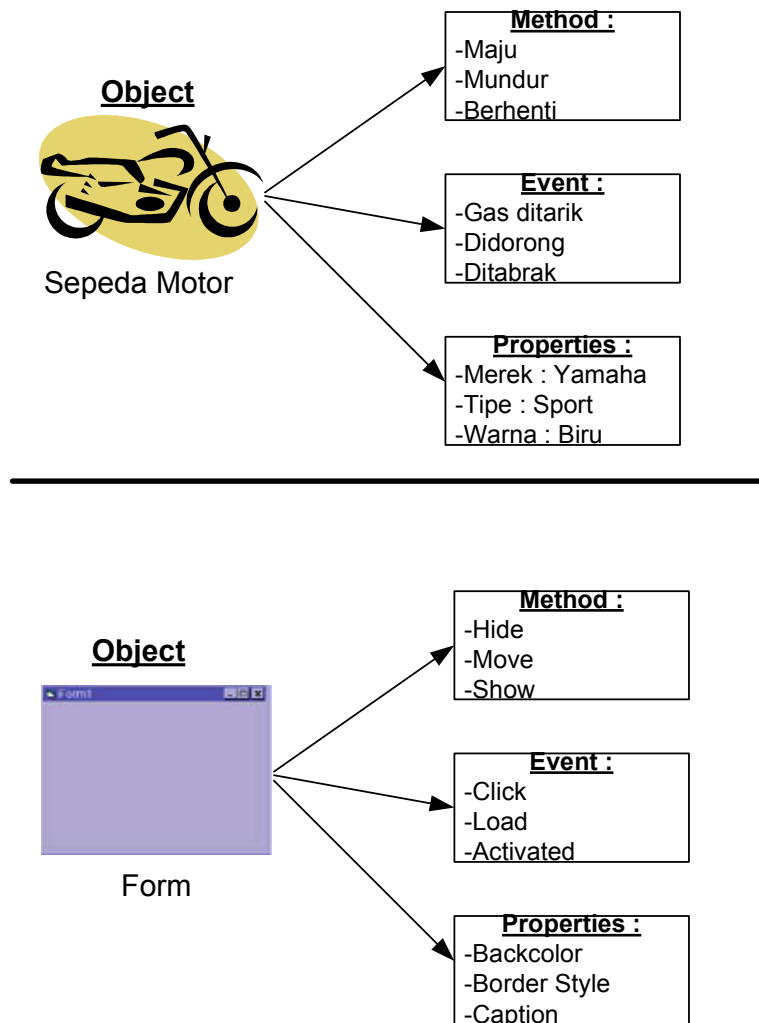
1. Klik ganda tulisan **Forms** pada jendela proyek, kemudian klik ganda tulisan form1, sehingga muncul tampilan Form1 di layar.
2. Klik icon **view Code**
3. Pada bagian code, klik tulisan general/form dan pilih Form. Otomatis akan muncul procedure Private Sub Form_load().
4. Pilih **Active** pada jendela procedure.
5. Ketik tulisan **form1.print "Coba Metoda.."**
6. Klik jendela procedure dan pilih **Click**
7. Ketik tulisan Form1.Circle(1000,1500), 500 yang berarti kita menggunakan metoda Circle pada Objek Form1 untuk membuat lingkaran.
8. Simpan form dan proyek dengan nama Latihan

9. Jalankan proyek dengan menekan **Run**, sehingga akan muncul hasilnya seperti ditunjukkan Gambar 1.8



Gambar 1.8 Contoh Metoda

Ilustrasi yang menunjukkan hubungan antara Objek, Event, Property dan metode ditunjukkan oleh Gambar 1.9.



Gambar 1.9 Ilustrasi Object, method, property dan Event

Dari ilustrasi bisa dilihat bahwa bahasa pemrograman Visual Basic adalah bahasa yang berorientasi Object (OOP)

Module

Module dapat disejajarkan dengan form, tetapi tidak mengandung objek dan bentuk standar. Module dapat berisi beberapa kode program atau prosedur yang dapat digunakan dalam program aplikasi.

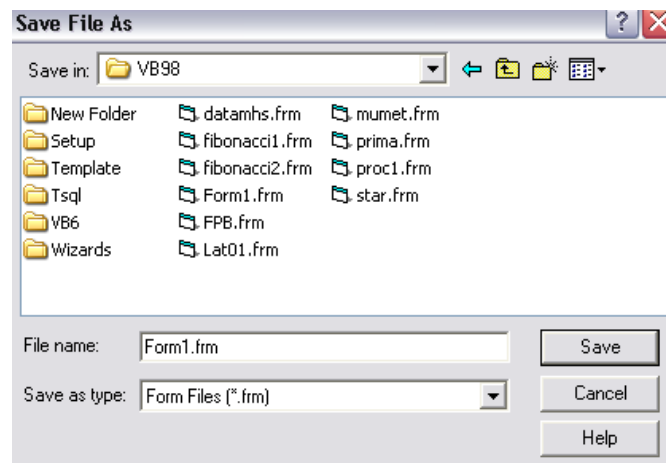
Untuk membuat modul sendiri, kita dapat memasukkan ke dalam program aplikasi dengan menu **Project** lalu **Add Module**.

Penyimpanan dan Kompilasi program

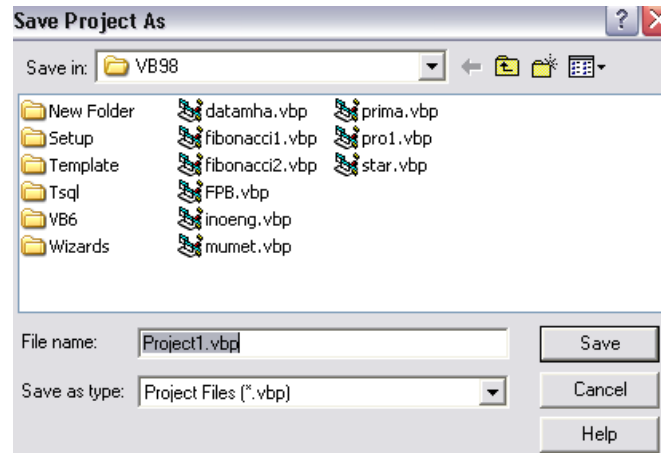
Sebuah proyek pada Visual Basic terdiri atas beberapa file. Jika progra semakin kompleks, maka program akan terdiri atas beberapa form. Form-form itu diletakkan pada file terpisah. Itulah yang kemudain menyebabkan saat program disimpan, proses penyimpanan akan membuat 2 file, yaitu file untuk menyimpan proyek dan sebuah file untuk menyimpan form. File proyek diberi nama ***.vbproj**, sedangkan file form diberi extention ***.frm**.

Penyimpanan proyek dapat dilakukan dengan langkah-langkah sebagai berikut :

1. Jalankan perintah **File | Save project**, maka akan ditampilkan kotak dialog seperti ditunjukkan gambar 1.10.
2. Kita akan diminta untuk memasukkan nama file untuk form ini. Ingat ...Nama file tidak ada kaitannya dengan nama form.
3. Setelah penyimpanan nama form, maka selanjutnya kita akan diminta untuk memasukkan nama proyek, seperti ditunjukkan oleh gambar 1.11. Masukkan nama proyek dan tekan tombol **Save**.



Gambar 1.10 Jendela penyimpanan Form

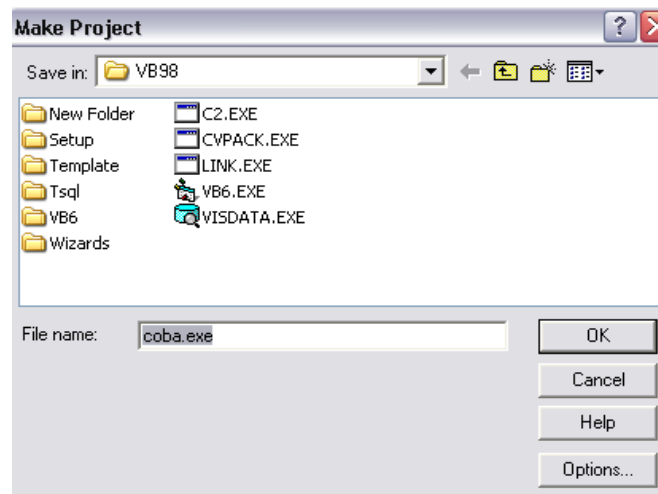


Gambar 1.11 Jendela Penyimpanan Proyek

Untuk menjalankan program aplikasi, dapat dilakukan dengan memilih menu **Run** lalu **Start** atau tekan tombol **F5**.

Selain digunakan untuk merancang program, Visual Basic juga merupakan sebuah compiler. Dengan compiler program dapat dijadikan suatu file yang dapat berdiri sendiri tanpa harus memerlukan bantuan program Visual Basic untuk menjalankannya. Langkah-langkah untuk mengkompilasi program :

1. Jalankan perintah **File | Make [nama project].exe**, lalu akan tampil kotak dialog pilihan seperti ditunjukkan Gambar 1.10



Gambar 1.10 Jendela Pembuatan Kompilasi program.

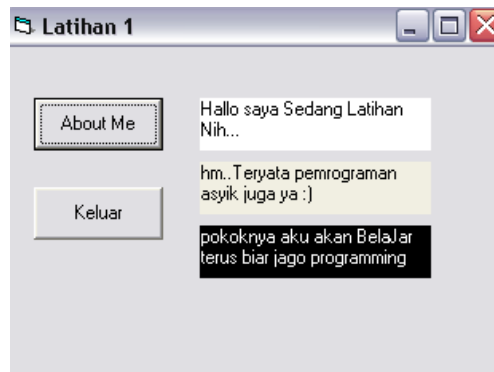
2. Dari kotak dialog itu, tentukan lokasi folder dan tuliskan nama file yang akan digunakan untuk pemanggilan program, lalu tekan tombol **OK**.

Latihan 1 (Teori)

1. Terangkan beberapa hal yang membedakan bahasa pemrograman Visual dengan pemrograman lainnya.
2. Apa perbedaan pokok dari Method, event dan Objek?
3. Apa perbedaan file form dan file project pada visual Basic?
4. Terangkan proses penyimpanan dan kopilasi program pada Visual Basic

Latihan 2 (Praktek)

1. Buatlah sebuah Program sederhana yang menampilkan gambar 1.11 dan gambar 1.12 berikut:



Gambar 1.11



Gambar 1.12

Pertemuan 2 & 3

Tipe Data, Variabel , Konstanta , Fungsi dan Operator

Tujuan Praktikum :

1. Mengenalkan Tipe Data
2. Mengenalkan Variabel dan Konstanta
3. Mengenalkan Operator

TIPE DATA

Sebelum kita mendeklarasikan suatu variabel, alangkah baiknya jika kita mengetahui terlebih dahulu tipe-tipe data yang digunakan dalam Visual Basic. Pemilihan tipe data yang tepat akan dapat menghemat memori, berpengaruh pada kecepatan proses , ketepatan penghitungan, dan lain-lain.

Tipe	Ukuran bytes	Range
Integer	2 byte	-32768 s.d 32767
Long	4 byte	-2147483648 s.d 2147483647
Single	4 byte	Negatif : -3.402823E38 s.d -1.401298E-45 Positif : 1.401298E-45 s.d 3.402823E38
Double	8 bytes	Negatif: -1.79769313486232E308 s.d -4.94065645841247E-324 Positif : 4.94065645841247E-324 s.d 1.79769313486232E308
Currency	8 bytes	-922337203685477.5808 s.d 922337203685477.5807
String	1 byte per karakter	0 s.d 2 milyar karakter (95/97/NT) dan 0 s.d sekitar 65535 karakter (versi 3.1)
Byte	1 byte	0 s.d 255
Boolean	2 byte	True (benar) atau False (salah)
Date	8 byte	1 Januari 100 s.d 31 Desember 9999
Object	4 byte	Referensi Objek
Variant	16 byte + 1 byte per karakter	Null, Error, dan tipe seluruh tipe data lain, misalnya boolean, numerik, string, objek, array.

Beberapa hal yang menjadi pertimbangan dalam penggunaan tipe data, adalah

Penggunaan memori dan rentang data

Terdapat perbedaan dalam pengalokasian memori untuk satu tipe data dengan tipe data yang lain. Tentu saja yang paling baik adalah penggunaan memorinya paling kecil. Misalnya jika Kita memproses data nilai yang mempunyai range (rentang) 0 sampai 100 tanpa desimal maka yang paling tepat adalah menggunakan tipe data Byte (perlu memori 1 byte) bila dibandingkan menggunakan tipe data Integer (2 byte) atau Long (4 byte). Tetapi tentu saja, jika bilangan yang diproses melebihi range tipe data byte atau mempunyai pecahan (desimal) maka gunakan tipe data selain byte.

Ketelitian perhitungan

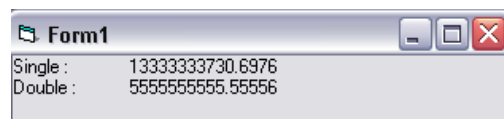
Meskipun ketelitian penghematan memori penting, tetapi kebenaran hasil perhitungan jauh lebih penting. Oleh karena itu, jika ketelitian perhitungan sangat dibutuhkan maka gunakan tipe data dengan ketelitian (presisi) paling tinggi.

Supaya lebih paham ada baiknya kita mencoba latihan berikut :

Tuliskan source code sebagai berikut :

```
Private Sub Form_Activate()  
Dim a As Single  
Dim b As Double  
a = 4 / 3  
b = 5 / 9  
Form1.Print "Single : ", a * 10000000000#  
Form1.Print "Double : ", b * 10000000000#  
End Sub
```

Setelah dijalankan, maka Anda akan mendapatkan hasil seperti diunjukkan oleh Gambar 2.1.



Gambar 2.1

Penggunaan tipe data variant

Tipe data variant dapat dipakai untuk menampung data Null, Error dan seluruh tipe data lain misalnya boolean, numerik, string, objek, array . Jika kita ingin langsung memakai variabel tanpa harus mendeklarasikannya, maka variabel yang kita gunakan itu akan beripe variant. Variabel variant dapat menyimpan semua tipe data dasar Visual Basic dan berubah format secara otomatis. Hanya saja tipe data ini selain tidak memberikan kepastian terutama jenis data

yang dikandungnya juga memerlukan byte memori yang lebih banyak. Sehingga penggunaannya haruslah sangat selektif .

Tipe data buatan sendiri

Sama seperti halnya bahasa pemrograman lainnya, di dalam bahasa Visual Basic, kita juga dapat membuat tipe data sendiri dengan mengambil satu atau beberapa tipe satandar yang disediakan. Biasanya pembuatan data sendiri ini digunakan dalam penanganan database yang sering membutuhkan variabel yang menampung record-record. Record data tersebut terdiri dari beberapa tipe data yang berbeda. Bentuk umum tipe data buatan sendiri adalah:

```
Private Type NamaTipeData
...
...
End Type

Atau

Private Type NamaTipeData
...
...
End Type
```

Contoh tipe databuatan sendiri adalah:

```
Public Type DataMahasiswa
NIM As String * 6
Nama As String * 25
Alamat As String * 25
Kota As String * 10
TempatLahir As String * 10
TanggalLahir As Date
MataKuliah As Long
End Type
```

VARIABEL

Variabel adalah sebuah nama yang diberikan pada sebuah bagian memori, dimana pada bagian tersebut diletakkan data, baik berupa data numerik maupun data string. Variabel juga sering diartikan sebagai suatu tempat dalam memori komputer yang diberi nama (sebagai pengenalan) dan dialokasikan untuk menampung data. Penggunaan variabel dimaksudkan untuk menjaga kompatibilitas dengan bahasa Basic.

Mendeklarasikan Variabel

Untuk menggunakan variabel, kita harus mendeklarasikannya terlebih dahulu. Hanya saja, deklarasi ini sifatnya optional, atau dengan kata lain variabel tersebut tidak menjadi masalah jika tidak dideklarasikan.

Variabel dideklarasikan dengan menggunakan pernyataan **Dim**, dan menggunakan sintaks berikut :

Dim NamaVariabel [As tipe]

- NamaVariabel : pengenalan/nama dari variabel yang digunakan
- As tipe : sifatnya optional, yang bertujuan menentukan tipe dari variabel yang dideklarasikan tersebut. Jika tipe variabel tidak didefinisikan, maka Visual Basic akan menganggap tipe variabel tersebut variant.

Contoh 1:

```
Dim Nama As String
Dim Alamat As String * 30
Dim Gaji As Long
```

Contoh2:

```
Nama$="MONi"
Alamat$="Jl. C Simanjutak"
Gadji$=200000
```

Contoh1 adalah cara pendeklarasian variabel yang umum digunakan. Cara ini lebih dianjurkan untuk dipakai karena lebih terencana dan terkontrol.

Contoh2 adalah cara pendeklarasian eksplisit. Deklarasi eksplisit membuat MS Visual Basic 6.0 mengalokasikan nama variabel beserta tipenya dengan langsung mengisi nilai variabelnya.

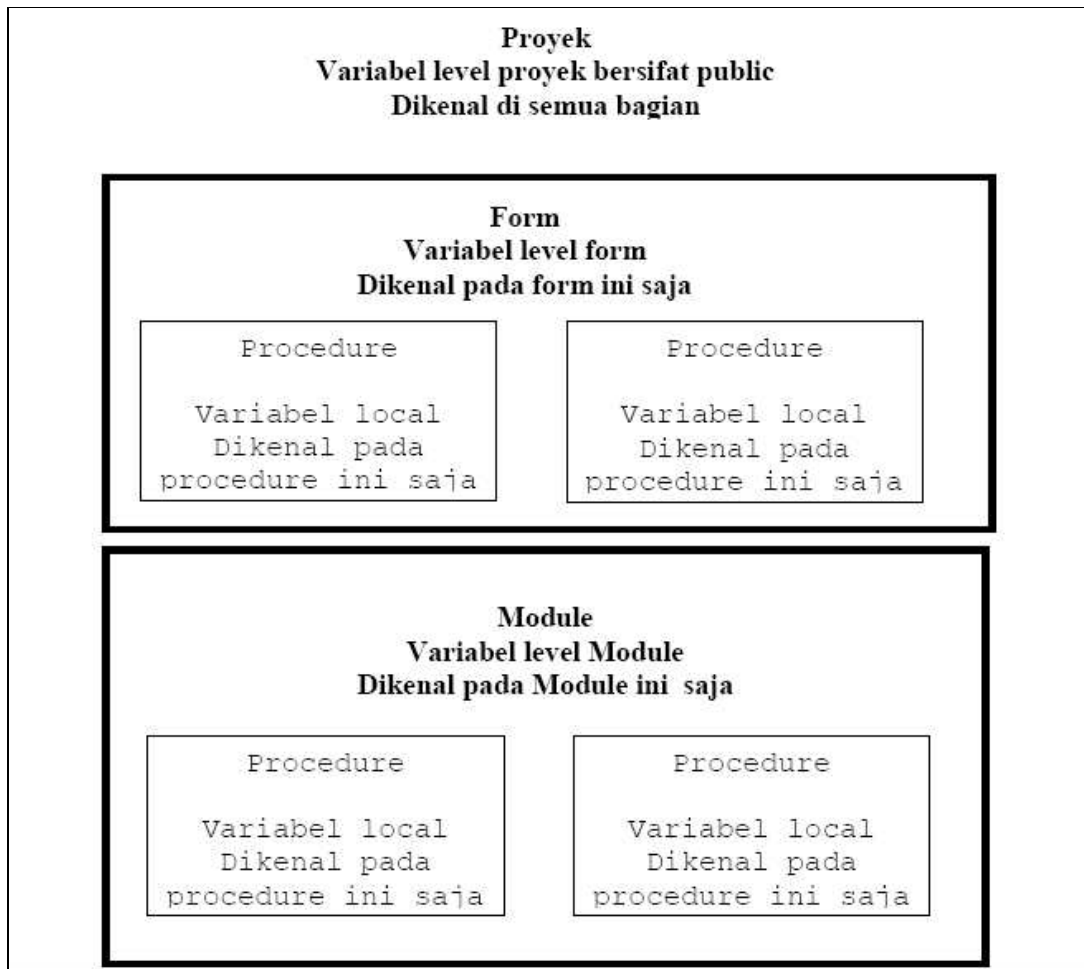
Memberikan Nama Variabel

Untuk memberi nama pada variabel, terdapat aturan-aturan tertentu yang diberlakukan. Aturan-aturan yang telah ditetapkan sebagai berikut:

- Harus dimulai dengan karakter alfabet (huruf), tidak diperbolehkan diulai dengan angka
- Huruf-huruf selanjutnya bisa karakter, angka dan karakter garis bawah(underscore)
- Harus unik (tidak boleh ada yang sama) pada ruang lingkup (scope) variabel yang sama. Jadi tidak diperkenankan menggunakan kata-kata yang dikenal oleh Visual Basic, seperti input, print

dan lainnya. Tapi kita bisa menggunakan kata-kata berikut yang diberi imbuhan, seperti `print_angka`, dan sebagainya.

- Maksimum 255 karakter, tetapi hanya 40 karakter pertama yang dianggap sebagai nama variabelnya (karakter ke-41 dan seterusnya diabaikan)
- Beberapa karakter tidak diperbolehkan dipakai, misalnya karakter yang dipakai untuk operator (`+`, `-`, `*`, `/`, `<`, `>` dan lain-lain), titik dua (`:`), koma (`,`) dan lain-lain.



Gambar 2.2 Deskripsi Ruang Lingkup Variabel

Ruang lingkup paling dalam dalam suatu variabel adalah ruang lingkup procedure. Jika kita mendeklarasikan secara eksplisit suatu variabel di dalam suatu procedure dengan perintah `Dim`, `Private` atau `Static`, maka variabel yang dideklarasikan mempunyai ruang lingkup procedure artinya variabel tersebut hanya dikenal dan dapat dipakai pada procedure yang bersangkutan. Jadi bersifat lokal pada procedure itu. Jika pada procedure lain atau pada level lain atau level di atasnya (form atau modul) mempunyai variabel dengan nama yang sama, maka tidak akan saling mengganggu. Contoh deklarasi level procedure:

```
Private Sub Command1_Click()  
Dim Nama As String  
Dim Alamat As String  
Static No As Integer  
...  
...  
End Sub
```

Perbedaan antara Dim dan Static adalah pada waktu berlakunya variabel. Pendeklarasian variabel level procedure dengan pernyataan Dim maka waktu berlakunya variabel tersebut hanya selama procedure masih berjalan. Jika procedure telah selesai maka variabel otomatis dihapus dari memori. Sedangkan jika menggunakan pernyataan Static maka waktu berlakunya adalah selama program aplikasi masih berjalan. Jadi walaupun procedure telah selesai, nilai terakhir dari variabel tersebut masih tersimpan dalam memori sehingga jika program masuk lagi ke procedure tersebut, nilai yang lama tetap dipertahankan, tetapi tentu saja jika program aplikasi selesai maka variabel tersebut hilang dari memori. Variabel dalam sebuah procedure dapat bersifat static dengan cara memberikan pernyataan Static di depan nama procedure seperti contoh berikut:

```
Static Sub Command1_Click()  
Dim Nama As String  
Dim Alamat As String  
Static No As Integer  
...  
...  
End Sub
```

Jika ada pernyataan Static di depan nama procedure, maka seluruh variabel dalam procedure tersebut bersifat static walaupun dideklarasikan dengan pernyataan Dim.

Ruang lingkup selanjutnya adalah variabel dengan ruang lingkup modul atau form dan ruang lingkup global (aplikasi), atau sering disebut sebagai **Variabel Global**. Keduanya dapat diperoleh dengan mendeklarasikan variabel di luar procedure.

Jika variabel dideklarasikan di luar procedure dengan pernyataan **Private** atau **Dim**, maka variabel tersebut mempunyai level modul atau form akibatnya variabel tersebut hanya dikenali dan dapat dipakai pada modul atau form yang bersangkutan tetapi tidak dapat dikenali pada modul atau form lain pada aplikasi yang sama. Sedangkan jika dideklarasikan dengan pernyataan **Public** maka

variabel tersebut akan mempunyai level global (aplikasi) artinya dapat dikenali pada seluruh modul, form atau procedure yang terdapat pada program aplikasi yang bersangkutan.

Contoh :

```
Public Judul As String * 80 'level global atau 'public atau aplikasi
Private Bilangan As Integer 'level modul/form
Dim Harga As Single 'level modul/form
Private Sub Command1_Click()
Dim Nama As String
Dim Alamat As String
Static No As Integer
...
...
End Sub
```

KONSTANTA

Konstanta adalah nama yang berarti yang menggantikan sebuah bilangan atau rangkaian teks yang tidak berubah. Dengan adanya konstanta, dapat meningkatkan keterbacaan kode program, menghemat memori, dan memudahkan perubahan global. Konstanta ini sebenarnya bekerja seperti variabel, tetapi tidak dapat dimodifikasi nilainya pada saat program berjalan. Sintaks pemakaian konstanta :

[Global] Const NamaKonst=ekspresi[, NamaKonst=ekspresi]....

- NamaKonst : nama konstanta yang digunakan
- Ekspresi : nilai yang memuat konstanta

Konstanta memiliki daya jangkauan yang sama dengan daya jangkau sebuah variabel. Jika konstanta dideklarasikan pada bagian form, maka konstanta tersebut hanya dikenal oleh fungsi tersebut. Untuk konstanta yang sifatnya global, harus di deklarasikan pada bagian deklarasi sebuah modul dengan tambahan pernyataan **Global**. Jika tidak dideklarasikan Global, maka konstanta tersebut hanya dikenal oleh modul tersebut.

OPERATOR

Macam-macam operator yang sering dipakai dalam MS Visual Basic 6.0

1. operator penugasan (Assignment)

Operator penugasan (Assignment) disimbolkan dengan tanda sama dengan (=) dan berfungsi untuk memasukkan suatu data ke dalam suatu variabel. Syntaxnya adalah:

```
<Variabel>=<Ekspresi>
```

Contoh :

```
Awal=Now
```

```
Bilangan = 0
```

```
Harga = 500
```

2. Operator aritmatika

Operator ini digunakan untuk operasi aritmatika. Operator aritmatika mempunyai hierarki paling tinggi dibanding dengan operator lainnya, yaitu operator perbandingan dan operator logika. Operator aritmatika dengan hierarki dari paling tinggi ke paling rendah adalah sebagai berikut:

Operator	Operasi
^	Pemangkatan
-	Tanda negatif
*, /	Perkalian dan pembagian
\	Pembagian integer
Mod	Modulus (sisa pembagian)
+, -	Penambahan dan pengurangan
+, &	Penggabungan string

Contoh:

Buat sebuah proyek baru

Klik ganda form sampai kita masuk ke Code Editor

Pilih Form pada Object Selector, sehingga akan muncul code seperti berikut :

```
Private Sub Form_Load()
```

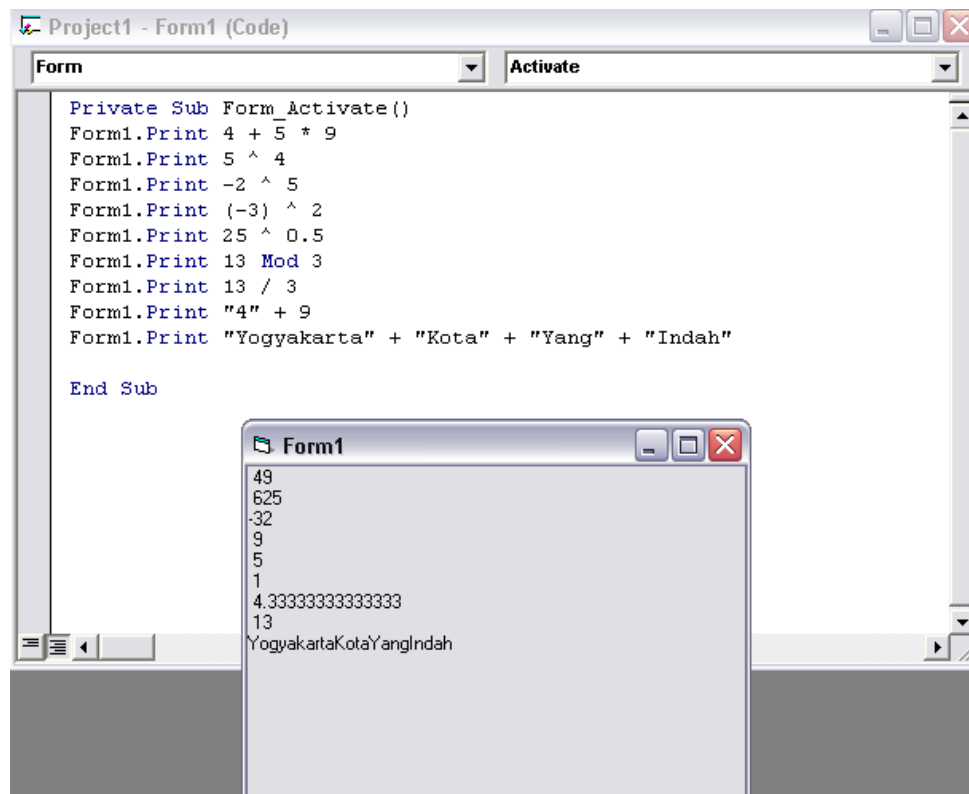
```
End Sub
```

Klik pada kotak jendela procedure lalu pilih Activate sehingga akan muncul

```
procedure Private Sub Form_Activate()
```

```
end sub
```

Ketikkan isi procedure Private Sub Form_Activate() seperti ditunjukkan oleh gambar 2.3. Jalankan Program. Hasil Operasi Aritmatika ini juga ditunjukkan dalam gambar 2.3



Gambar 2.3 Code Operasi Aritmatika dan Hasilnya

3. Operator perbandingan

Operator perbandingan digunakan untuk membandingkan suatu data (ekspresi) dengan data lain dan menghasilkan nilai logika (boolean) benar atau salah.

Tabel operator perbandingan yang disediakan oleh MS Visual Basic 6.0

Operator	Keterangan
=	Sama dengan
<>	Tidak sama dengan
<	Lebih kecil
>	Lebih besar

<=	Lebih kecil atau sama dengan
>=	Lebih besar atau sama dengan
Like	Mempunyai ciri yang sama
Is	Sama referensi objek

4. Operator logika

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

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

Operator logika Not menghasilkan nilai kebalikan dari suatu data logika dan mempunyai hierarki paling tinggi pada operator logika.

Tabel logika Not adalah sebagai berikut :

Ekspresi	Hasil
Not True	False
Not False	True

Tabel logika And adalah sebagai berikut :

Ekspresi	Hasil
False And False	False
False And True	False
True And False	False
True And True	True

Tabel logika Or adalah sebagai berikut :

Ekspresi	Hasil
False Or False	False
False Or True	True
True Or False	True
True Or True	True

Tabel logika Xor adalah sebagai berikut :

Ekspresi	Hasil
False Xor False	False
False Xor True	True
True Xor False	True
True Xor True	False

Tabel logika Eqv adalah sebagai berikut :

Ekspresi	Hasil
False Eqv False	True
False Eqv True	False
True Eqv False	False
True Eqv True	True

Tabel logika Imp adalah sebagai berikut :

Ekspresi	Hasil
False Imp False	True
False Imp True	True
True Imp False	False
True Imp True	True

PENGGUNAAN FUNGSI

Visual Basic menyediakan fungsi-fungsi yang dapat membantu bekerja dalam bilangan-bilangan formula. Seperti fungsi lainnya, fungsi matematik perlu dipakai dalam pernyataan program, dan fungsi ini akan mengembalikan sebuah nilai ke program. Berikut beberapa fungsi yang sering digunakan dalam bahasa Basic beserta kegunaannya:

Fungsi	Kegunaan
Abs(n)	Menghasilkan nilai absolut dari n
Atn(n)	Menghasilkan arkus tangen, dalam radian, dari n
Cos(n)	Menghasilkan kosinus sebuah sudut n. Sudut n dinyatakan dalam radian
Exp(n)	Menghasilkan konstanta e pangkat n
Rnd(n)	Menghasilkan bilangan acak antara 0 dan 1
Sgn(n)	Menghasilkan -1 jika n lebih kecil dari nol, 0 jika n sama dengan nol, dan +1 jika n lebih besar dari nol
Sin(n)	Menghasilkan sinus dari sudut n. Sudut n dinyatakan dalam radian
Sqr(n)	Menghasilkan akar kuadrat dari n
Str(n)	Mengubah nilai numerik menjadi teks
Tan(n)	Menghasilkan tangen dari sudut n. Sudut n dinyatakan dalam radian
Val(n)	Mengubah nilai teks menjadi bilangan

Contoh Program

Program berikut menunjukkan bagaimana operator melakukan operasi aritmatika.

1. Buatlah Project Baru
2. Buat interface program seperti ditunjukkan oleh gambar 2.4



Gambar 2.4 Tampilan Form

Pengaturan properti untuk setiap objeknya adalah sebagai berikut:

Objek	Properti	Nilai
Form1	Caption	Operasi Aritmatika
Label1	Caption	INPUT 1
Label2	Caption	INPUT 2
Label3	BackColor	Menubar
Button1	Caption	REPEAT
Frame1	Caption	Operasi Aritmatika
OptionButton1	Caption	^ (Pangkat)
OptionButton2	Caption	- (Pengurangan)
OptionButton3	Caption	* (Perkalian)
OptionButton4	Caption	/ (Pembagian)
OptionButton5	Caption	\ (Pembagian Integer)
OptionButton6	Caption	Mod
OptionButton7	Caption	+ (Penambahan)
OptionButton8	Caption	& (string)

3. Buka jendela a code dan pada bagian Code Editor ketikkan kode program berikut:

```
Dim a1 As Single, a2 As Single
Dim hasil As Single

Private Sub Command1_Click()
Text1.Text = ""
```

```
Text2.Text = ""
Label3.Caption = ""

End Sub

Private Sub Option1_Click()
a1 = Text1.Text
a2 = Text2.Text
hasil = a1 ^ a2
Label3.Caption = hasil
End Sub

Private Sub Option2_Click()
a1 = Text1.Text
a2 = Text2.Text
hasil = a1 - a2
Label3.Caption = hasil
End Sub

Private Sub Option3_Click()
a1 = Text1.Text
a2 = Text2.Text
hasil = a1 * a2
Label3.Caption = hasil
End Sub

Private Sub Option4_Click()
a1 = Text1.Text
a2 = Text2.Text
hasil = a1 / a2
Label3.Caption = hasil
End Sub

Private Sub Option5_Click()
a1 = Text1.Text
a2 = Text2.Text
hasil = a1 \ a2
Label3.Caption = hasil
End Sub

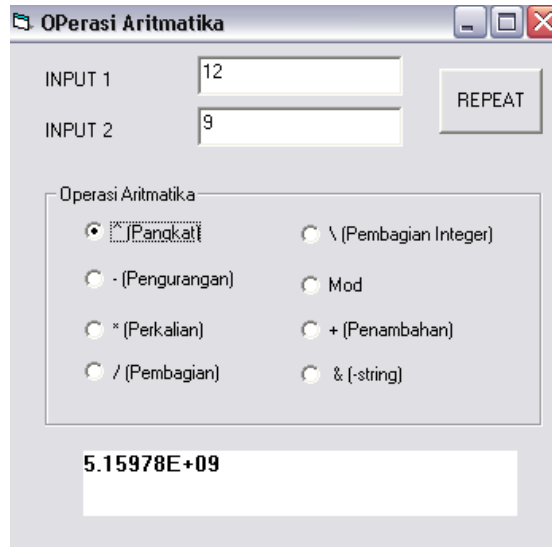
Private Sub Option6_Click()
```

```
a1 = Text1.Text
a2 = Text2.Text
hasil = a1 Mod a2
Label3.Caption = hasil
End Sub

Private Sub Option7_Click()
a1 = Text1.Text
a2 = Text2.Text
hasil = a1 + a2
Label3.Caption = hasil
End Sub

Private Sub Option8_Click()
a1 = Text1.Text
a2 = Text2.Text
hasil = a1 & a2
Label3.Caption = hasil
End Sub
```

4. Simpan Project dengan nama aritmatika
5. Jalankan Program
6. Ketikkan pada input 1 sembarang bilangan (misalnya 200), tekan TAB. Cursor akan ke kotak kedua
7. Ketikkan pada input 2 sembarang bilangan (misalnya 45)
8. Pilih Operator yang akan digunakan (misalnya mod)
9. Maka akan ditampilkan hasilnya seperti ditunjukkan oleh gambar 2.5.



Gambar 2.5 Hasil Tampilan

Latihan

1. Buatlah program untuk Operasi Perbandingan dan Operasi Logika berdasarkan contoh program
2. Buatlah sebuah program untuk mencari Luas Permukaan dan Volume dari Tabung dan Kerucut, dimana User dapat memilih bangun ruang yang akan di cari luas permukaan dan volumenya
3. Buatlah suatu program untuk mencari akar, nilai diskriminan, sumbu simetri, nilai maksimal dan nilai minimal dari suatu persamaan kuadrat $ax^2 + bx + c=0$

Pertemuan 4

STATEMEN INPUT/OUTPUT

Tujuan Praktikum :

1. Mengenalkan fungsi Inputbox dan Message Box
2. Menjelaskan proses penampilan pesan pada MS Visual 6.0

Target Praktikum :

1. Praktikan dapat menggunakan Inputbox dan Message Box
2. praktikan memahami fungsi dari InputBox dan Message Box

Dasar teori

Pada bab ini kita akan belajar mengenai inputbox dan messagebox. Selain itu kita juga akan dijelaskan mengenai cara pemakaian variabel untuk menyimpan data secara sementara dalam program.

Salah satu langkah dalam pembuatan program adalah mendapatkan suatu input dari pengguna program dan kemudian menampilkan hasil dari pengolahan input tadi sehingga menjadi suatu informasi yang lebih berguna. Ada banyak cara untuk mendapatjan input tadi, baik itu dari keyboard atau dari piranti lain seperti mouse, tetapi untuk pertemuan kali ini kita hanya akan membahas beberapa hal saja..

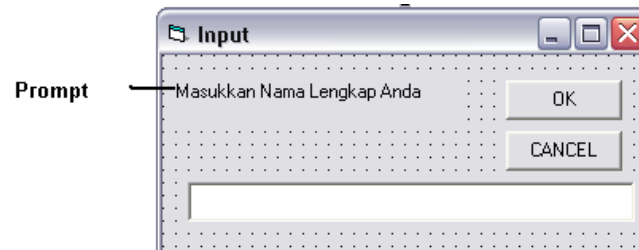
Mendapat masukan dengan InputBox

Kita bisa mendapatkan masukan dari user secara efektif memakai fungsi InputBox dan sebuah variabel. Sebagai contoh perhatikan prosedur di bawah ini :

```
Dim Prompt, FullName
Prompt = "Masukkan nama lengkap Anda."
FullName = InputBoxš(Prompt)
Label1.Caption = FullName
```

Kita telah mendeklarasikan dua variabel dengan pernyataan Dim : Prompt dan FullName. Baris kedua dalam prosedur di atas memasukkan sekelompok karakter atau rangkaian teks, ke variabel Prompt. Pesan ini akan digunakan sebagai argumen teks untuk fungsi InputBox (argumen adalah nilai atau ekspresi yang dilewatkan ke sebuah prosedur atau fungsi). Baris berikutnya memanggil fungsi InputBox dan menunjukkan hasilnya ke variabel FullName.

InputBox adalah fungsi khusus Visual Basic yang menampilkan sebuah kotak dialog di layar dan meminta pemakai untuk memberi masukan, seperti ditunjukkan gambar 3.1.



Gambar 3.1 Kotak dialog InputBox

Menampilkan pesan dengan MsgBox

Fungsi MsgBox memakai rangkaian teks untuk menampilkan keluaran dalam sebuah kotak dialog. Fungsi ini juga mendukung sejumlah argumen tambahan. Sebagai contoh :

```
ButtonClicked= MsgBox(Message,NumberOfButtons, Title)
```

Message adalah teks yang akan ditampilkan di layar, NumberOfButtons adalah jumlah tombol (1 sampai 5), dan Title adalah judul kotak pesan. Contoh hasil dari MsgBox ditunjukkan oleh Gambar 3.2 :

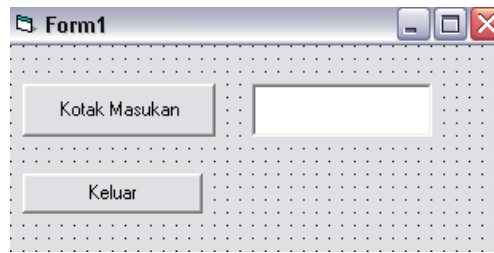


Gambar 3.2 Message Box

Untuk lebih jelasnya kerjakanlah contoh program berikut ini.

Contoh Program :

1. Jalankan Visual Basic, pada menu file pilih New Project.
2. Buatlah interface seperti ditunjukkan Gambar 3.3 :



Gambar 3.3 Tampilan form

3. Klik dua kali tombol perintah Kotak Masukan.
Prosedur Command1_Click muncul dalam jendela Code.
4. Ketikkan pernyataan program berikut untuk mendeklarasikan variabel dan memanggil fungsi InputBox serta MsgBox :

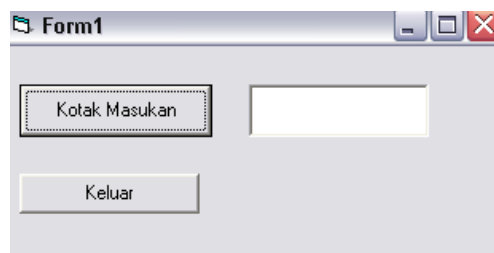
```

Command1 Click
Private Sub Command1_Click()
    Dim prompt, fullname, msg
    prompt = "Masukkan nama lengkap Anda"
    fullname = InputBox$(prompt)
    msg = "hello" + fullname
    Text1.Text = fullname
    MsgBox (msg), , "Input Result"
End Sub

Private Sub Command2_Click()
End
End Sub
    
```

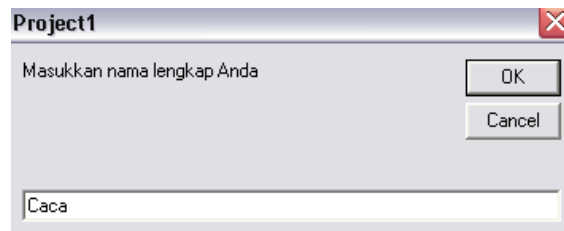
5. Simpan dengan nama Input_Output_Project.
6. Untuk melihat hasilnya, jalankan program yang telah Anda buat tersebut.

Sebagai perbandingan hasil program ditunjukkan oleh gambar 3.4:



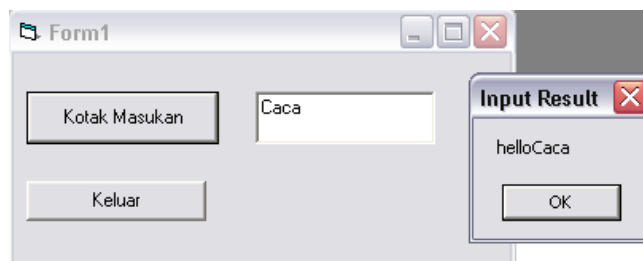
Gambar 3.4 Output Program

Klik tombol *Kotak Masukan* , lalu akan muncul InputBox seperti ditunjukkan gambar 3.5:



Gambar 3.5 Kotak Input Box

Isikan nama Anda kemudian klik tombol *OK*, sehingga muncul MsgBox seperti ditunjukkan gambar 3.6:



Gambar 3.6 Message Box

Klik tombol *OK*, kemudian klik tombol *Keluar*.

Latihan

1. Buatlah program reminder yang dapat mengingatkan kita jadwal kegiatan kita dalam 1 hari, yang muncul setiap 30 menit.

Dengan ketentuan sebagai berikut :

- User memasukkan beberapa jadwal ke dalam program
- Program menampilkan reminder setiap 30 menit dan 5 menit sebelum kegiatan dimulai.
- User dapat mengubah settingan waktunya

Pertemuan 5

STRUKTUR KONTROL KEPUTUSAN

Tujuan Praktikum :

1. Mengenalkan Struktur Kontrol Keputusan
2. Menjelaskan penulisan ekspresi bersyarat
3. Menjelaskan perbedaan dari setiap bentuk struktur kontrol keputusan

Target Praktikum :

1. Praktikan dapat menulis ekspresi bersyarat
2. Praktikan dapat menggunakan struktur kontrol keputusan untuk membuat program
3. Praktikan dapat membedakan pemakaian dari setiap bentuk struktur kontrol keputusan

Struktur Kontrol Keputusan

Struktur Kontrol Keputusan adalah salah satu dari Struktur kontrol yang digunakan dalam bahasa pemrograman. Struktur kontrol adalah instruksi (perintah) dengan bentuk dan ciri khas tertentu yang digunakan untuk mengatur jalannya program. Selain struktur kontrol keputusan, Visual Basic 6 juga mengenal Struktur kontrol Pengulangan yang akan kita bahas pada bab selanjutnya.

Struktur kontrol keputusan digunakan untuk memutuskan pernyataan mana dari suatu program yang akan dikerjakan berdasarkan suatu kondisi tertentu.

Hal yang paling dominan dalam struktur kontrol keputusan ini adalah bahwa sebagai alat bantu pengolahan informasi dalam sebuah prosedur, pernyataan bersyarat dalam struktur ini memberikan pernyataan benar atau salah tentang sebuah properti. Sehingga seringkali kita akan menggunakan Operator perbandingan. Berikut beberapa operator perbandingan yang sering digunakan :

Operator	Arti
Perbandingan	
=	Sama dengan
<>	Tidak sama dengan
>	Lebih besar dari
<	Lebih kecil dari

>=	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan

Selain Operator Perbandingan, dalam menguji suatu pernyataan bersyarat, kita seringkali membutuhkan operator logika dengan penjelasan sebagai berikut :

Operator logika	Arti
And	Jika kedua pernyataan bersyarat True, hasilnya True
Or	Jika salah satu pernyataan bersyarat True, hasilnya True
Not	Hasilnya merupakan kebalikan dari pernyataan bersyarat
Xor	Hasilnya True jika hanya satu pernyataan bersyarat bernilai True

Struktur kontrol keputusan yang sering digunakan dalam pemrograman Visual Basic ada 2 macam, yaitu :

1. Struktur Keputusan **If...Then**
2. Struktur Keputusan **Select...Case**

Struktur Keputusan if...Then

Struktur Keputusan merupakan suatu blok pernyataan khusus yang urutan pelaksanaan pernyataannya dikendalikan oleh suatu pernyataan bersyarat.

Struktur keputusan If..Then dapat ditulis dalam 2 bentuk penulisan , yaitu :

1. **IF <kondisi> THEN <pernyataan>**

dimana *kondisi* adalah pernyataan bersyarat, dan *pernyataan* adalah pernyataan program Visual Basic yang sah. Bila <kondisi> bernilai True maka <pernyataan> akan dikerjakan.

Sebagai contoh,

```
If nilai >= 80 Then Textbox1.Text=A
```

2. **IF <kondisi> THEN**

<pernyataan 1>

ELSE

<pernyataan 2>

END IF

Dimana Bila <kondisi1> bernilai True maka <pernyataan 1> akan dikerjakan, tetapi bila <kondisi1> bernilai False maka <pernyataan 2> yang akan dikerjakan.

Contoh :

```
If nilai >=60 then
```

```
Label1.caption="Anda LULUS"  
Else  
Label1.caption="Maaf Anda Belum Lulus"  
endif
```

Atau

```
IF <kondisi1> THEN  
    <pernyataan jika kondisi1 bernilai True>  
ELSEIF <kondisi2>  
    <pernyataan jika kondisi2 bernilai True>  
ELSEIF <kondisi3>  
    <pernyataan jika kondisi3 bernilai True>  
ELSE  
    <pernyataan jika tidak ada kondisi yang bernilai True>  
END IF
```

Dimana <kondisi> yang merupakan suatu ekspresi logika akan diuji perintah if yang pertama yang ada disebelah kirinya. Jika <kondisi> benar maka blok perintah yang terletak di bawahnya akan dikerjakan sampai menemui perintah Elself atau Else lalu langsung melompat ke pernyataan End If.

Sebagai contoh,

```
If nilai>=85  
    Then Label1.Caption="Anda mendapatkan nilai A"  
Elseif nilai>=70  
    Then Label1.Caption="anda mendapatkan nilai B"  
Elseif nilai>=50  
    Then Label1.Caption="Anda mendapatkan nilai C"  
Else  
    Label1.Caption="Anda mendapatkan nilai D"  
endif
```

Struktur Keputusan Select Case

Select Case sebenarnya serupa dengan struktur If...Then...Elself, hanya saja struktur ini lebih efisien, mudah dibaca dan mudah digunakan terutama jika percabangan dibuat berdasarkan satu variabel kunci, atau uji kasus (test case). Struktur ini biasanya digunakan untuk percabangan dengan pilihan lebih dari 2 (percabangan dengan banyak pilihan).

Tata penulisan untuk struktur Select Case adalah sebagai berikut:

```
SELECT CASE <pilihan>
```

```

CASE <pilihan 1>
    <pernyataan program yang dilaksanakan jika pilihan=pilihan1>
CASE <pilihan 2>
    <pernyataan program yang dilaksanakan jika pilihan=pilihan2>
CASE <pilihan n>
    <pernyataan program yang dilaksanakan jika pilihan=pilihann>
[CASE ELSE
    <pernyataan program yang dilaksanakan jika pilihan1 sampai n tidak ada yang sesuai
    maka pernyataan ini akan dikerjakan>
END SELECT
    
```

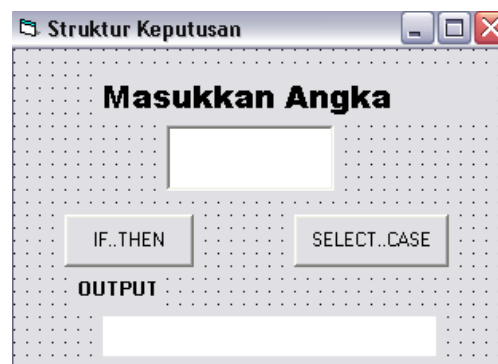
Sebagai contoh :

```

Select Case Nilai
Case >=85
    Label1.Caption=" Anda mendapatkan nilai A"
Case >=70
    Label1.Caption="anda mendapatkan nilai B"
Case >=50
    Label1.Caption="anda mendapatkan nilai C"
Else
    Label1.Caption="Anda mendapatkan nilai D"
End Select
    
```

Contoh program :

1. Jalankan Visual Basic, pada menu file pilih New Project.
2. Buatlah interface seperti ditunjukkan Gambar 4.1



Gambar 4.1 Interface Contoh program Struktur Keputusan

Pengaturan properti untuk setiap objeknya adalah sebagai berikut:

Objek	Properti	Nilai
Form1	Caption	Struktur Keputusan
Label1	Caption	Masukkan Angka
	Font	Arial Black : Size :14
Label2	Caption	OUTPUT
Label3	Caption	dikosongkan
	BackColor	menubar
Text1	Text	dikosongkan
Command1	Caption	IF..THEN
Command2	Caption	SELECT..CASE

3. Masukkan kode program berikut ini :

```

Private Sub Command1_Click()
    bil = Val(Text1.Text)
    If bil < 0 Then
        Label3.Caption = "Bilangan Negatif"
    ElseIf bil <= 10 Then
        Label3.Caption = "Bilangan asli kurang dari 10"
    ElseIf bil <= 50 Then
        Label3.Caption = "Bilangan asli kurang dari 50"
    Else
        Label3.Caption = "Bilangan asli diatas 50"
    End If
End Sub

Private Sub Command2_Click()
    bil = Val(Text1.Text)
    Select Case bil
    Case Is < 0
        Label3.Caption = "Bilangan Negatif"
    Case Is <= 10
        Label3.Caption = "Bilangan asli kurang dari 10"
    Case Is <= 50
        Label3.Caption = "Bilangan asli kurang dari 50"
    Case Else
        Label3.Caption = "Bilangan asli diatas 50"
    End Select
End Sub

```

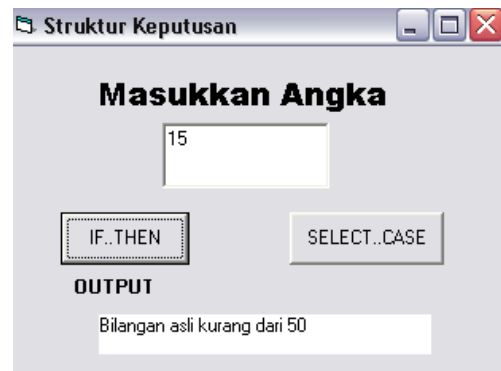
4. Simpan dengan nama Strukturkeputusan
5. Jalankan program tersebut untuk melihat bagaimana hasilnya.
6. Berikut ini adalah hasil program di atas :



Gambar 4.2 Output program

Isikan nilai angka kemudian klik Tombol **IF..THEN** atau **SELECT CASE**

Misalkan kita masukkan Angka 15, maka pada bagian Output akan muncul tulisan "Bilangan asli kurang dari 50", seperti ditunjukkan Gambar 4.3.



Gambar 4.3 Output Program setelah diinputkan nilai

Latihan

1. Buatlah sebuah program untuk menentukan harga barang : Buku (2500) ; pensil (12000); Penggaris (5000) ; pensil case (8500) dan bolpoint (9000). User kemudian menginputkan jumlah barang yang dibeli, dan program tersebut akan menghitung jumlah harga dari barang tersebut. Jika jumlah harga > 40.000 ; pembeli mendapatkan diskon 5 %, jika jumlah harga >100.000; pembeli akan mendapatkan diskon 10 %.
2. Buatlah sebuah program untuk menentukan nilai dan kelulusan mahasiswa dengan 2 persyaratan sebagai berikut :
 - ✓ Jika Mahasiswa Mendapatkan Nilai 80-100 : "A"
 - Jika Mahasiswa Mendapatkan Nilai 60-79 : "B"
 - Jika Mahasiswa Mendapatkan Nilai 50-59 : "C"
 - Jika Mahasiswa Mendapatkan Nilai 35-49 : "D"
 - Jika Mahasiswa Mendapatkan Nilai di bawah 35 : "E"
 - ✓ Jika Mahasiswa mendapatkan nilai A, B dan C akan muncul pesan "Anda Lulus"
 - Jika Mahasiswa mendapatkan nilai D dan E akan muncul pesan "Anda tidak Lulus"User Hanya memasukkan nilai, dan program akan menampilkan pesan apakah mahasiswa tersebut lulus atau tidak.

Pertemuan 6 dan 7

STRUKTUR PENGULANGAN (Looping)

Tujuan Praktikum :

1. Mengenalkan struktur-struktur pengulangan dalam visual Basic
2. Menjelaskan perbedaan penggunaan dari setiap bentuk struktur pengulangan

Target Praktikum :

1. Praktikan dapat menggunakan struktur pengulangan untuk meningkatkan efisiensi program
2. Praktikan dapat membedakan fungsi dari setiap bentuk struktur pengulangan

Struktur Kontrol Pengulangan

Struktur pengulangan digunakan untuk mengulang perintah, sehingga kita tidak perlu menulis ulang perintah sebanyak pengulangan yang diinginkan. Beberapa jenis struktur kontrol pengulangan adalah sebagai berikut :

1. Do...Loop

Struktur ini digunakan untuk mengulang sebuah blok perintah sampai jumlah tertentu. Pada struktur bentuk ini kita akan mengulang suatu blok pernyataan/perintah selama (while) suatu kondisi (syarat) bernilai True atau sampai (until) suatu kondisi menjadi True.

Bentuk penulisan (*syntax*) struktur Do...Loop memiliki bentuk penulisan (*syntax*) sebagai berikut :

- **DO WHILE <kondisi>**
<perintah>
LOOP

Dimana perintah akan diulang **selama** <kondisi> bernilai TRUE. Pengulangan berhenti bila <kondisi> sudah bernilai **FALSE**.

Bentuk lain dari struktur ini adalah

DO

<perintah>

LOOP WHILE <kondisi>

Dalam bentuk ini perintah akan dieksekusi terlebih dahulu, baru kemudian menjalankan pengujian kondisi di akhir eksekusi. Sehingga paling tidak blok perintah sudah pernah dieksekusi minimal satu kali. Hal ini tentu saja berbeda dengan bentuk Do While..Loop yang bisa saja bentuk ini tidak mengeksekusi blok perintah sama sekali.

- **DO UNTIL <kondisi>**

<perintah>

LOOP

Dimana perintah akan diulang sampai <kondisi> bernilai TRUE. Pengulangan berhenti bila <kondisi> sudah bernilai **TRUE**.

Pada bentuk struktur kontrol di atas pengulangan yang dilakukan minimal adalah nol kali. Sedangkan pada struktur kontrol di bawah ini blok perintah minimal dieksekusi satu kali.

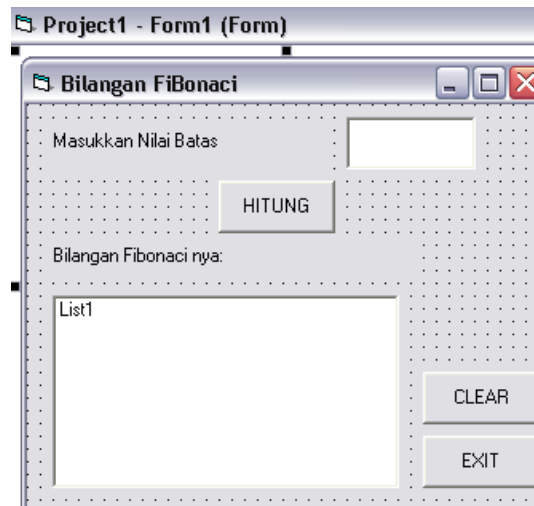
DO

<perintah>

LOOP UNTIL <kondisi>

Contoh Program 1

Program ini merupakan contoh pemrograman pengulangan dengan struktur kontrol Do...Loop. Struktur pengulangan ini dimanfaatkan untuk mencari barisan bilangan Fibonacci. Bilangan Fibonacci adalah barisan yang anggota pertamanya adalah bilangan 1. Contoh barisan bilangan Fibonacci adalah :1, 1, 3, 5, 7, 9, 11...dan seterusnya. Desain form bagi program ini tampak seperti gambar 5.1.



Gambar 5.1 Tampilan Form

Pengaturan properti setiap objeknya adalah sebagai berikut :

Objek	Properti	Setting
Form 1	Caption	Bilangan Fibonacci
CommanButton1	Name Caption	CmdHitung Hitung
CommanButton2	Name Caption	Command2 CLEAR
CommandButton3	Name Caption	Command3 END
Label1	Caption	Masukkan Nilai Batas
Label2	Caption	Bilangan Fibonacci nya:
TextBox1	Text	Kosongkan
ListBox1		

Masukkan kode programnya sebagai berikut :

```

Dim batas As String
Dim x, y, z As Integer
Sub inputbatas()
    batas = Val(Text1.Text)
    If batas = "" Then
        MsgBox "Anda belum memasukkan batas", , "Pesan"
    End If
End Sub

```

```
End Sub

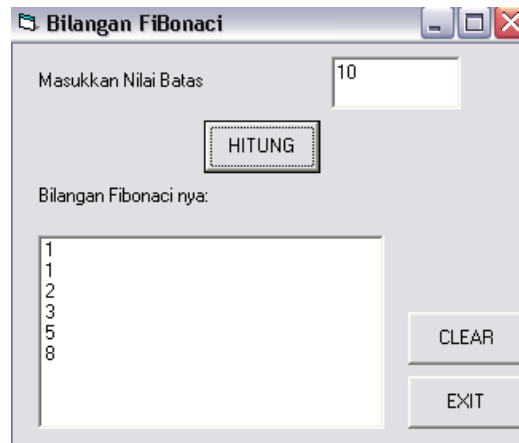
Private Sub CmdHitung_Click()
List1.Clear
inputbatas
If CInt(batas) < 1 Then
List1.AddItem "Tidak ada bilangan Fibonacci kurang dari 1"
Else
x = 1
y = 0
z = 1
Do Until x >= CInt(batas)
List1.AddItem x
x = y + z
y = z
z = x
Loop
End If
End Sub

Private Sub Command1_Click()
End
End Sub

Private Sub Command2_Click()
Text1.Text = ""
List1.Clear

End Sub
```

Saat program dijalankan, ketika tombol Hitung ditekan akan ditampilkan bilangan-bilangan fibonacci dari dengan batas yang diinputkan, seperti ditunjukkan gambar 5.2.



Gambar 5.2 Output program Fibonacci

2. For...Next

Bentuk ini digunakan untuk melakukan pengulangan (iterasi) suatu blok program beberapa kali yang ditentukan nilai awal dan nilai akhir (banyak pengulangannya sudah pasti).

Pada struktur kontrol For...Next, kita tidak perlu menuliskan sebuah kondisi untuk diuji oleh Visual Basic. Kita hanya perlu menuliskan nilai awal dan akhir variabel penghitung. Nilai variabel penghitung ini akan secara otomatis bertambah atau berkurang setiap kali sebuah pengulangan (loop) dilaksanakan.

Standar penulisan struktur ini adalah :

```
FOR <pencacah> = <awal> TO <akhir> [STEP penambahan]
<perintah>
NEXT <pencacah>
```

Dimana ,

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

Pada penambahan Kita dapat menggunakan bilangan negatif atau positif. Jika digunakan bilangan positif, maka nilai akhir harus lebih besar dari nilai awal, demikian sebaliknya jika digunakan bilangan negatif. Sebagai standar Visual Basic selalu menganggap Step penambahan bernilai 1 kecuali jika Kita definisikan sendiri.

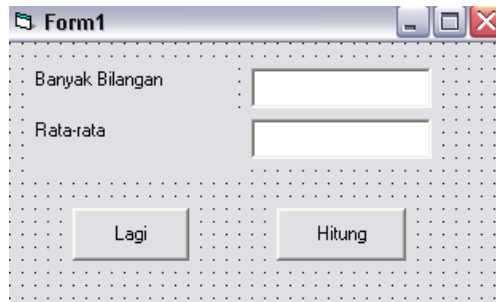
Saat bertemu dengan struktur For...Next Visual Basic akan melakukan langkah-langkah berikut ini :

- ☛ Memasukkan nilai variabel penghitung sama dengan nilai awal.
- ☛ Menguji apakah nilai variabel penghitung lebih dari nilai akhir (pertambahan bernilai positif). Jika benar maka pengulangan akan berhenti dan jika salah Visual Basic akan mengeksekusi perintah di bawah For.
- ☛ Menambah variabel penghitung sebanyak nilai pertambahan.
- ☛ Mengulangi langkah 2 dan 3 di atas.

Contoh Program 2

Program ini merupakan contoh penggunaan struktur kontrolpengulangan For...Next. Struktur kontrol tersebut dimanfaatkan untuk menghitung nilai rata-rata dari sederet bilangan yang Anda masukkan.

Tampilan desain formnya tampak seperti gambar 5.3 :



Gambar 5.3 Tampilan Form

Setting masing-masing objek adalah :

Objek	Properti	Setting
Form 1	Name Caption	Form1 Pangkat
Label1	Name Text	LblBanyakBil Banyak Bilangan
Label2	Name Caption	LblRata Rata-rata
TextBox1	Name Text	TxtBanyakBil (kosongkan)
TextBox2	Name Text	TxtRata (kosongkan)
CommandButton1	Name Caption	CmdHitung &Hitung

CommandButton2	Name Caption	CmdLagi &Lagi
----------------	-----------------	------------------

Code programnya adalah sebagai berikut :

```

Dim Bilangan
Dim Jumlah As Currency
Dim BanyakBilangan, Indeks As Integer
Sub InputBilangan ( )
    Bilangan = InputBox ("Bilangan ke-" & Indeks, "Data")
    If Bilangan = " " Then
        MsgBox "Anda belum memasukkan bilangan ke-" & Indeks,
            "Pesan"
        InputBilangan
    End If
End Sub
Private Sub CmdHitung Click ( )
    BanyakBilangan = TxtBanyakBil.Text
    Jumlah = 0
    For Indeks = 1 To BanyakBilangan
        InputBilangan
        Jumlah = Jumlah + Bilangan
    Next
    TxtRata.Text = Jumlah / BanyakBilangan
    CmdLagi.SetFokus
End Sub
Private Sub CmdLagi _Click ( )
    TxtBanyakBil.Text = " "
    TxtRata.Txt = " "
    TxtBanyakBil.SetFocus
End Sub

```

Saat program dijalankan dan tombol Hitung ditekan, maka akan muncul kotak input tempat Kita memasukkan data pertama. Setelah Kita mengklik OK maka akan ditampilkan kotak input yang kedua, dan demikian seterusnya sebanyak nilai yang telah Kita masukkan pada kotak Banyak bilangan.

Contoh Program 3

Masukkan Source Code sebagai berikut:

```
Private Sub Form_Activate()  
  Dim i, j As Integer  
  For i = 1 To 10  
    For j = 1 To i  
      Form1.Print j;  
    Next j  
    Print  
  Next i  
  Form1.Print  
  Form1.Print  
  For i = 1 To 10  
    For j = 10 To i Step -1  
      Form1.Print 11 - j;  
    Next j  
    Print  
  Next i  
End Sub
```

Jalankan program tersebut. Bagaimana hasilnya?

Latihan

1. Buatlah sebuah program untuk mencari FPB dan KPK dari dua bilangan yang diinputkan
2. Buatlah program untuk menentukan apakah suatu bilangan yang diinputkan termasuk bilangan prima atau tidak

Pertemuan 8 dan 9

SUBROUTINE

Tujuan Praktikum :

1. Menjelaskan konsep subroutine
2. Menjelaskan perbedaan prosedur dan fungsi

Target Praktikum :

1. Praktikan dapat membuat prosedur dan fungsi dalam suatu program
2. Praktikan dapat membedakan penggunaan prosedur dan fungsi

SUBROUTINE

Dalam konsep pemrograman, terutama pemrograman terstruktur, dikenal adanya konsep subroutine. Semula subroutine dikembangkan untuk menyingkat penulisan kode program jika beberapa langkah rumit harus dilakukan beberapa kali di tempat yang berlainan. Selain itu, adanya fasilitas subroutine, program menjadi lebih efisien, jelas, dan waktu penulisan juga menjadi lebih singkat. Istilah “subroutine” sendiri menunjukkan adanya sekelompok statemen bebas yang merupakan bagian dari suatu program atau subroutine lain. Dalam Visual Basic dikenal dua jenis subroutine, yaitu prosedur dan fungsi.

Pada dasarnya prosedur dan fungsi mempunyai tujuan yang sama, hanya perbedaannya terdapat pada apakah prosedur tersebut mengembalikan sebuah nilai atau tidak. Jika mengembalikan sebuah nilai, maka disebut dengan fungsi, sedangkan jika tidak disebut dengan prosedur.

Setelah sebuah prosedur atau fungsi dibuat, maka prosedur atau fungsi tersebut dapat dipanggil sesuai dengan tugas yang telah ditetapkan pada prosedur atau fungsi tersebut. Prosedur atau fungsi ini dapat ditulis pada *form* atau pada modul khusus untuk meletakkan subroutine. Perbedaan peletakan subroutine pada *form* atau modul adalah jangkauan pemakaian subroutine tersebut. Subroutine yang terdapat pada *form* hanya dapat digunakan pada *form* dimana subroutine tersebut dibuat, sedangkan subroutine yang terdapat pada modul dapat digunakan pada seluruh bagian program. Jelas terlihat adanya manajemen pembuatan program dalam Visual Basic.

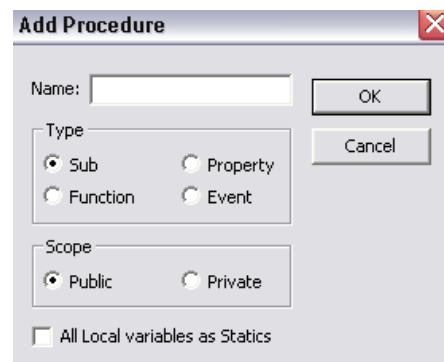
Tugas untuk menentukan apakah sebuah subroutine terletak pada suatu *form* atau modul merupakan tugas pemrogram. Pemrogram dituntut untuk membuat suatu program mudah dikelola.

Yang pasti adalah jika subroutine tersebut digunakan oleh seluruh bagian program, maka buatlah subroutine tersebut pada modul. Tetapi jika subroutine tersebut hanya digunakan oleh sebuah *form*, maka buatlah subroutine tersebut pada *form* walaupun sebenarnya dapat diletakkan pada sebuah modul.

Membuat subroutine pada form

Untuk membuat subroutine pada *form*, langkah pertama yang harus dilakukan adalah mengaktifkan *form* tempat subroutine diletakkan. Untuk lebih jelasnya perhatikan langkah-langkah berikut :

1. Setelah *form* aktif, pilih menu *View > Code*, maka jendela kode akan muncul.
2. Pilih menu *Tools > Add Procedure...* Muncul kotak dialog *Add Procedure* seperti ditunjukkan gambar 6.1



Gambar 6.1 Kotak Dialog Add Procedure

3. *Property* dan *Event*. Jika yang akan Anda buat adalah prosedur, pilih tipe *Sub*, sedangkan jika yang Anda buat adalah fungsi, pilih tipe *Function*.
4. Kotak grup *Scope*, menentukan jangkauan subroutine yang kita buat, Jika yang dipilih *Public*, maka dapat diakses dari mana saja, sedangkan jika yang dipilih *Private*, maka hanya dapat diakses oleh objek (form atau modul)itu sendiri.
5. Tekan *OK* jika telah selesai. Secara otomatis Visual Basic akan membuat *template* subroutine, yaitu judul dan akhir dari sebuah subroutine. Jadi Anda tinggal mengisi parameter (jika diperlukan) dan kodenya.

```
Public Sub IniProsedur ( )
```

```
End Sub
```

Membuat subroutine pada modul

Langkah-langkah untuk membuat subroutine pada sebuah modul adalh sebagai berikut :

1. Jika belum mempunyai modul, tambahkan modul pada program (aplikasi) yang sedang Anda buat dengan memilih menu *Project > Add Module*. Jika akan menambah subroutine pada modul yang sudah ada, aktifkan modul tersebut.
2. Lakukan langkah-langkah seperti pada pembuatan subroutine pada *form*.

Lebih jauh dengan prosedur (Sub)

Secara umum, aturan penulisan prosedur adalah sebagai berikut :

```
[Public | Private] [Static] Sub NamaProsedur [( [Optional] [ByVal | ByRef]
[ParamArray] Param1[( )] [As Tipe] [=NilaiAwal] [, [Optional] [ByVal | ByRef]
[ParamArray] Param2 [( )] [As Tipe] [=NilaiAwal],...)]
    [Pernyataan]
[Exit Sub]
    [Pernyataan]
End Sub
```

- *NamaProsedur* menyatakan nama prosedur yang akan digunakan sebagai pengenal untuk memanggil prosedur tersebut.
- **As Tipe** menyatakan tipe data parameter.
- **= NilaiAwal** menyatakan nilai awal (*default*) parameter. Jadi, jika parameter tersebut tidak disertakan dalam pemanggilan prosedur, maka nilai yang digunakan adalah nilai awal tersebut.
- **[Optional] [ByVal | ByRef] [ParamArray] Param1[()] [As Tipe] [=NilaiAwal] [, [Optional] [ByVal | ByRef] [ParamArray] Param2[()] (As Tipe) [=NilaiAwal], ...]** menyatakan parameter yang akan digunakan oleh prosedur.
- **Optional** menyatakan bahwa parameter tersebut dapat disertakan atau tidak pada saat pemanggilan prosedur.
- **ByVal** menyatakan parameter dilewatkan berdasarkan nilai (*passed by value*)
- **ByRef** menyatakan parameter dilewatkan berdasarkan referensi (*passed by reference*)
- **ParamArray** digunakan sebagai parameter terakhir yang menyatakan bahwa parameter terakhir adalah **Optional** dan merupakan *array* dengan elemen-elemennya bertipe **Variant**. ParamArray memungkinkan prosedur dengan parameter yang jumlahnya tidak menentu. *Keyword* ini tidak dapat digabungkan penggunaannya dengan *keyword* **Optional**, **ByVal**, dan **byRef**.
- **()** menyatakan bahwa suatu parameter adalah *array*.
- **[Pernyataan]** adalah kumpulan pernyataan yang harus dikerjakan oleh prosedur.

- Pernyataan **Exit Sub** digunakan untuk keluar dari prosedur secara paksa (dapat digunakan untuk kondisi tertentu).
- **Static** menyatakan bahwa semua variabel dalam prosedur adalah statis (tidak akan dihapus dari memori walaupun sudah keluar dari prosedur).
- Kata di dalam kurung (...) adalah *optional*, boleh ditulis atau tidak.

Memanggil prosedur

Ada dua cara untuk memanggil prosedur, yaitu dengan menuliskan nama prosedur atau dengan pernyataan **Call** diikuti dengan nama prosedur tersebut. Berikut adalah aturan pemanggilan tersebut :

```
Call NamaProsedur (Param1, Param2,...)
```

atau

```
NamaProsedur Param1, Param2,...
```

Yang perlu diingat adalah jika menggunakan pernyataan **Call** untuk memanggil prosedur maka parameter yang diperlukan harus ditulis dalam tanda kurung "()", sedangkan jika tidak memakai pernyataan **Call** maka tidak perlu ditulis dalam tanda kurung.

Sebagai contoh :

```
Call IniProsedur ("Saya", 56, 6.78)
```

atau

```
IniProsedur "Saya", 56, 6.78
```

Walaupun tidak memberikan perbedaan, pemanggilan prosedur menggunakan **Call** lebih dianjurkan, karena memberikan kejelasan pemanggilan prosedur yang didefinisikan sendiri, berbeda dengan prosedur yang telah didefinisikan oleh Visual Basic.

Lebih jauh dengan fungsi

Fungsi mempunyai aturan penulisan yang mirip dengan prosedur, yaitu :

```
[Public | Private] [Static] Function NamaFungsi [(Optional) [ByVal | ByRef]
```

```
[ParamArray] Param1[( )] [As Tipe] [=NilaiAwal] [, (Optional) [ByVal | ByRef]
```

```
[ParamArray]Param2[( )] [As Tipe] [=NilaiAwal,...]] [As Tipe]
```

```
[Pernyataan]
```

```
[Exit Function]
```

```
[Pernyataan]
```

```
NamaFungsi = Pernyataan
```

```
End Function
```

- *NamaFungsi* menetapkan nama fungsi yang akan digunakan sebagai pengenalan untuk memanggil fungsi tersebut.
- Pernyataan **Exit Function** digunakan untuk keluar dari fungsi secara paksa (dapat digunakan untuk kondisi tertentu).
- *NamaFungsi = Pernyataan* menetapkan nilai yang akan dikembalikan oleh fungsi. Setiap fungsi harus mempunyai pernyataan ini.
- **As Tipe** pada akhir deklarasi fungsi menetapkan tipe data yang dikembalikan oleh fungsi.
- Untuk keterangan lain lihat aturan prosedur.

Memanggil fungsi

Karena sebuah fungsi selalu mengembalikan suatu nilai, maka nilai yang dikembalikan dapat diletakkan pada sebuah variabel, atau ditampilkan langsung, atau dijadikan bagian dari suatu ekspresi.

Contoh :

```
X = Akar (100)           `ditampung dalam sebuah variabel
```

atau

```
Print Akar (100)       `ditampilkan langsung
```

atau

```
X = 23 * Akar (100)    `menjadi bagian dari suatu ekspresi
```


Pertemuan 10

PEMROGRAMAN GRAFIS

Tujuan Praktikum :

1. Mengenalkan jenis-jenis kontrol grafis dalam Visual Basic
2. Menjelaskan metode-metode yang digunakan dalam pemrograman grafis.

Target Praktikum :

1. Praktikan mampu menggunakan kontrol grafis
2. Praktikan memahami konsep pemrograman grafis

Pemrograman Grafis

Gambar dan pemrograman grafis merupakan hal yang penting untuk dipelajari dalam Visual Basic, karena bisa menambah kemampuan Anda dalam mendesain visual effect bagi program-program yang Anda buat.

- Line, digunakan untuk membuat garis lurus diatas form.
- Shape, digunakan untuk membuat lingkaran, persegi empat, elips, atau persegi empat bersiku bulat.
- Image, untuk menampilkan gambar pada form
- Picture Box, digunakan untuk menampilkan gambar di form. Namun kontrol ini paling baik jika digunakan untuk dipindahkan atau dibuat animasi.

Konsep Pemrograman Grafis

1. Sistem Koordinat

Hampir semua operasi pemrograman grafis bekerja dengan sistem koordinat, seperti berpindah, mengubah ukuran, dan lain-lain. Pada Visual Basic hanya dikenal sistem koordinat dua dimensi, walaupun nantinya bisa saja dibuat sistem koordinat tiga dimensi dengan tambahan rutin yang Anda buat.

Untuk mendefinisikan koordinat pada form atau objek yang lain digunakan penulisan (x,y). Standar awal dari koordinat adalah (0,0) pada titik paling kiri atas dari objek yang akan Anda definisikan koordinatnya.

2. Warna

Untuk mendefinisikan warna pada Visual Basic bisa digunakan :

- Fungsi RGB
- Fungsi QBColor untuk memilih satu dari 16 warna pada Microsoft Quick Basic.
- Salah satu konstanta intrinsic yang bisa dilihat pada Object Browser.
- Langsung mengisikan nilai warna, misalnya `εH00FFFFFFε` untuk warna putih.

Untuk menggunakan fungsi RGB digunakan standar penulisan rutin RGB (merah, hijau, biru). Fungsi ini akan menghasilkan warna sebagai kombinasi dari nilai-nilai yang Anda isikan pada merah, hijau, dan biru. Nilai yang bisa diisikan adalah dari 0 hingga maksimum 255. Sebagai contoh :

```
Text1.BackColor = RGB (0,0,0)
Text1.BackColor = RGB (255,255,255)
Pset (100,100), RGB (255,255,0)
```

Untuk menggunakan fungsi QBColor digunakan standar penulisan QBColor (warna). Fungsi ini akan mengembalikan warna yang berasosiasi dengan nilai yang dimasukkan pada parameter warna, yang antara 0 sampai 15. Sebagai contoh :

```
Text1.BackColor = QBColor (2)
```

Kontrol untuk Pemrograman Grafis

1. Kontrol Line

Kontrol ini digunakan untuk menggambar garis lurus yang menghubungkan antara dua titik pada form, saat berada dalam metode desain. Anda bisa mengatur posisi, warna, dan gaya dari garis yang akan Anda perlihatkan di atas form.

Untuk mengubah tebal garis digunakan properti `BorderWidth` dari 1 hingga 20, dan untuk mengubah warna garis digunakan properti `BorderColor`.

2. Kontrol Shape

Kontrol ini menyediakan beberapa bentuk geometri yang bisa Anda buat. Visual Basic juga memberikan berbagai cara pengisian bentuk-bentuk geometri tersebut.

3. Kontrol Image

Fungsi utama dari kontrol ini adalah menampilkan gambar. Namun demikian kontrol ini memiliki sebuah properti unik bernama `Stretch`. Jika properti ini bernilai `True` maka fungsinya adalah menarik gambar bitmap (.BMP), metafile (.WMF), dan icon (.ICO) menjadi sebesar kontrol Image. Hal ini

berbeda dengan picture box atau gambar latar belakang form, yang mempunyai fungsi "Stretch" hanya untuk jenis gambar metafile.

4. Kontrol Picture Box

Kontrol picture box fungsinya mirip dengan kontrol Image. Kelebihannya adalah kemampuan untuk menerima objek lain di atasnya, bahkan Anda bisa menggambar atau menulis di atas kontrol picture box tersebut. Jika kemudian Anda memindahkan posisi kontrol picture box terhadap form, posisi objek yang berada di dalam picture box tetap tak akan berubah. Prinsip ini mirip dengan penggunaan kontrol Frame.

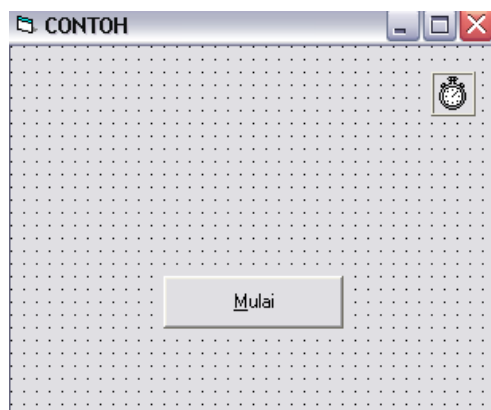
Metoda-Metoda Pemrograman Grafis

Visual Basic menyediakan beberapa metoda yang bisa digunakan dalam pemrograman grafis yang meliputi :

- Cls, pemanggilan metoda ini membuat semua grafis akan dihapus.
- Pset, digunakan untuk mengatur warna pada sebuah titik terpilih. Standar bagi penulisan rutusnya adalah : (Objek.) Pset (x,y) [,warna]
- Point, digunakan untuk mengembalikan nilai warna pada suatu titik.
- Line, digunakan untuk membuat garis, persegi, atau mengisi kotak.
(Objek,) Line [(x1,y1) –(x2,y2) [, warna]
- Circle, digunakan untuk menggambar lingkaran, elips.
- PaintPicture, mengecat grafis pada suatu lokasi terpilih.

Contoh Program 1

Berikut ini merupakan contoh program menggunakan Pset. Tampilan formnya tampak seperti gambar 8.1 :



Gambar 8.1 Tampilan Form

Settingan masing-masing objek :

Objek	Properti	Setting
Form1	Name	FrmContoh1
	Caption	Contoh Pset
	DrawWidth	5
CommandButton1	Name	CmdBerhenti
	Caption	&Mulai
Timer1	Name	TmrPset
	Interval	10
	Enabled	False

Rutin programnya :

```

Private Sub CmdBerhenti_Click ( )
    If CmdBerhenti.Caption = "&Mulai" Then
        TmrPset, Enabled = True
        CmdBerhenti.Caption = "&Berhenti"
    Else
        Cls
        TmrPset.Enabled = Falsa
        CmdBerhenti.Caption = "&Mulai"
    End If
End Sub

Private Sub TmrPset _ Timer ( )
    Merah = Rnd * 255
    hijau = Rnd * 255
    biru = Rnd * 255
    x = Rnd * FrmContoh1.Width
    y = Rnd * FrmContoh1.Height
    Pset (x,y, RGB (merah, hijau, biru)
End Sub

```

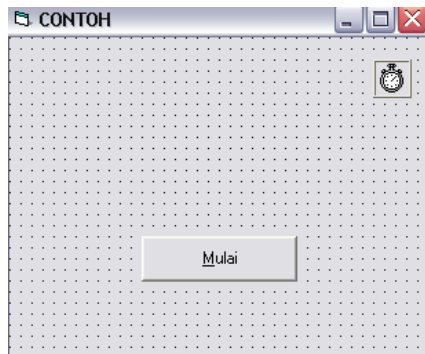
Catatan :

Rnd adalah salah satu fungsi dalam Visual Basic yang akan mengembalikan nilai secara acak (random) mulai dari 0 hingga 1.

Jika program Contoh 1 dijalankan maka form akan berisikan titik yang diberi warna secara acak.

Contoh Program 2

Berikut ini merupakan contoh program menggunakan Line. Tampilan formnya tampak seperti gambar 8.2 :



Gambar 8.2 Tampilan Form

Settingan masing-masing objek adalah :

Objek	Properti	Setting
Form1	Name Caption DrawWidth	FrmContoh2 Contoh Line 2
CommandButton1	Name Caption	CmdBerhenti &Mulai
Timer1	Name Interval Enabled	TmrLine 10 False

Source Code nya :

```

Private Sub CmdBerhenti_Click ( )
    If CmdBerhenti.Enabled = "&Mulai" Then
        TmrLine.Enabled = True
        CmdBerhenti.Caption = "&Berhenti"
    Else
        Cls
        TmrLine.Enabled = False
        CmdBerhenti.Caption = "&Mulai"
    End If
End Sub

```

```

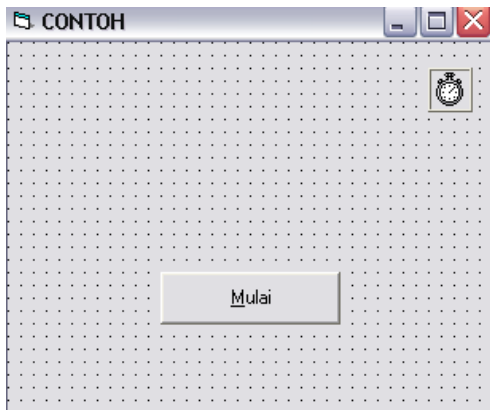
Private Sub TmrLine_Timer ( )
    Static X1, Y1, X2, Y2 As Currency
    merah = Rnd * 255
    hijau = Rnd * 255
    biru = Rnd * 255
    X2 = Rnd * FrmContoh1.ScaleWidth
    Y2 = Rnd * FrmContoh1.ScaleHeight
    Line (X1, Y1) - (X2, Y2, RGB (merah, hijau, biru)
    X1 = X2
    Y1 = Y2
End Sub

```

Jika program Contoh2 dijalankan,form akan berisikan garis-garis yang selalu menyambung dan diberi warna secara acak.

Contoh Program 3

Berikut ini merupakan contoh program menggunakan Circle. Tampilan formnya tampak seperti gambar 8.3 :



Gambar 8.3 Tampilan Form

Settingan masing-masing objek adalah :

Objek	Properti	Setting
Form1	Name	FrmContoh3
	Caption	Contoh Circle

	DrawWidth	3
CommandButton1	Name	CmdBerhenti
	Caption	&Mulai
Timer1	Name	TmrPset
	Interval	10
	Enabled	False

Source Code nya :

```
Private Sub CmdBerhenti_Click ( )
    If CmdBerhenti.Caption = "&Mulai" Then
        TmrCircle.Enabled = True
        CmdBerhenti.Caption = "&Berhenti"
    Else
        Cls
        TmrCircle.Enabled = False
        CmdBerhenti.Caption = "&Mulai"
    End If
End Sub

Private Sub Circle_Timer ( )
    merah = Rnd * 255
    hijau = Rnd * 255
    biru = Rnd * 255
    X = ScaleWidth / 2
    Y = ScaleHeight / 2
    Radius = Y * Rnd
    Circle (X, Y), Radius, RGB, (merah, hijau, biru)
End Sub
```

Jika program Contoh3 tersebut Anda jalankan, maka form akan berisikan lingkaran yang berpusat di tengah-tengah form dengan radius dan warna yang acak.