

Modul 1, Pengenalan Visual Basic

Visual Basic adalah salah satu development tools untuk membangun aplikasi dalam lingkungan Windows. Dalam pengembangan aplikasi, Visual Basic menggunakan pendekatan Visual untuk merancang user interface dalam bentuk form, sedangkan untuk kodingnya menggunakan dialek bahasa Basic yang cenderung mudah dipelajari. Visual Basic telah menjadi tools yang terkenal bagi para pemula maupun para developer dalam pengembangan aplikasi skala kecil sampai ke skala besar.

Dalam lingkungan Window's *User-interface* sangat memegang peranan penting, karena dalam pemakaian aplikasi yang kita buat, pemakai senantiasa berinteraksi dengan User-interface tanpa menyadari bahwa dibelakangnya berjalan instruksi-instruksi program yang mendukung tampilan dan proses yang dilakukan.

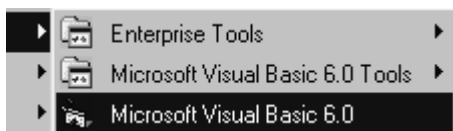
Pada pemrograman Visual, pengembangan aplikasi dimulai dengan pembentukan *user interface*, kemudian *mengatur properti dari objek-objek* yang digunakan dalam user interface, dan baru dilakukan *penulisan kode program* untuk menangani kejadian-kejadian (event). Tahap pengembangan aplikasi demikian dikenal dengan istilah pengembangan aplikasi dengan *pendekatan Bottom Up*.

IDE Visual Basic

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

Menjalankan IDE

Salah satu cara untuk mengaktifkan IDE Visual Basic adalah menjalankannya dari Menu **Start**, pilih menu **Command**, dan pilih **Microsoft Visual Basic 6.0** dan akhirnya **Microsoft Visual Basic 6.0**.



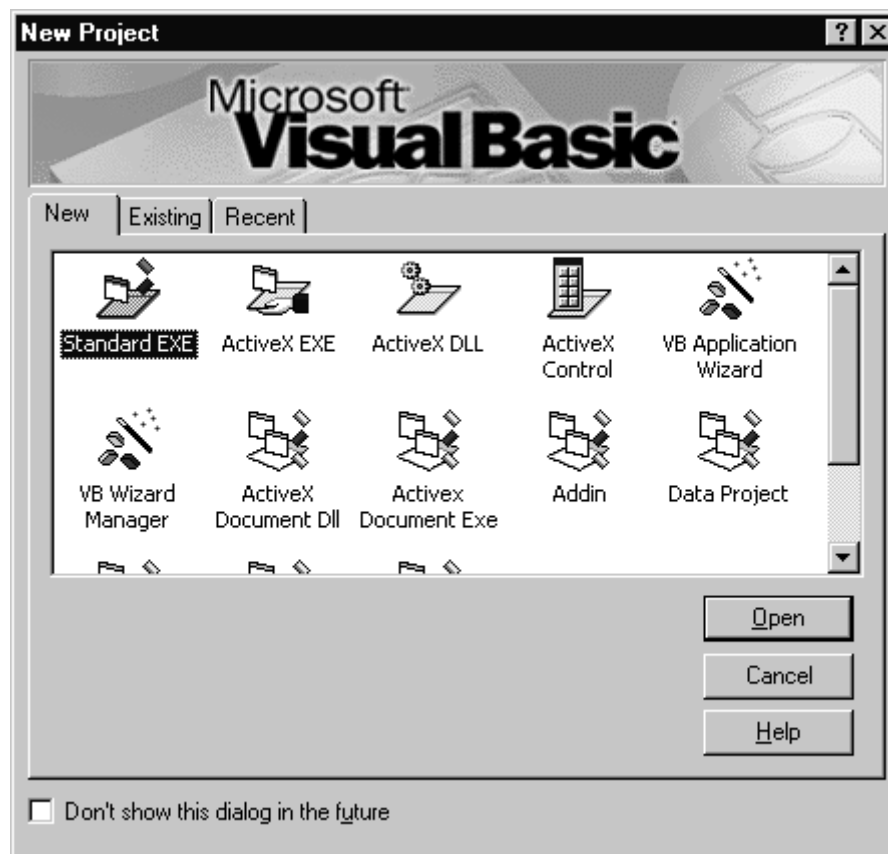
Gambar 1-1. Mengaktifkan IDE Visual Basic 6.0

Catatan : Seperti aplikasi Windows umumnya, anda dapat juga mengaktifkan IDE Visual Basic dengan melakukan open terhadap file yang berkaitan dengan Visual Basic, misalnya file *.vbp yang merupakan file project dari Visual Basic.

Memilih jenis Project

Sesaat anda aktif di IDE Visual Basic, maka anda akan dihadapkan kepada suatu pilihan terhadap jenis Project yang ingin anda buat sebagaimana yang ditunjukkan oleh **Gambar 1-2**.

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



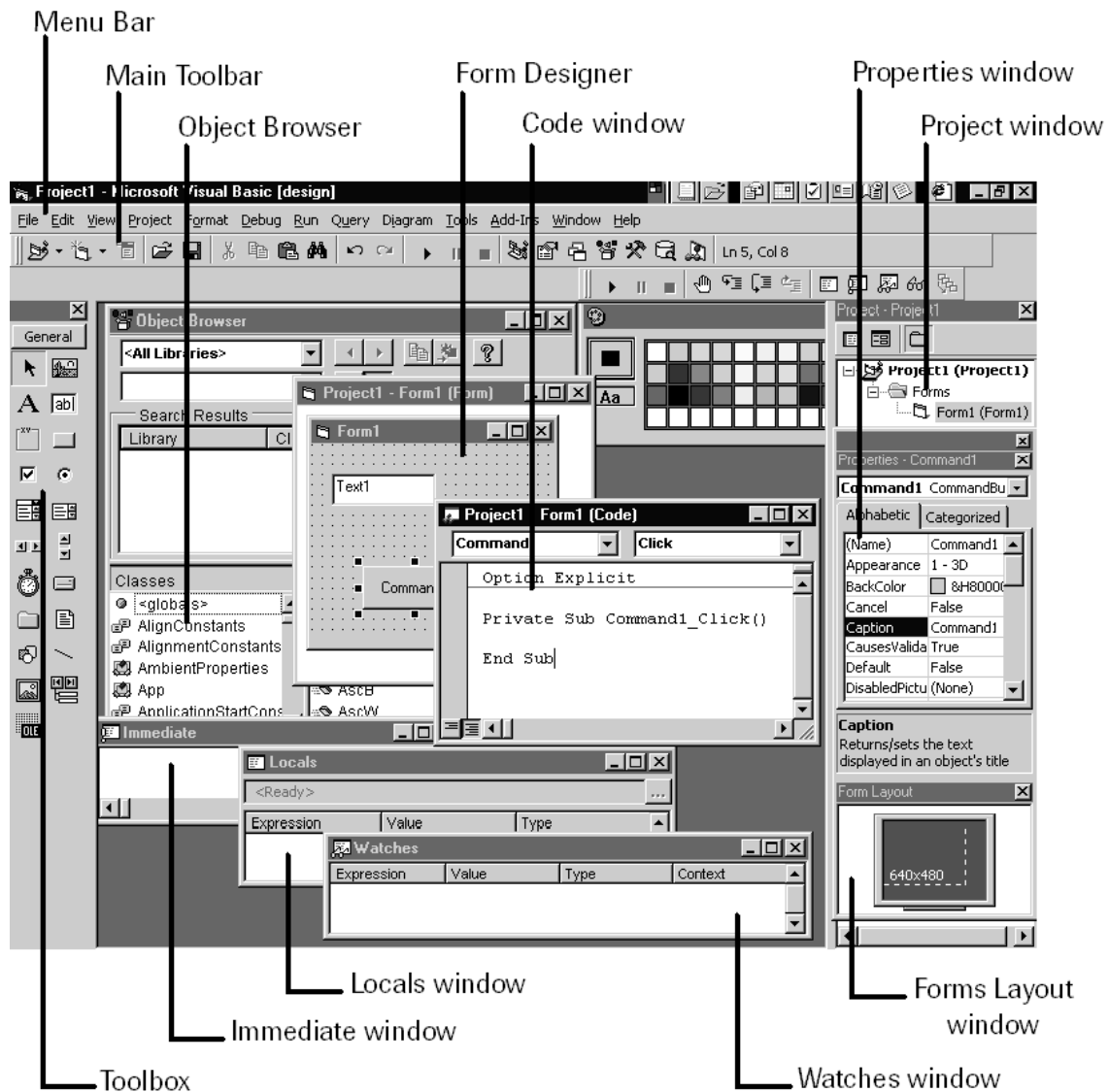
Gambar 1-2. Dialog box New Project ditampilkan sesaat anda menjalankan IDE Visual Basic 6.

Jendela IDE

IDE Visual Basic 6 menggunakan model MDI (Multiple Document Interface). Berikut ini adalah gambar yang menunjukkan bagian-bagian dan nama-nama jendela yang dapat tampil pada IDE Visual Basic. Mungkin pada IDE anda hanya ditampilkan sebagian

jendela dibandingkan dengan **Gambar 1-3** anda tidak perlu terkejut, karena penampilan dari jendela-jendela tersebut dapat diatur dengan perintah pada menu View.

Sebagai langkah awal dari proses belajar, tidak semua jendela akan kita gunakan, tetapi hanya beberapa yang penting, sedangkan yang lainnya bersifat khusus.



Gambar 1-3. IDE Visual Basic dengan jendela-jendela yang terbuka.

Sebagaimana dengan proses belajar ini, kita akan fokus pada beberapa jendela yang penting terlebih dahulu sehingga konsentrasi tidak menjadi pecah, dan peserta belajar menjadi bingung. Adapun jendela-jendela yang perlu anda perhatikan adalah sebagai berikut :

- *Menu Bar*, digunakan untuk memilih tugas-tugas tertentu seperti menyimpan project, membuka project, dll

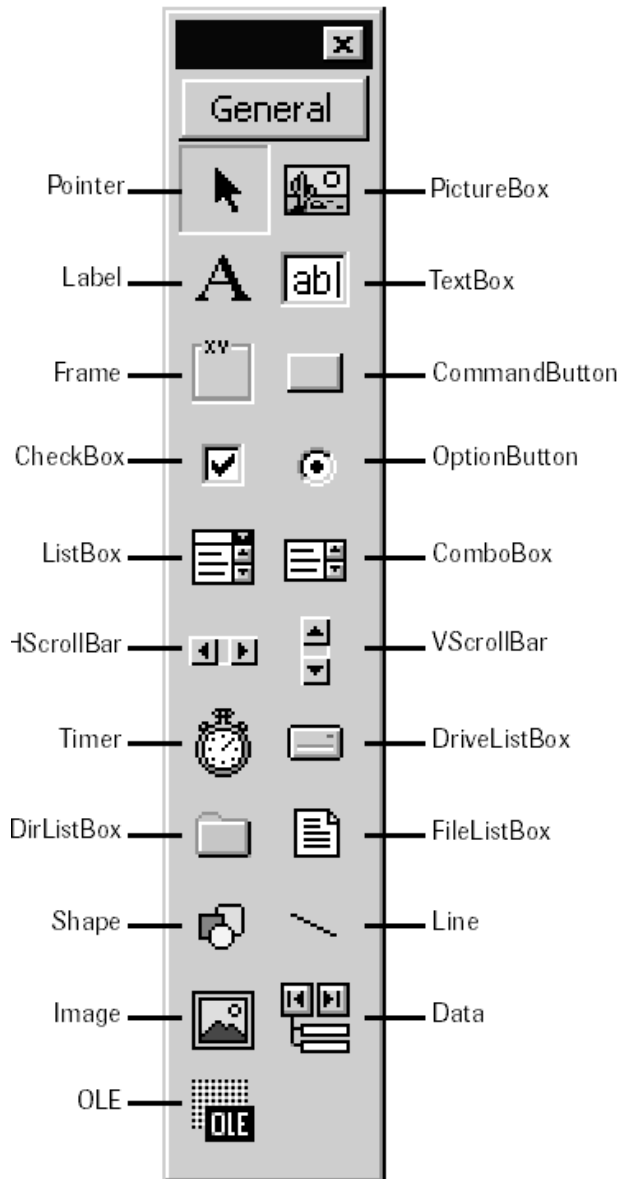
- *Main Toolbar*, digunakan untuk melakukan tugas-tugas tertentu dengan cepat.
- *Jendela Project*, jendela ini berisi gambaran dari semua modul yang terdapat dalam aplikasi anda. Anda dapat menggunakan icon Toggle Folders untuk menampilkan modul-modul dalam jendela tersebut secara di group atau berurut berdasarkan nama. Anda dapat menggunakan Ctrl+R untuk menampilkan jendela project, ataupun menggunakan icon Project Explorer.
- *Jendela Form Designer*, jendela ini merupakan tempat anda untuk merancang user interface dari aplikasi anda. Jadi jendela ini menyerupai kanvas bagi seorang pelukis.
- *Jendela Toolbox*, jendela ini berisi komponen-komponen yang dapat anda gunakan untuk mengembangkan user interface.
- *Jendela Code*, merupakan tempat bagi anda untuk menulis koding. Anda dapat menampilkan jendela ini dengan menggunakan kombinasi Shift-F7.
- *Jendela Properties*, merupakan daftar properti-properti object yang sedang terpilih. Sebagai contohnya anda dapat mengubah warna tulisan (foreground) dan warna latarbelakang (background). Anda dapat menggunakan F4 untuk menampilkan jendela properti.
- *Jendela Color Palette*, adalah fasilitas cepat untuk mengubah warna suatu object.
- *Jendela Form Layout*, akan menunjukkan bagaimana form bersangkutan ditampilkan ketika runtime.

Jika jendela-jendela tersebut tidak ada, anda dapat memunculkannya dengan **Menu View** dan pilih :

- Project Explorer (Ctrl+R)
- Properties Windows (F4)
- Form Layout Windows
- Property Pages (Shift+F4)
- Toolbox
- Color Pallette
- Toolbars

Toolbox

Jendela Toolbox merupakan jendela yang sangat penting bagi anda. Dari jendela ini anda dapat mengambil komponen-komponen (object) yang akan ditanamkan pada form untuk membentuk user interface.



Gambar 1-3. *Toolbox Visual Basic 6 dengan semua kontrol intrinsic.*

Adapun secara garis besar fungsi dari masing-masing intrinsic kontrol tersebut adalah sebagai berikut :

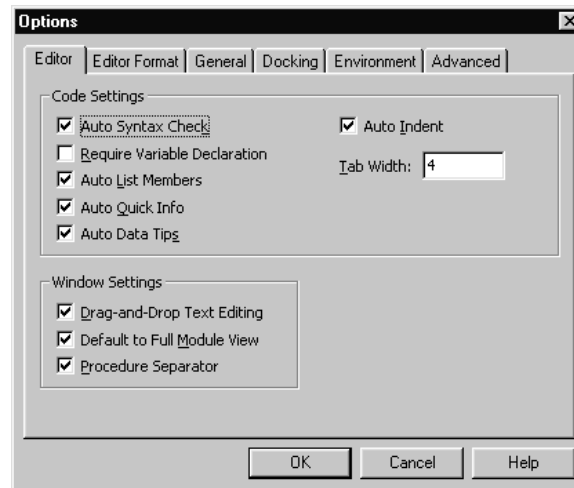
- *Pointer* bukan merupakan suatu kontrol; gunakan icon ini ketika anda ingin memilih kontrol yang sudah berada pada form.
- *PictureBox* adalah kontrol yang digunakan untuk menampilkan image dengan format: BMP, DIB (bitmap), ICO (icon), CUR (cursor), WMF (metafile), EMF (enhanced metafile), GIF, dan JPEG.
- *Label* adalah kontrol yang digunakan untuk menampilkan teks yang tidak dapat diperbaiki oleh pemakai.

- *TextBox* adalah kontrol yang mengandung string yang dapat diperbaiki oleh pemakai, dapat berupa satu baris tunggal, atau banyak baris.
- *Frame* adalah kontrol yang digunakan sebagai kontainer bagi kontrol lainnya.
- *CommandButton* merupakan kontrol hampir ditemukan pada setiap form, dan digunakan untuk membangkitkan event proses tertentu ketika pemakai melakukan klik padanya.
- *CheckBox* digunakan untuk pilihan yang isinya bernilai yes/no, true/false.
- *OptionButton* sering digunakan lebih dari satu sebagai pilihan terhadap beberapa option yang hanya dapat dipilih satu.
- *ListBox* mengandung sejumlah item, dan user dapat memilih lebih dari satu (bergantung pada property *MultiSelect*).
- *ComboBox* merupakan kombinasi dari *TextBox* dan suatu *ListBox* dimana pemasukkan data dapat dilakukan dengan pengetikkan maupun pemilihan.
- *HScrollBar* dan *VScrollBar* digunakan untuk membentuk scrollbar berdiri sendiri.
- *Timer* digunakan untuk proses background yang diaktifkan berdasarkan interval waktu tertentu. Merupakan kontrol non-visual.
- *DriveListBox*, *DirListBox*, dan *FileListBox* sering digunakan untuk membentuk dialog box yang berkaitan dengan file.
- *Shape* dan *Line* digunakan untuk menampilkan bentuk seperti garis, persegi, bulatan, oval.
- *Image* berfungsi menyerupai image box, tetapi tidak dapat digunakan sebagai kontainer bagi kontrol lainnya. Sesuatu yang perlu diketahui bahwa kontrol image menggunakan resource yang lebih kecil dibandingkan dengan *PictureBox*
- *Data* digunakan untuk *data binding*
- *OLE* dapat digunakan sebagai tempat bagi program eksternal seperti Microsoft Excel, Word, dll.

Mengatur Lingkungan Kerja Visual Basic

Pengaturan IDE Visual Basic dapat dilakukan dengan menu Tools, Option, pengaturan dapat dilakukan sesuai dengan selera dan kebiasaan programmer sehingga dapat bekerja dengan baik dan efektif. Mulai Visual Basic 5.0, IDE Visual Basic memperkenalkan MDI Development Environment, dan beberapa hal dapat diatur dengan menggunakan menu Tools, Option adalah sebagai berikut :

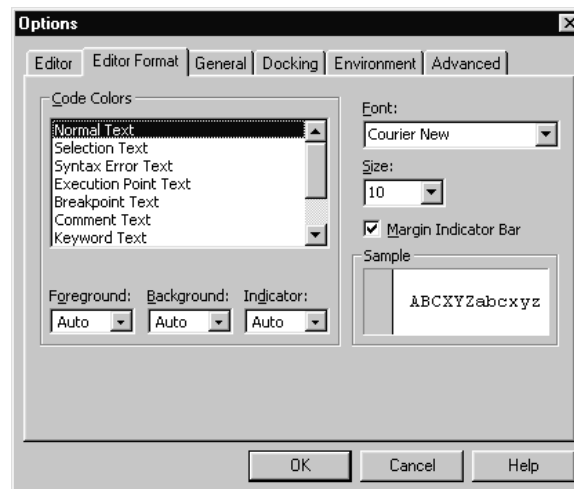
- **Mengatur Editor**



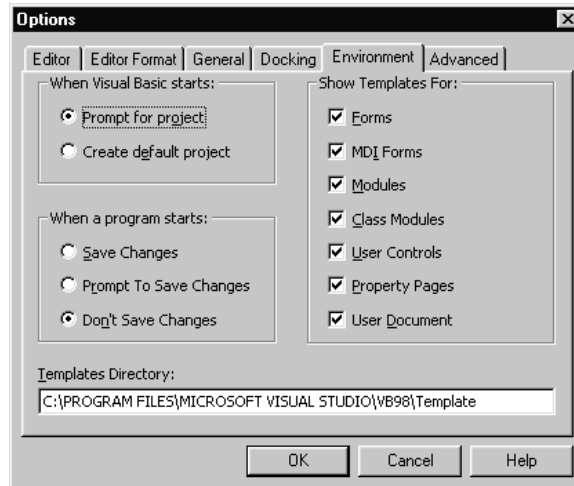
Catatan penulis:

Sebagai programmer pemula, penulis menyarankan agar senantiasa mengaktifkan Require Variable Declaration yang berarti bahwa setiap variabel yang dipakai dalam program yang akan kita buat harus dideklarasikan terlebih dahulu.

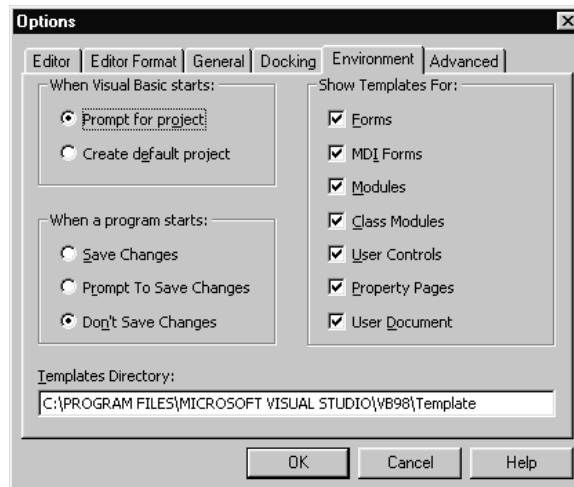
- **Mengatur Format Editor**



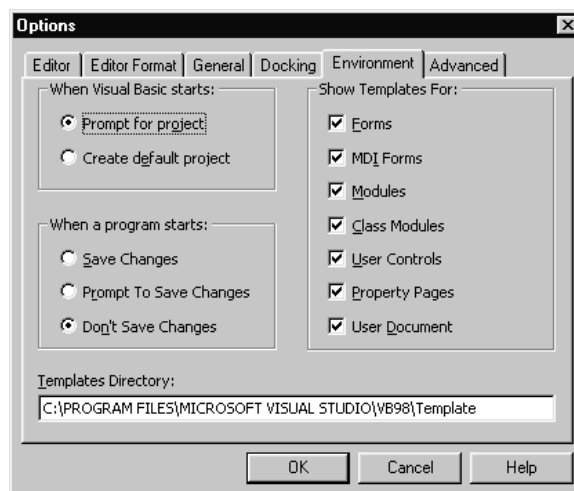
- **Mengatur hal-hal yang General**



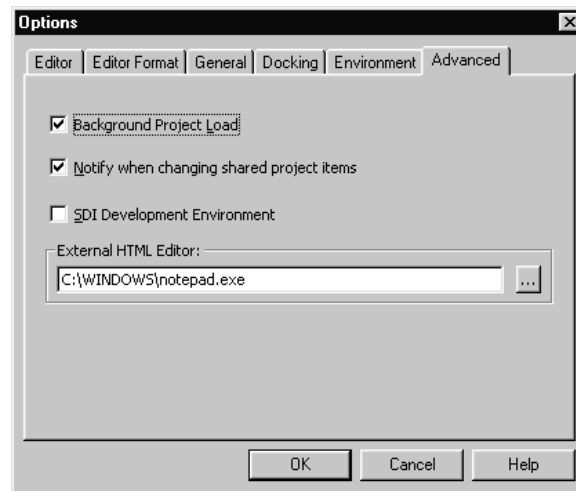
- Mengatur Docking Jendela



- Mengatur Environment



- Mengatur SDI Development Environment atau MDI



Catatan Penulis :

Pada latihan berikut ini mengandung pembelajaran step by step, tetapi sebelum anda memulai latihan ini sebaiknya anda membaca semua tulisan modul ini. Penulis tidak akan berpanjang lebar dengan teori-teori pemrograman, tetapi sebaiknya kita dapat langsung bekerja dan memahami dasar-dasar dari Visual Basic.

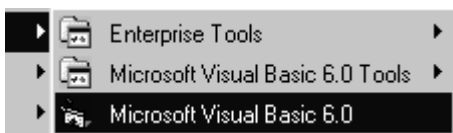
Semua latihan ini dirancang berdasarkan pengalaman penulis dalam pengajaran untuk proses pembelajaran yang efektif

Latihan 1

Judul : Mengenal IDE Visual Basic

Petunjuk, sebaiknya lembar latihan ini anda cetak ke printer.

1. Aktifkan IDE Visual Basic anda dengan memilih **Start**, kemudian **Programs**, kemudian **Microsoft Visual Basic 6.0**, dan **Microsoft Visual Basic 6.0**.

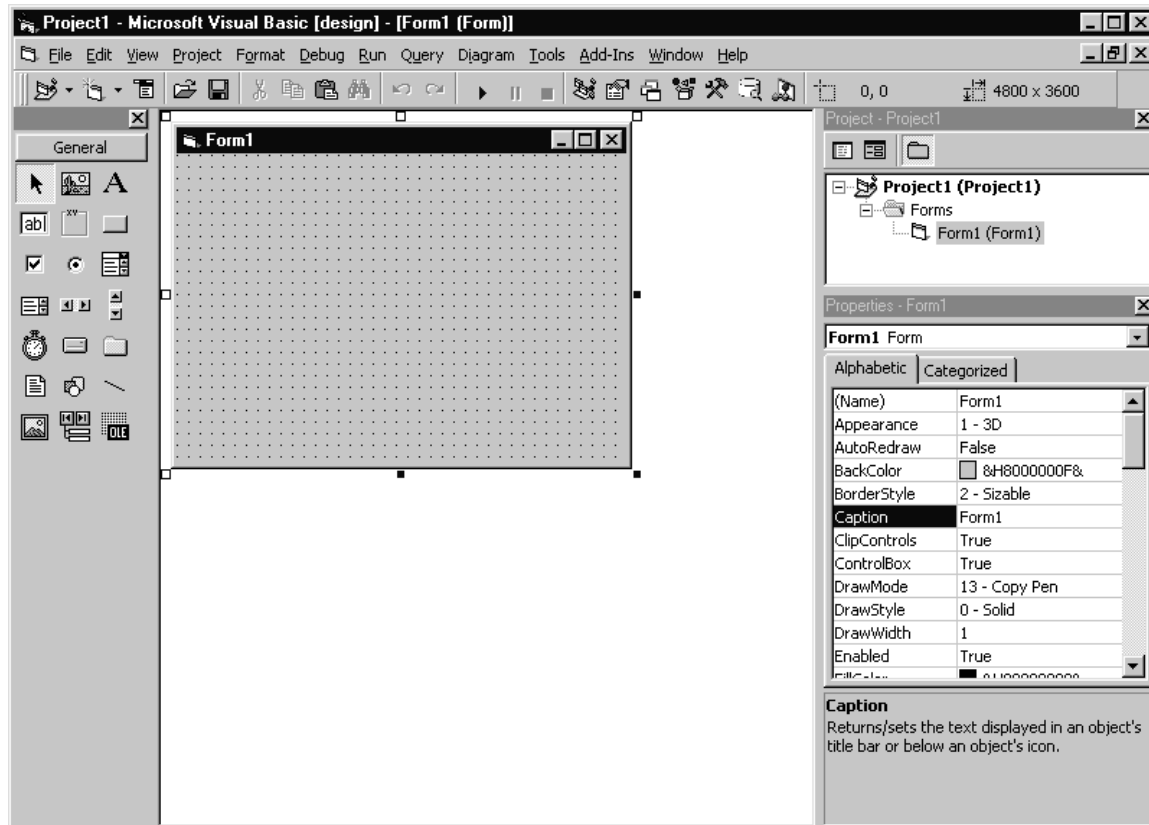


2. Pada dialog box **New Project**, pilihlah **Standard EXE**, kemudian klik pada **Open**



3. Pada hadapan anda akan muncul IDE Visual Basic, dan kenali masing-masing bagian berikut :

- Menu Bar
- Main Toolbar
- Jendela Toolbox
- Jendela Form
- Jendela Project Explorer
- Jendela Properties

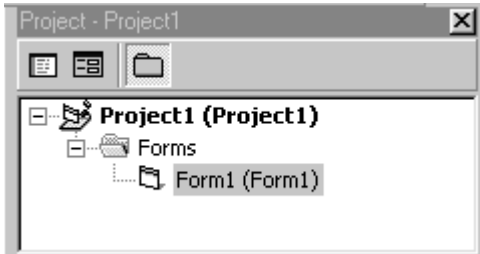


4. Sekarang Perhatikan Jendela Toolbox, dan kenali kontrol-kontrol berikut :



- Pointer
- PictureBox
- Label
- Textbox
- Frame
- CommandButton
- Checkbox
- OptionButton
- Combobox
- Listbox
- Image

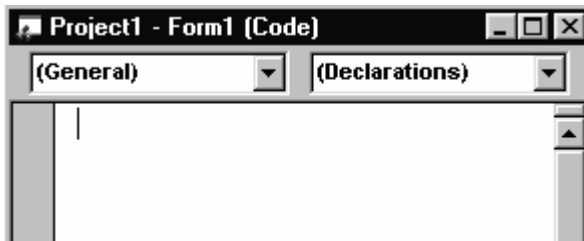
5. Sekarang Perhatikan Jendela Project Explorer



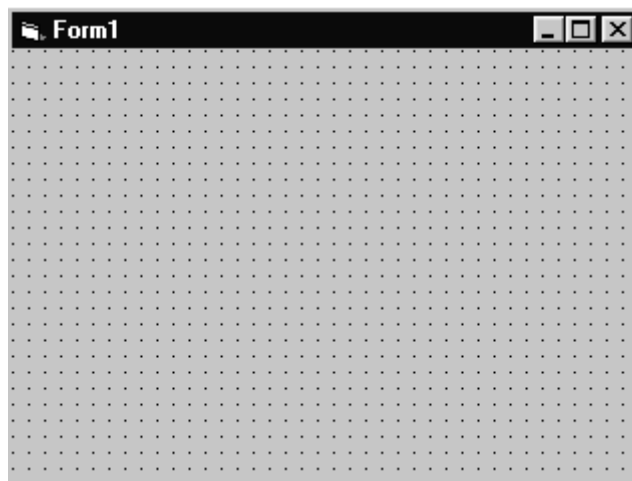
Perhatikan Icon-icon berikut :

Icon	Fungsi
	View Code
	View Object
	Toggle Folder

Lakukan klik pada **Form1 (Form1)**, dan klik pada icon **View Code**, maka akan ditampilkan Jendela Code



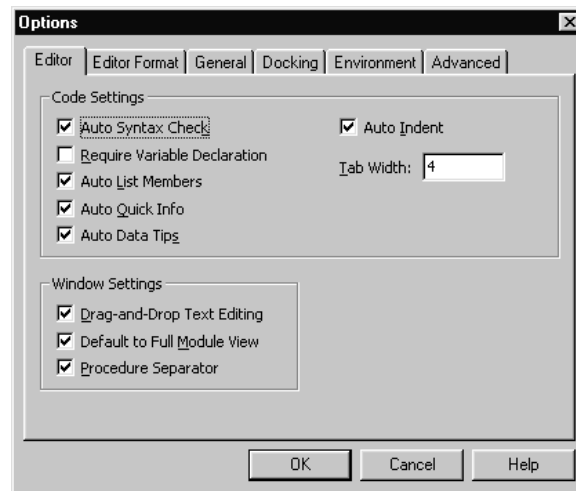
Lakukan klik pada icon **View Object**, maka akan ditampilkan Form1



6. Periksalah setting IDE Visual Basic, dan aktifkan hal-hal berikut pada menu Tools, Option :

- *Auto Syntax Check*, secara otomatis memeriksa kebenaran pengetikan program.
- *Require Variable Declaration*, secara otomatis menambahkan perintah Option Explicit pada form dan module.

- *Auto List Member*, secara otomatis menampilkan syntax penulisan perintah dan fungsi



Keluar dari IDE Visual Basic, dengan memilih menu File, pilih Exit, atau menekan Alt+Q

Modul 2, Bahasa Visual Basic Application

Dasar dari pemrograman pada Visual Basic adalah VBA yang menggunakan dialek Basic. Bagi anda yang pernah belajar bahasa Basic, tidak akan sulit untuk belajar VBA.

Jadi penguasaan terhadap VBA merupakan syarat mutlak bagi programmer yang ingin mendalami pemrograman Visual Basic.

Type Variabel

Dibandingkan dengan type data yang terdapat pada bahasa basic, maka pada VBA, type data yang disediakan lebih banyak, seperti type Currency, Decimal, Object, dan Variant.. Variant merupakan type variabel yang istimewa, karena dapat berubah dari satu type ke type yang lain, sesuai dengan evaluasi ekspresi oleh Visual Basic.

Ketepatan pemilihan type variabel akan sangat menentukan pemakaian resources oleh aplikasi yang dihasilkan, adalah tugas programmer untuk memilih type yang sesuai untuk menghasilkan program yang efisien dan berperformace tinggi.

Type Data	Ukuran Storage	Jangkauan
Byte	1 byte	0 s/d 255
Boolean	2 byte	True atau False
Integer	2 byte	-32,768 s/d 32767
Long	4 byte	-2,147,483,648 s/d 2,147,483,647
Single	4 byte	-3.402823E38 s/d -1.401298E-45 (-) 1.401298E-45 s/d 3.402823E38 (+)
Double	8 byte	-1.79769313486232E308 s/d -4.94065645841247E-324 (-)
Currency	8 byte	-922,337,203,685,477.5808 s/d 922,337,203,685,477.5807
Decimal	14 byte	+/-79,228,162,514,264,337,593,543,950,335
Date	8 byte	1 Januari 100 s/d 31 Desember 9999
Object	4 byte	Mengacu pada objek tertentu
String (panjang variabel)	10 byte + panjang string	0 sampai lebih kurang 2 milyar
String (panjang tetap)	panjang dari string	1 sampai lebih kurang 65,400
Variant (dengan angka)	16 byte	Sembarang angka sampai jangkauan jenis Double
Variant (dengan karakter)	22 byte + panjang string	Sama dengan jangkauan variabel String

Tabel 1-1. Type variabel, pemakaian storage dan jangkauan masing-masing

Catatan penulis :

Sebagai awal proses belajar, anda tidak perlu terlalu dipusingkan oleh banyaknya tipe data yang disediakan oleh Visual Basic, yang penting adalah kalau anda membutuhkan tipe data berupa kondisi benar (true) atau salah (false), anda dapat menggunakan tipe Boolean. Kalau anda membutuhkan bilangan bulat (integer), anda dapat memilih salah satu dari byte, integer, long sesuai dengan jangkauan yang dibutuhkan, sedangkan kalau anda membutuhkan tipe pecahkan (floating point) anda dapat memilih single, double, dan decimal sesuai dengan ketelitian yang dibutuhkan. Khusus untuk uang dapat digunakan

currency, khusus untuk tanggal dan waktu dapat digunakan date, dan untuk kalimat dapat digunakan string.

Ada apa dengan Variant ?

Variant merupakan tipe data khusus yang sangat digemari oleh programmer pemula, karena variabel dengan tipe data ini dapat menampung data dari semua tipe variabel yang ada tanpa menyebabkan kesalahan pada saat kompilasi maupun runtime. Tetapi penulis menyarankan untuk tidak terlena dengan tipe data ini, karena pada pengembangan selanjutnya (VB.net), tipe data ini tidak didukung lagi, karena tipe ini dapat menyebabkan bugs jika program telah berkembang menjadi besar.

Operator Pada Visual Basic dan urutan operasinya

Visual basic menyediakan operator aritmatika, komparasi dan logika, salah satu hal yang harus dipahami oleh programmer adalah tata urutan operasi dari masing-masing operator tersebut sehingga mampu membuat ekspresi yang akan menghasilkan nilai yang benar, Tabel 1-2, menunjukkan operator dan urutan operasinya dari atas kebawah.

Contoh :

A = 1 + 2 * 3 'Akan menghasilkan 7

B = (1 + 2) * 3 'Akan menghasilkan 9

Aritmatika	Komparasi	Logika
Pangkat (^)	Sama (=)	Not
Negatif (-)	Tidak sama (<>)	And
Kali dan Bagi (*, /)	Kurang dari (<)	Or
Pembagian bulat (\)	Lebih dari (>)	Xor
Sisa Bagi (Mod)	Kurang dari atau sama (<=)	Eqv
Tambah dan Kurang (+,-)	Lebih dari atau sama (>=)	Imp
Pengabungan String (&)	Like	

Tabel 1-2. Operator pada Visual Basic dan urutan operasi dari atas ke bawah

Catatan penulis :

Salah satu bugs yang sering terjadi pada programmer pemula adalah kesalahan dalam pemakaian operator karena kurangnya memahami urutan operasi.

Operator Like

Salah satu operator yang menarik untuk dibahas adalah operator like, karena operator ini tidak tersedia pada bahasa BASIC. Operator digunakan untuk operasi pencocokan pola pada string yang akan sangat membantu programmer.

Syntax :

result = string Like pattern

Karakter dalam pola	Penyamaan dalam string
?	Sembarang karakter tunggal
*	Nol atau lebih karakter
#	Sembarang digit tunggal (0-9)
[charlist]	Sembarang karakter yang berada dalam charlist
[!charlist]	Sembarang karakter yang tidak berada dalam charlist

Tabel 1-3. *Character dalam pencocokan pola pada operator Like*

Contoh :

- MyCheck = "aBBBa" Like "a*a" ' Returns True.
- MyCheck = "F" Like "[A-Z]" ' Returns True.
- MyCheck = "F" Like "[!A-Z]" ' Returns False.
- MyCheck = "a2a" Like "a#a" ' Returns True.
- MyCheck = "aM5b" Like "a[L-P]#[!c-e]" ' Returns True.
- MyCheck = "BAT123khg" Like "B?T*" ' Returns True.
- MyCheck = "CAT123khg" Like "B?T*" ' Returns True.

Deklarasi Variabel

Visual Basic memungkinkan kita untuk menggunakan variabel tanpa deklarasi. Tetapi hal ini adalah kurang baik untuk praktek pemrograman yang terstruktur dan menghindari kesalahan pengolahan yang diakibatkan oleh kesalahan dalam pengetikkan nama variabel.

Agar setiap variabel yang digunakan harus dideklarasikan, dapat digunakan perintah :

Option Explicit

Pada setiap awal module, atau pada menu Tools, pilih Option, pilih tab Editor, buat tanda check pada Require Variable Declaration.

Deklarasi variabel pada Visual Basic dapat dilakukan dengan Keyword berikut :

Keyword	Digunakan pada
Public	Berlaku pada level modul
Private	Berlaku pada level modul
Dim	Berlaku pada level modul dan level procedure
Static	Berlaku pada level procedure

Tabel 1-4. *Keyword pada deklarasi variabel*

Syntax :

<keyword> NamaVariabel [**As** TypeVariabel]

Contoh :

Dim Nama As String

Catatan penulis :

Penjelasan tentang Public, Private, dan Static akan menjadi lebih mudah setelah anda mempelajari topik Module, Sub dan Function.

Konvensi penamaan dalam Visual Basic

Ketika anda menulis Code Visual Basic, anda mendeklarasikan banyak elemen (Sub dan Function procedures, variables, constants, dan lainnya). Nama dari procedure, variabel, dan konstanta yang mana anda deklarasikan pada Visual Basic harus mengikuti petunjuk berikut :

- Harus dimulai dengan suatu Huruf
- Tidak dapat mengandung titik atau spesial karakter
- Tidak dapat lebih dari 255 huruf, nama dari kontrol, form, class, dan module tidak dapat melebihi 40 karakter.
- Tidak dapat sama dengan keywords yang tercadang.

Keyword yang tercadang adalah kata yang Visual Basic gunakan sebagai bagian dari bahasanya. Ini terdiri dari predefined statements (seperti If dan Loop), function (seperti Len dan Abs), dan operator (seperti Or dan Mod).

Deklarasi Variabel

- Deklarasi variabel pada bagian deklarasi (general declaration) di suatu form, standard, atau class module, dari pada dalam suatu procedure, membuat variabel itu berlaku untuk semua procedure dan function dalam module tersebut
- Deklarasi variabel dengan menggunakan keyword Public membuatnya berlaku pada keseluruhan aplikasi anda.
- Deklarasi suatu variabel lokal dengan menggunakan keyword Static akan menyimpan nilainya ketika suatu procedure berakhir.

Deklarasi Implicit

Anda tidak perlu mendeklarasikan suatu variabel sebelum menggunakan, tetapi fasilitas ini tidak disarankan oleh penulis, karena dapat mengakibatkan bug-bug tersembunyi

karena pemakaian variabel yang tidak konsisten dan kemungkinan kesalahan pengetikan nama variabel.

Defaultnya Visual Basic menggunakan Deklarasi Implicit.

Deklarasi Explicit

Untuk mencegah kesalahan menetik variabel, dan Visual Basic akan selalu memberikan peringatan jika menemukan nama yang tidak dideklarasikan terlebih dahulu sebagai suatu variabel.

Catatan :

Pernyataan Option Explicit hanya bekerja per-module, sehingga harus diletakkan pada bagian deklarasi pada setiap form, dan class module yang mana anda ingin Visual Basic memaksakan suatu explicit variabel deklarasi

Mengenal Struktur Kendali

Struktur kendali memungkinkan anda untuk mengatur jalannya program anda, Jika membiarkan tanpa di periksa oleh statement control-flow, suatu logika program akan berjalan dari kiri ke kanan dan dari atas kebawah. Hanya program yang sangat sederhana dapat ditulis tanpa statement control-flow.

Struktur Keputusan

Struktur keputusan yang didukung oleh Visual Basic adalah sebagai berikut :

If...Then

Gunakan suatu struktur If...Then untuk menjalankan suatu pernyataan secara kondisional. Anda dapat menggunakan syntax satu baris ataupun syntax banyak baris :

If kondisi Then pernyataan

atau

```
If kondisi Then
    pernyataan-pernyataan
End If
```

Kondisi biasanya berupa suatu perbandingan, maupun ekspresi yang menghasilkan nilai numerik. Visual Basic menginterpretasikan False sebagai nol (0), dan True sebagai bukan nol.

If...Then...Else

Gunakan If...Then...Else untuk mendefinisikan beberapa blok pernyataan yang akan dijalankan salah satu berdasarkan kondisi yang memenuhi syarat

```
If kondisi1 Then
    [blok pernyataan-1]
[ElseIf kondisi2 Then
    [blok pernyataan-2]] ...
[Else
    [blok pernyataan-n]]
```

End If

Visual Basic awalnya akan mencoba kondisi1. Jika False, maka Visual Basic akan memeriksa kondisi2, dan seterusnya sampai menemukan suatu kondisi True untuk dijalankan blok pernyataannya.

Contoh :

```
If JlhRec > 0 Then
    Posisi = 1
Else
    Posisi = 0
End If
```

Select Case

Visual Basic menyediakan struktur Select Case sebagai suatu alternatif terhadap If...Then...Else. Suatu Select Case statement memiliki kemampuan yang sama dengan If...Then...Else..., tetapi membuat code lebih mudah dibaca.

Struktur Select Case bekerja dengan suatu percobaan tunggal yang hanya dievaluasi satu kali pada bagian atas struktur. Visual Basic then membandingkan hasil ekspresi dengan nilai pada setiap Case didalam struktur tersebut, jika ada yang sesuai, akan dijalankan blok statement yang sesuai

```
Select Case ekspresiyangdicoba
    [Case ekspresi1
        [blokpernyataan-1]]
    [Case ekspresi2
        [blokpernyataan-2]]
    ...
    [Case Else
        [blokpernyataan-n]]
```

End Select

Contoh :

```
Select Case x
  Case 0: angka = "Nol"
  Case 1:
    If posisi = 1 Then
      angka = "Satu "
    Else
      angka = "Se"
    End If
  Case 2: angka = "Dua "
  Case 3: angka = "Tiga "
  Case 4: angka = "Empat "
  Case 5: angka = "Lima "
  Case 6: angka = "Enam "
  Case 7: angka = "Tujuh "
  Case 8: angka = "Delapan "
  Case 9: angka = "Sembilan "
  Case 10: angka = "Sepuluh "
  Case 11: angka = "Sebelas "
  Case 12: angka = "Duabelas "
  Case 13: angka = "Tigabelas "
  Case 14: angka = "Empatbelas "
  Case 15: angka = "Limabelas "
  Case 16: angka = "Enambelas "
  Case 17: angka = "Tujuhbelas "
  Case 18: angka = "Delapanbelas "
  Case 19: angka = "Sembilanbelas "
End Select
```

Deklarasi Konstanta

Anda sering menjumpai suatu kode yang mengandung nilai yang berulang-ulang ataupun nilai yang memiliki arti khusus, atau anda menemukan bahwa angka tersebut susah diingat. Dalam hal ini anda dapat membuat code anda menjadi lebih mudah dibaca-lebih mudah ditangani-dengan menggunakan suatu konstanta. Suatu konstanta adalah nama yang menyimpan dari suatu nilai yang tidak dapat berubah. Ada dua sumber dari suatu konstanta :

- Intrinsic atau System-defined konstanta yang disediakan oleh suatu aplikasi atau control. Konstanta Visual Basic terdaftar pada Visual Basic (VB), Visual Basic for Application (VBA), dan Data Access (DAO).
- Symbolic atau User-defined konstanta adalah dideklarasikan dengan menggunakan statement Const.

Membuat Konstanta Anda Sendiri

Syntax untuk deklarasi konstanta adalah :

[Public|Private] Const constantname [As type] = expression

Contoh :

```
Const Pi = 3.12159265358979
Public Const Planet As Integer = 9
Const ReleaseDate = #7/1/95#
```

Scope dari suatu user-defined konstanta

- Membuat suatu konstanta yang mana hanya ada dalam suatu procedure, deklarasikan di dalam procedure tersebut
- Membuat suatu konstanta berlaku pada semua procedure dalam suatu module, deklarasikan dia pada bagian deklarasi di module tersebut
- Membuat suatu konstanta berlaku pada semua aplikasi, deklarasikan dia dengan keyword Public sebelum kata Const.

Struktur Perulangan

Struktur loop memperbolehkan anda untuk melaksanakan sekelompok baris lebih dari satu kali :

Do...Loop

Gunakan Do loop untuk mengeksekusi suatu blok statement dengan jumlah perulangan yang tak terhingga, ada beberapa Do...Loop statement, tetapi masing-masing mengevaluasi suatu kondisi untuk menentukan apakah melanjutkan eksekusi

Berikut ini adalah statement Do...Loop, yang dijalankan selama kondisi benar :

```
Do While condition
    statements
Loop
```

Ketika Visual menjalankan Do loop ini, pertama kali akan di coba kondisinya, jika kondisi False (zero), akan diloncati semua statements yang mengikuti kondisi tersebut. Visual Basic akan menjalankan statements jika kondisi benar dan kembali ke Do...Loop berikutnya.

Function Faktorial (x)

```
Dim Hasil As Double
Dim Count As Long
Hasil = 1
count = 2
Do While Count <= x
    Hasil = Hasil * count
    count = count + 1
Loop
```

```
Faktorial = count  
End Function
```

Variasi lain dari statement Do...Loop, yang menjamin minimal satu kali statement dijalankan :

```
Do  
    statements  
Loop While condition
```

Dua variasi yang analog dengan dua contoh sebelumnya adalah perulangan minimal satu kali. Dan berikut ini adalah berulang atau tidak sama sekali :

```
Do Until condition  
    statements  
Loop  
Do  
    statements  
Loop Until condition
```

For...Next

Do loops bekerja dengan baik, ketika anda tidak tahu berapa banyak kali untuk butuhkan untuk menjalankan statement. Ketika anda mengetahui harus menjalankan statement sejumlah kali, bagaimanapun For...Next adalah pilihan yang lebih baik. Tidak seperti Do Loop, For...loop menggunakan suatu variabel yang disebut counter yang mana akan bertambah atau berkurang pada setiap perulangan. Syntaxnya adalah :

```
For counter = start To end [Step increment]  
    statements  
Next [counter]
```

Argumen counter, start, end, dan increment semuanya adalah numerik

1. Dalam melakukan For loop, Visual Basic :
2. Menyet nilai counter sama dengan start
3. Mencoba apakah counter lebih besar dari end. Jika ya, Visual Basic meninggalkan looping tersebut.
4. Menjalankan statements tersebut.
5. Menaikkan counter dengan 1 atau dengan nilai step yang ditentukan
6. Mengulangi langkah 2 sampai 4

Code berikut akan mencetak semua item yang terdapat pada suatu daftar list.

```
Private Sub Form_Click ()  
Dim I As Integer  
For i = 0 To List1.ListCount-1  
    Print List1.List(I)  
Next  
End Sub
```

Struktur Kendali Nested

Anda dapat meletakkan suatu struktur kendali didalam struktur kendali lainnya, yang dikenal dengan istilah nested. Struktur kendali pada Visual Basic dapat nested sebanyak level yang anda inginkan.

Meninggalkan suatu Struktur Kendali

Statement Exit memperbolehkan anda untuk keluar langsung dari suatu For loop, Do Loop, syntax untuk statement Exit adalah sederhana : Exit For dapat muncul sebanyak mungkin dalam suatu For. Loop, demikian juga Exit Do untuk Do loop.

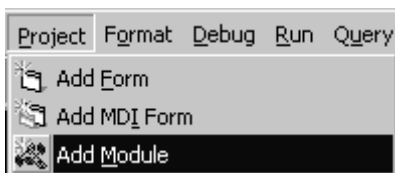
```
For counter = start To end [Step increment]
    [statementblock]
    [Exit For]
    [statementblock]
Next [counter[, counter] [,...]]

Do [{While | Until} condition]
    [statementblock]
    [Exit Do]
    [statementblock]
Loop
```

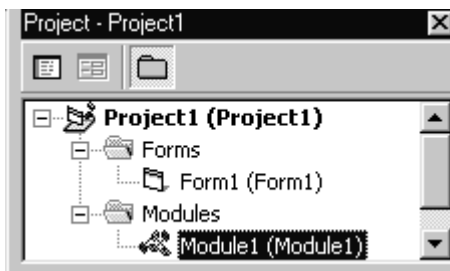
Statement Exit Do bekerja pada semua versi syntax Do loop. Exit For dan Exit Do adalah kadang-kadang perlu untuk keluar dari suatu loop dengan segera, tanpa melakukan iterasi yang berikutnya.

Module Pada Visual Basic

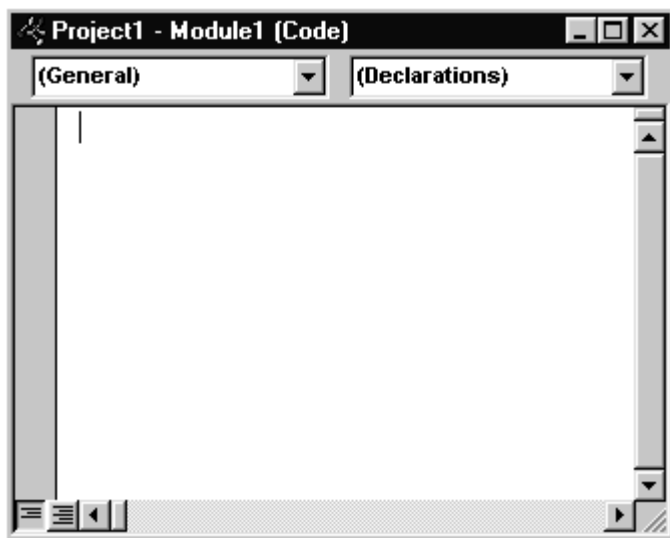
Visual Basic menyediakan module yang dapat digunakan untuk memuat fungsi,subrutin, konstanta, variabel dan type ciptaan yang akan disediakan bagi keseluruhan project. Untuk membuat module dalam project anda dapat menggunakan perintah **Project, Add Module**.



Sesaat setelah anda menambahkan module, maka pada Jendela Project Explorer akan nampak Module1



Untuk melakukan koding, anda dapat melakukan double klik pada Module tersebut.



Deklarasi Variabel Pada Module

Pada Module anda dapat mendeklarasikan variabel dengan awalan DIM, PRIVATE maupun PUBLIC, dimana awalan DIM dan PRIVATE anda membentuk variabel Module Level (hanya berlaku didalam pemakaian module bersangkutan), sedangkan awalan PUBLIC akan menghasilkan variabel Global yang akan berfungsi bagi keseluruhan Program.

Contoh :

Dim A as Integer	'Variabel A adalah Module Level
Private B as Integer	'Variabel B adalah Module Level
Public C as Integer	'Variabel C dapat digunakan oleh program keseluruhan

Deklarasi Fungsi dan Sub

Sub dan Function yang dideklarasikan dalam suatu Module bersifat Global bagi keseluruhan program, kecuali kalau diawali dengan awalan PRIVATE.

Contoh :

Sub Tengah (x)

```
x.Move (Screen.Width - x.Width) \ 2, _  
      (Screen.Height - x.Height) \ 2
```

End Sub

Sub Tengah diatas berlaku untuk program secara keseluruhan.

Mengenal Procedure

Procedure digunakan untuk memadatkan tugas-tugas berulang ataupun proses yang digunakan bersama, seperti perhitungan yang sering dilakukan, text dan manipulasi kontrol, serta operasi database.

Secara umum ada dua keuntungan dengan pemakaian procedure dalam program :

1. Procedure memungkinkan anda untuk memecahkan program anda pada unit logika yang lebih kecil, sehingga anda dapat dengan mudah melakukan proses debug dari pada jika keseluruhan program tanpa menggunakan procedure.
2. Procedure yang digunakan dalam suatu program dapat bertindak sebagai suatu blok yang dibangun untuk program lain dengan sedikit perubahan atau tanpa perubahan sama sekali.

Ada beberapa jenis procedure yang digunakan dalam Visual Basic

- Sub procedure yang tidak mengembalikan nilai
- Function procedure yang mengembalikan nilai
- Property procedure yang dapat mengembalikan nilai dan diisi nilai yang mengacu pada suatu objek.

Sub Procedure

Syntax penulisan Sub procedure:

[Private|Public][Static]Sub *namaprocedure* (*argumen-argumen*)

pernyataan-pernyataan

End Sub

Setiap kali procedure dipanggil, maka pernyataan-pernyataan yang berada di antara Sub dan End Sub akan dijalankan.

Argumen pada procedure adalah nilai yang akan dilewatkan saat pemanggilan procedure.

Di Visual Basic Sub Procedure dapat dibagi atas dua yaitu :

- **General Procedure**

, procedure yang diaktifkan oleh aplikasi

- **Event Procedure**

, Procedure yang diaktifkan oleh system sebagai respon terhadap event.

Contoh, sub Tengah yang dapat digunakan untuk menampilkan form ketengah Layar, dimana x adalah parameter yang merupakan form yang akan dibuat ketengah layar.

```

Sub Tengah(x As Form)
    x.Top = (Screen.Height - x.Height) \ 2
    x.Left = (Screen.Width - x.Width) \ 2
End Sub
Private Sub Form_Load()
    Call Tengah(Me)
End Sub
    
```

Function Procedure

Pada Visual Basic telah tersedia berbagai fungsi bawaan seperti Sqr, Cos, dan Chr, tetapi fungsi-fungsi yang tersedia tersebut bersifat umum dan kadang-kadang tidak memenuhi kebutuhan programer, untuk keperluan tersebut anda dapat menciptakan fungsi-fungsi sendiri yang dikenal dengan Function procedure.

Adapun syntax penulisan function procedure :

[Private|Public][Static]Function *namaprocedure* (*argumen-argumen*) [**As type**]

statements

End Function

Ada tiga perbedaan antara function dan procedure :

- Umumnya anda dapat memanggil suatu function dengan mengikutkan nama function sisi kanan dari statement atau ekspresi. (returnvalue = function()).
- Function memiliki type data seperti suatu variabel. Ini menentukan type yang dari nilai yang dikembalikan.
- Nilai kembali dimasukkan ke namafunction itu sendirinya, dan suatu function dapat menjadi bagian dari suatu ekspresi yang panjang.

Contoh Fungsi ciptaan untuk mengembalikan nama bulan dari suatu tanggal dalam bahasa Indonesia.

Function Bulan(x As Date)

```
Dim sRet As String
```

```
Select Case Month(x)
```

```
    Case 1: sRet = "Januari"
```

```
    Case 2: sRet = "Februari"
```

```
    Case 3: sRet = "Maret"
```

```
    Case 4: sRet = "April"
```

```
    Case 5: sRet = "Mei"
```

```
    Case 6: sRet = "Juni"
```

```
    Case 7: sRet = "Juli"
```

```
    Case 8: sRet = "Agustus"
```

```
    Case 9: sRet = "September"
```

```
    Case 10: sRet = "Oktober"
```

```
    Case 11: sRet = "Nopember"
```

```
    Case 12: sRet = "Desember"
```

```
    Case Else
```

```
        sRet = "tidak sah"
```

```
    End Select
```

```
Bulan = sRet
```

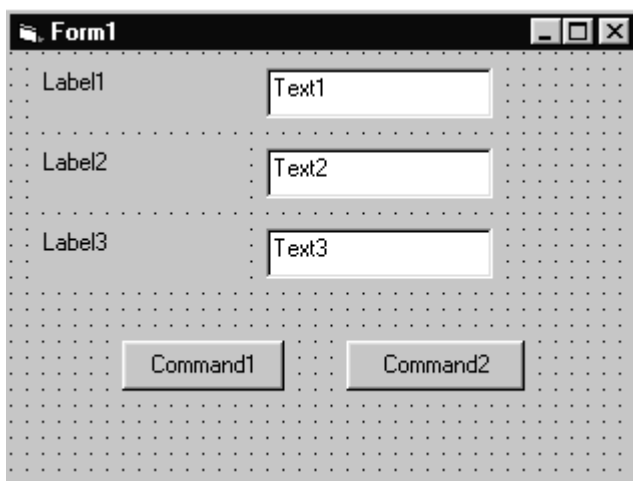
```
End Function
```

Latihan 2

Judul : Project anda yang pertama

Petunjuk, sebaiknya lembar latihan ini anda cetak ke printer.

1. Aktifkan IDE Visual Basic dan pilih **Standard EXE**.
2. Pada form1 tanamkan kontrol label (Label1, Label2, Label3) dan kontrol text (Text1, Text2, Text3), serta CommandButton (Command1 dan Command2)



Caranya :

Pada jendela toolbox, klik pada kontrol label, kemudian gambarkan pada Form1, lakukan hal sama untuk Label2 dan Label3, demikian juga untuk TextBox dan CommandButton.

Anda dapat juga melakukan dengan cara double klik pada kontrol label pada toolbox dan kemudian dengan drag & drop memindahkan posisinya pada form1

Catatan : Kontrol, kontrol tersebut ditanamkan satu per satu, dan jangan menggunakan cara Copy dan Paste karena akan menyebabkan kontrol-kontrol tersebut menjadi kontrol array.

3. Aturilah properti-properti dari masing-masing kontrol sehingga menjadi sebagai berikut:

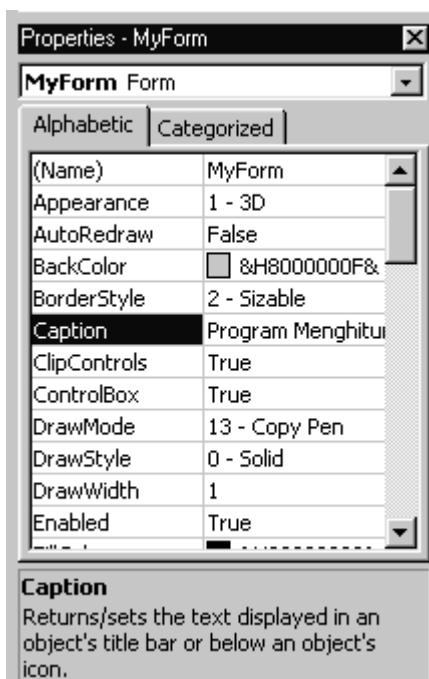


Caranya :

Klik pada form1, kemudian pada jendela properti cari baris name, kemudian ketik nilainya MyForm, kemudian cari baris caption, kemudian ketik nilainya menjadi Program Menghitung Luas Persegi.

Klik pada Label1, kemudian pada jendela properti cari baris caption, kemudian ketik nilainya &Panjang. Lakukan juga untuk kontrol lainnya berdasarkan tabel berikut.

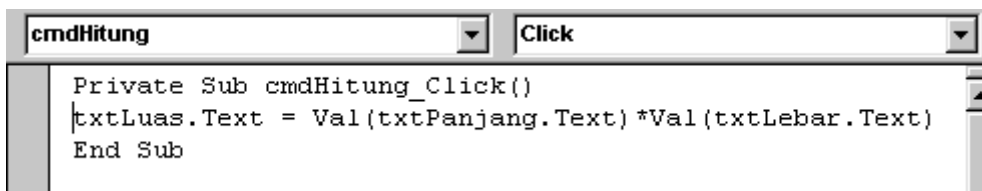
Catatan : Untuk menampilkan jendela properties, anda dapat menggunakan **F4**



Objek/Kontrol	Properti	Nilai
Form1	Name	Myform
	Caption	Program Menghitung Luas Persegi
Label1	Caption	&Panjang
Label2	Caption	&Lebar
Label3	Caption	Lu&as
Text1	Name	txtPanjang
	Text	""
Text2	Name	txtLebar
	Text	""
Text3	Name	txtLuas
	Text	""
Command1	Name	cmdHitung
	Caption	&Hitung
Command2	Name	cmdSelesai
	Caption	&Selesai

Catatan : tanda ampersand & pada Visual Basic digunakan sebagai penanda kunci akses penekanan Alt oleh pemakai.

4. Lakukan koding untuk menghitung Luas, ketika pemakai melakukan klik atau Alt H, pada cmdHitung



Caranya :

Lakukan double klik pada kontrol cmdHitung, sehingga akan muncul jendela koding dan procedure event cmdHitung_Click :

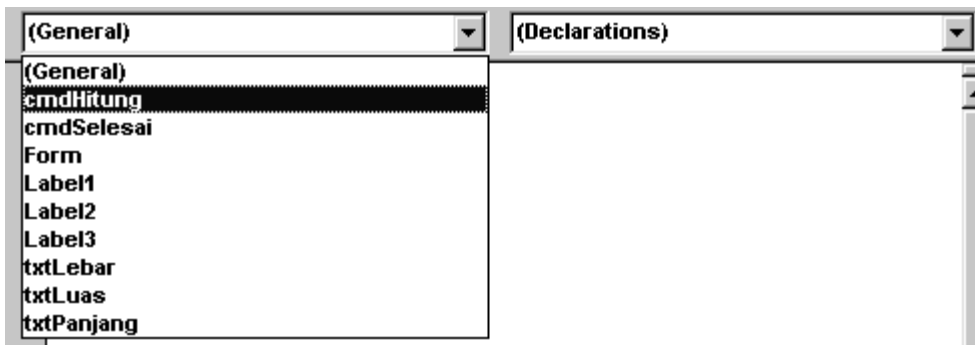
```
Private Sub cmdHitung_Click()
```

```
End Sub
```

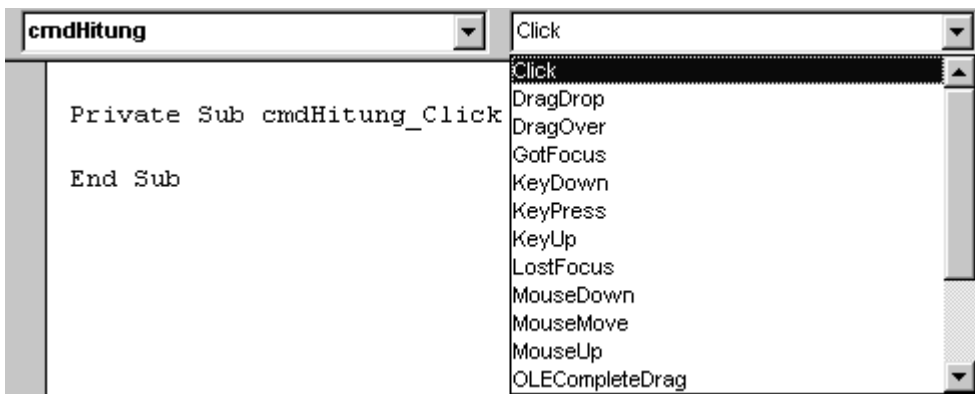
procedure event, adalah subrutin yang akan dibangkitkan oleh aplikasi, ketika pemakai melakukan klik pada kontrol cmdHitung. Perintah-perintah yang anda ketik diantara Sub dan End Sub akan dijalankan.

Cara lain :

Tekan F7 untuk menampilkan jendela koding, kemudian pilih nama kontrol dari daftar kontrol seperti gambar berikut :



kemudian pilih daftar kejadian (event) seperti yang ditunjukkan pada gambar berikut :



Catatan : sebaiknya anda senantiasa menggunakan salah satu cara diatas untuk melakukan koding untuk menjamin kebenaran penulisan pada masing-masing event procedure.

Lakukan juga koding untuk cmdSelesai :

```
Private Sub cmdSelesai_Click()
```


```
Unload Me
```

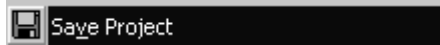
```
End Sub
```

5. Tampilkan kembali MyForm dengan klik **View Object** pada jendela Project Explorer

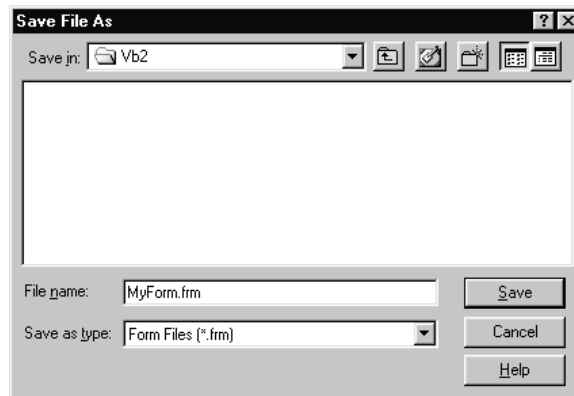
6. Menyimpan project :

Buatlah bernama C:\VB2

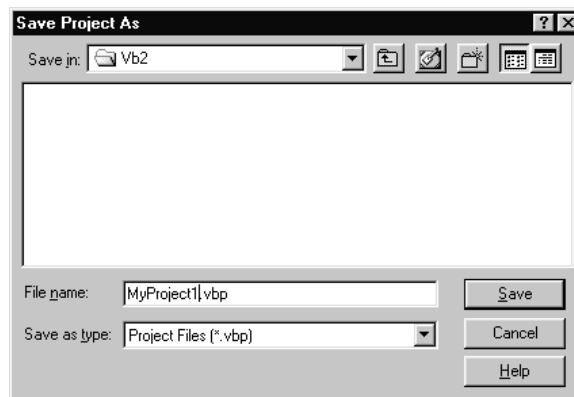
Klik pada icon save  atau gunakan menu File, dan pilih




Pilih Save in pada folder Vb2, dan nama form MyForm.frm, dan klik Save

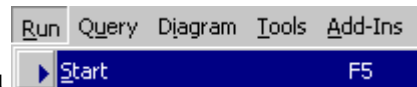


Simpan project sebagai MyProject1.vbp



Pada proses penyimpanan diatas akan menghasilkan file berikut : MyForm.frm, MyProject1.vbp, dan MyProject1.vbw. Coba periksa ini folder C:\VB2

7. Jalankan program anda dengan klik pada , atau



9. Tutup aplikasi Visual Basic anda.

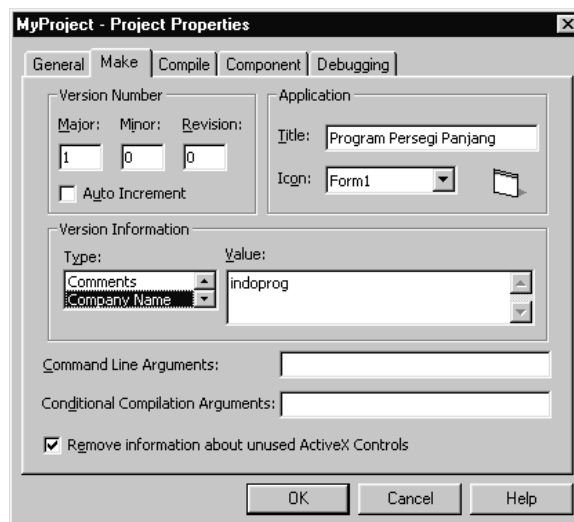
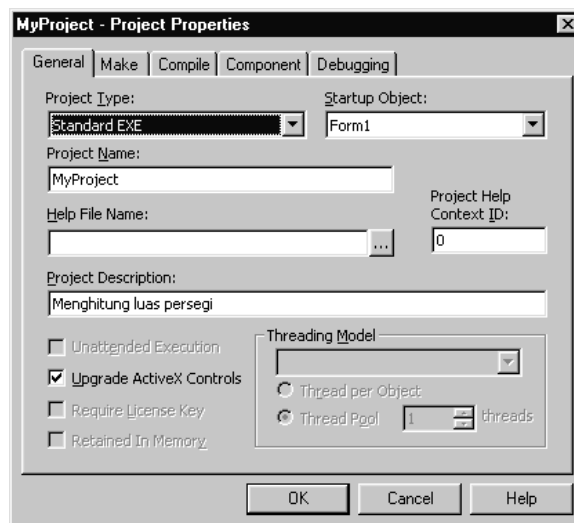
10. Atur tab index masing-masing kontrol, dimana Label1 (0), txtPanjang (1), Label2 (2), txtLebar (3), Label3 (4), txtLuas(5), cmdHitung (6), cmdSelesai (7).

Caranya :

Klik pada cmdSelesai, atau properti tabindexnya menjadi 0, kemudian klik pada cmdHitung, atur properti indexnya menjadi 0, kemudian txtLuas, Label3, txtLebar, Label2, txtPanjang, Label1, masing-masing menjadi 0, sekarang semua kontrol telah berurut sesuai dengan yang kita inginkan. (Ini adalah salah satu trik untuk menomori tabindex dengan mudah)

11. Jalankan kembali program anda, sekarang coba tekan Alt+P, Alt+L, sekarang anda dapat mengaktifkan textbox Panjang dengan Alt+H (akses key dari Label1), dst.

12. Atur properti project anda, dengan menu Project, kemudian Project Properties, akan muncul dialog sebagai berikut

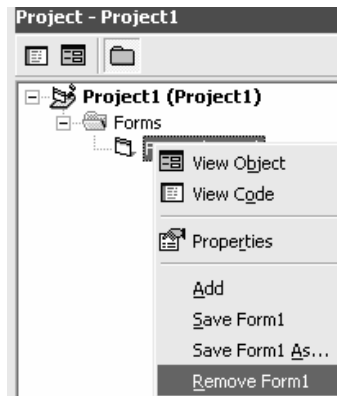


13. Simpan project anda, dan keluar dari Visual Basic

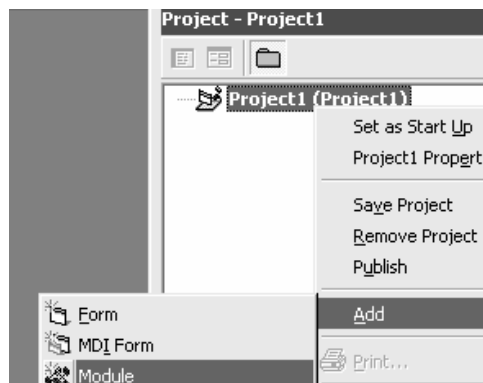
14. Aktifkan kembali Visual Basic anda, dan klik pada tab Recent, dan pilih kembali MyProject1.

Bagaimana kalau program Visual Basic tanpa form ?

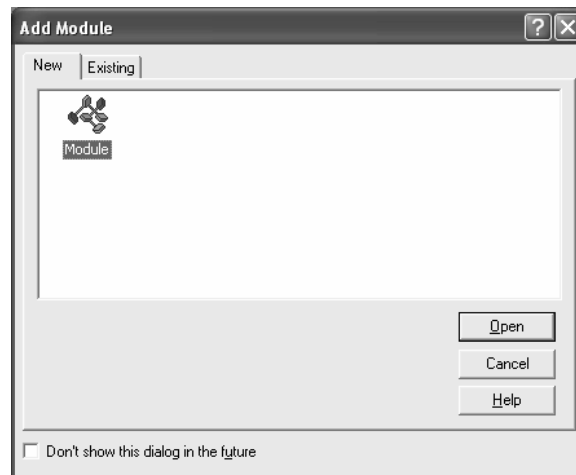
1. Buatlah sebuah project baru dengan pilihan Standard Exe
2. Pada jendela Project Explorer, klik kanan pada form1, dan pilih Remove Form1



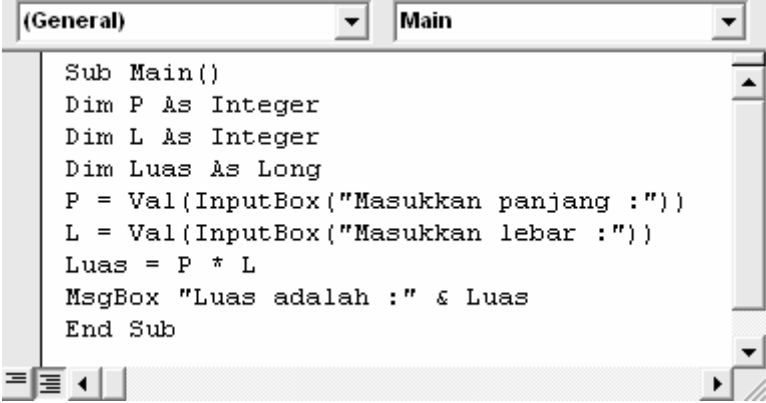
3. Pada Project1, klik kanan dan pilih Add, Module



Pilih Module dan klik pada Open



Selanjutnya akan muncul jendela koding untuk Module1, dan ketikkan program berikut



```
Sub Main()  
Dim P As Integer  
Dim L As Integer  
Dim Luas As Long  
P = Val(InputBox("Masukkan panjang :"))  
L = Val(InputBox("Masukkan lebar :"))  
Luas = P * L  
MsgBox "Luas adalah :" & Luas  
End Sub
```

Tekan F5 untuk menjalankannya.

Simpanlah program diatas dengan nama MyProject2, dan analisa file apa saja yang terbentuk berkaitan dengan project diatas.

Catatan penulis:

Saya yakin anda lebih familiar dengan alur program kedua dimana program berjalan mulai dari masukkan panjang, kemudian lebar, menghitung luas, serta mencetaknya. Hal tersebut sesuai dengan program konvensional, sedangkan pada program yang menggunakan form, eksekusi perhitungan baru dilakukan setelah pemakai melakukan klik pada tombol Hitung.

Disinilah letak tantangan bagi anda yang berasal dari latar belakang pemrograman seperti Basic, Pascal, C/C++ maupun Clipper yang mulai belajar Visual Basic, dimana eksekusi program ditentukan oleh Event yang terjadi sehingga menuntut ketelitian untuk mengantisipasi segala aksi yang mungkin dilakukan pemakai, yang kalau tidak dapat menyebabkan bug pada program.

Kesulitan yang sama juga terjadi pada sisi pengajar, dimana dalam menerangkan alur program menjadi agak susah.

Latihan 3

Judul : Memahami operator pada VBA

Petunjuk, sebaiknya lembar latihan ini anda cetak ke printer.

1. Aktifkan jendela Immediety dengan menggunakan menu View, Immediety Window atau menekan Ctrl + G



dan cobalah beberapa ekspresi berikut :

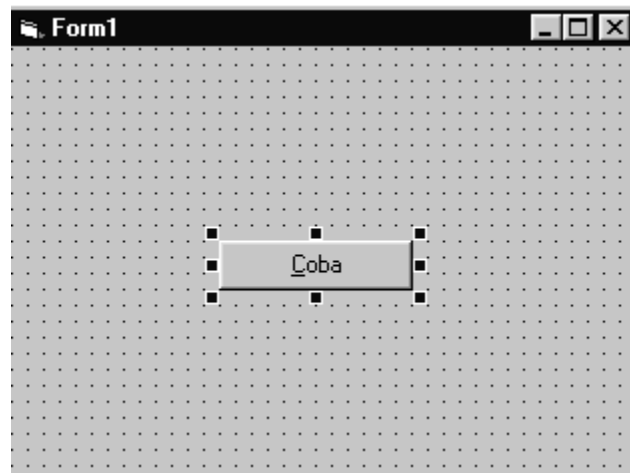
- ? 1 + 2
- ? "1" + "2"
- ? 1 + "2"
- ? "1" + 2
- ? 1 & 2
- ? "1" & "2"
- ? "1" & 2

Catatan : Operator + dapat digunakan untuk penjumlahan maupun pengabungan string, tetapi untuk pengabungan string lebih disarankan untuk menggunakan Operator &.

- ? #18 May 2001# + 31
- ? #19 June 2001# - 1
- ? #19 June 2001# - #17 June 2001#
- ? #05/18/2001# + 31
- ? #06/19/2001# - 1
- ? #06/19/2001# - #06/17/2001#
- ? #01:01:01# + 0.25 'Mengapa ?
- ? #06:01:01# - 0.25
- ? #06:01:01# - #05:30:30#

Catatan : Pada Visual Basic data date/time ditulis dengan #tanggal#, Operator + digunakan untuk menambah sejumlah hari, Operator - digunakan untuk mengurangi sejumlah hari atau selisih hari.

- ? True + 0



Lakukan koding pada event cmdCoba_Click, sebagai berikut :

```
Private Sub MyCoba_Click()  
Dim a As Integer  
Dim b As Integer  
Dim c As String  
Dim d As String  
a = 1.2 + 2.2 'Berapa nilai a  
b = 1.3 + 2.3 'Berapa nilai b  
c = "100" 'Berapa nilai c  
d = c + 200 'Berapa nilai d  
Print a  
Print b  
Print c  
Print d  
End Sub
```

Jalankan program, dan klik pada tombol Coba, dan perhatikan hasilnya, apakah sesuai dengan perkiraan anda?

Modul 3, Memahami Objek pada Visual Basic

Apa yang dimaksud dengan Objek

Objek merupakan suatu kombinasi dari kode dan data yang dapat diperlakukan sebagai satu kesatuan. Suatu Objek dapat merupakan bagian dari aplikasi seperti suatu kontrol atau suatu form. Secara keseluruhan dari aplikasi dapat juga berupa suatu objek.

Catatan Penulis :

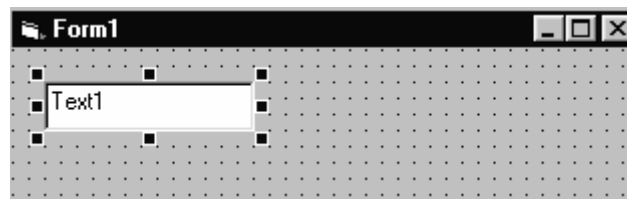
Sebagai pemula anda jangan terlalu dibingungkan/berkonsentrasi tentang istilah objek, yang penting adalah ada memahami bagaimana menggunakan objek-objek yang tersedia untuk membuat suatu program yang dapat bermanfaat.

Sesuatu yang perlu ditekankan disini adalah belajar bahasa pemrograman Visual Basic tidak sama dengan belajar pemrograman berorientasi objek, lagi pula sesuatu hal yang harus disadari adalah Visual Basic tidak mendukung konsep OOP secara murni seperti C++ maupun Java.

Dari mana datangnya Objek

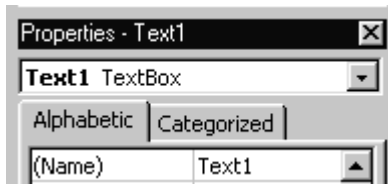
Setiap objek pada Visual Basic didefinisikan oleh suatu class. Untuk memahami hubungan antara objek dengan class-nya, perhatikan contoh berikut :

- Kontrol-kontrol yang berada pada jendela Toolbox pada Visual Basic merupakan class-class. Objek tersebut dikenal sebagai kontrol tidak akan ada sampai anda menanamkannya pada suatu form. Ketika anda menanamkan suatu kontrol, anda telah membuat suatu duplikasi atau instance dari class kontrol tersebut.
- Form dimana anda bekerja pada saat desain adalah suatu class. Pada saat run time, Visual Basic menciptakan suatu instance dari class form.



Gambar 2-1, kontrol Text1 yang ditanam diatas form berasal dari class Textbox

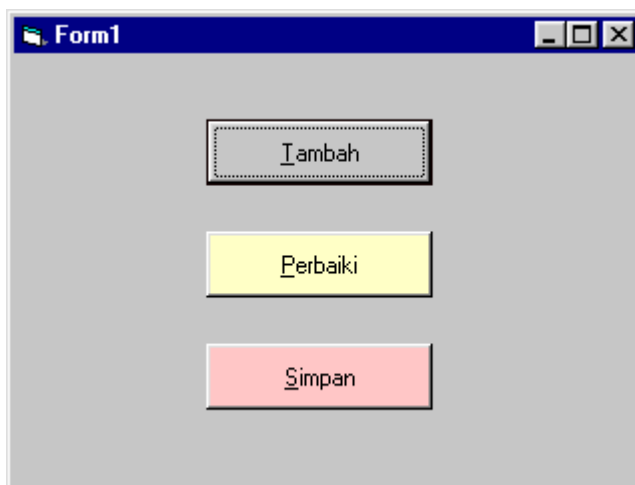
Misalnya anda menanamkan sebuah textbox diatas form1, coba perhatikan jendela properti kontrol tersebut akan menampilkan Text1 (nama dari kontrol textbox tersebut) dan Textbox (nama dari class kontrol tersebut).



Gambar 2-2, jendela property menunjukkan nama kontrol dan jenis Class-nya

Semua objek merupakan duplikasi dari class mereka. Sesaat setelah menjadi objek berdiri sendiri, masing-masing dapat memiliki properti yang berbeda-beda.

Misalnya anda menanam tiga buah command button pada suatu form, masing-masing command button merupakan instance dari class CommandButton sehingga memiliki sejumlah karakteristik dan kemampuan (properti-properti, metode-metode, dan event-event) yang sama, dimana yang terdefinisi pada class-nya. Tetapi, masing-masing dapat memiliki Name, Caption, BackColor, Style yang berbeda sesuai dengan pengaturan pada masing-masing properti.



Gambar 2-3, masing-masing kontrol dapat memiliki properti masing-masing

Catatan penulis :

Jadi tombol Tambah, Perbaiki dan Simpan berasal dari Class yang sama yaitu Command Button, tetapi setelah mereka ditanam diatas form, masing-masing tombol merupakan objek yang berdiri sendiri dan memiliki properti yang berbeda.

Dalam istilah OOP tombol Tambah, Perbaiki dan Simpan adalah objek yang merupakan instance dari Class Command Button.

Bekerja dengan objek

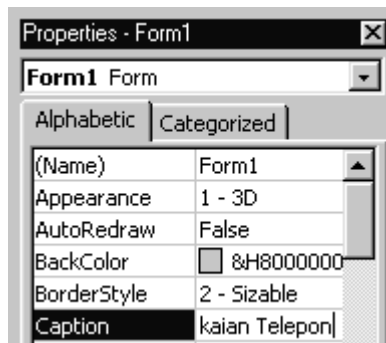
Objek-objek Visual Basic memiliki properti-properti, metode-metode, dan event-event. Pada Visual Basic, suatu data dari objek (setting dan atribut) disebut sebagai properti, sedangkan berbagai subrutin yang dapat beroperasi pada objek tersebut disebut sebagai metodenya. Dan suatu event adalah aksi yang dikenali oleh suatu objek, seperti klik pada mouse atau penekanan tombol, sehingga anda dapat menulis kode untuk menangani event tersebut.

Mengatur Nilai properti

Anda dapat langsung mengatur nilai properti suatu kontrol pada jendela properti, maupun pada saat runtime dengan menggunakan bentuk penulisan berikut :

Objek.properti = ekspresi

Contoh :



Gambar 2-4, pengaturan properti dapat dilakukan melalui jendela properti pada waktu desain

atau

```
Private Sub Form_Load()  
MyForm.Caption = "Pemakai Telepon"  
End Sub
```

Mengambil nilai dari Properti

Anda dapat mengambil nilai dari properti suatu objek dengan penulisan sebagai berikut :

Variabel = Objek.properti

Contoh :

```
Private Sub cmdProses_Click()  
Nama = txtNama.Text  
End Sub
```


Menggunakan Metode dalam Kode

Ketika anda menggunakan suatu metode dalam kode anda, hal tersebut tergantung kepada bagaimana kaitan perintah dan berapa argumen yang diperlukan, dan apakah metode tersebut mengembalikan suatu nilai. Ketika suatu metode tidak membutuhkan argumen anda dapat menulisnya sebagai berikut ::

Objek.Metode

Contoh :

```
Private Sub cmdBuka_Click()  
frmPemakai.show  
End Sub
```

Catatan penulis :

Salah satu kesulitan programmer yang menggunakan bahasa konvensional seperti Basic, Pascal, maupun Clipper yang mempelajari Visual Basic adalah dalam memahami topik tentang objek dan event programming, karena merupakan hal yang tidak ada pada bahasa tersebut.

Jika anda memikirkan kembali istilah properti, metode dan event, maka properti merupakan ciri-ciri yang membedakan, misalnya si Budi dan si Amat yang sama-sama merupakan instance dari Class manusia, tetapi properti mereka dapat berbeda

Properti	si Budi	Si Amat
Tinggi	160 cm	170 cm
Berat	60 kg	70 kg
Rambut	Lurus	Kriting
Warna kulit	Kuning langsung	Sawo mata

Sedangkan metode merupakan aksi yang dapat dilakukan oleh si Budi dan si Amat seperti berjalan, berlari, berbicara, ketawa, dll

Sedangkan event merupakan reaksi terhadap suatu kejadian, contoh kalau dipukul maka apa yang harus dilakukan ?

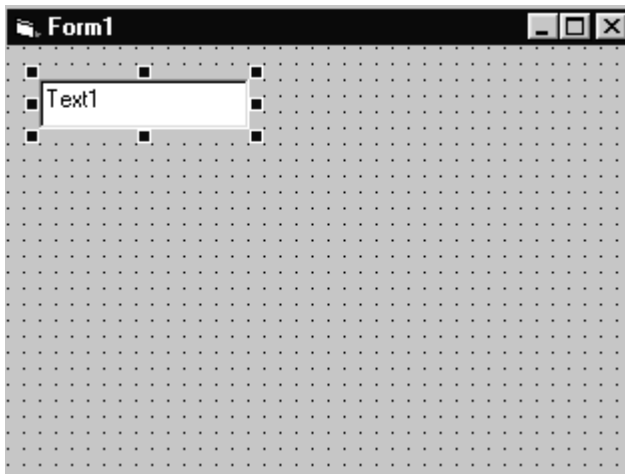
Program pada Visual Basic akan berjalan berdasarkan event-event yang terjadi, baik yang event dibangkitkan oleh sistem operasi maupun event yang dibangkitkan karena aksi oleh pemakai.

Beberapa properti yang umum pada kontrol

Berikut ini akan dibahas beberapa properti yang umum yang hampir terdapat pada semua intrinsic kontrol pada Visual Basic.

Properti *Left*, *Top*, *Width* dan *Height*

Semua objek yang visible memiliki properti yang menentukan letak dan ukurannya. Nilai properti *Left* dan *Top* sifatnya relatif terhadap kontainernya (misalnya suatu textbox yang ditanamkan diatas form, maka form tersebut disebut sebagai kontainer bagi textbox tersebut), sedangkan *Width* dan *Height* menentukan ukuran dari objek tersebut.



Gambar 2-5, *Form1* merupakan kontainer bagi *Text1*

Nb. Pada defaultnya nilai properti ini dinyatakan dalam twips (1 inch = 1440 twips; 1 cm = 576 twips)

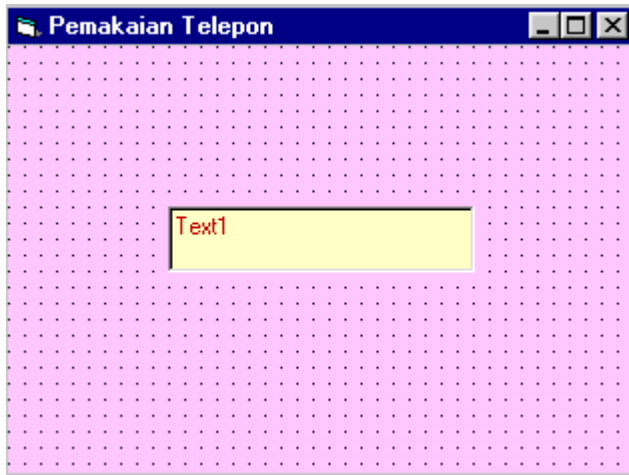
Perhatian : Properti *Width* dan *Height* pada control Combobox sifatnya readonly.

Catatan penulis :

Tidak semua objek pada Visual Basic dapat menjadi kontainer bagi objek lainnya, kontrol standar yang dapat berlaku sebagai kontainer adalah form, frame, picturebox.

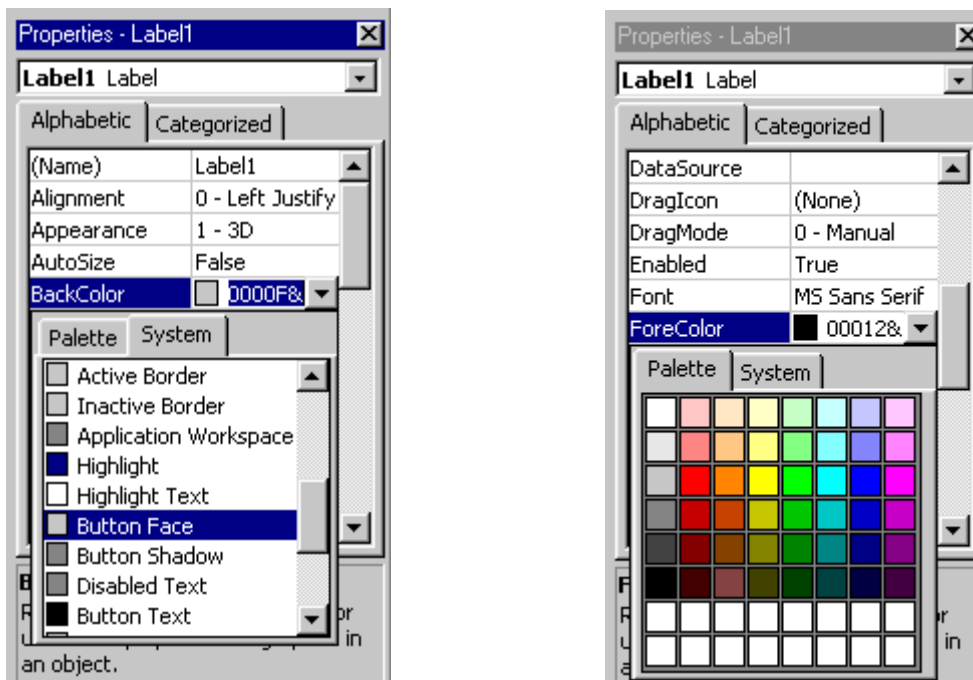
Properti *ForeColor* dan *BackColor*

Umumnya objek yang visible memiliki properti *ForeColor* dan *BackColor*, yang mempengaruhi warna tulisan dan warna latarbelakang, tetapi beberapa diantaranya tidak ada (misalnya control scroll-bars), pada command button tidak terdapat properti *ForeColor*, dan *BackColor*nya hanya berfungsi jika anda mengganti properti *style* menjadi 1 - Graphical.



Gambar 2-6, form dengan pemakaian bgcolor dan forecolor

Nb. Setting properti Backcolor menjadi tidak berfungsi jika anda menset properti BackStyle menjadi 0 - transparent.



Gambar 2-7, pengaturan Backcolor dan ForeColor melalui jendela properti

Ketika anda menentukan nilai untuk properti ini, anda dapat menggunakan suatu warna standard, atau warna custom (saya menyarankan anda untuk menggunakan warna standard untuk menjaga nilai rasa pemakai integrasi dengan sistem operasi)

Properti *Font*

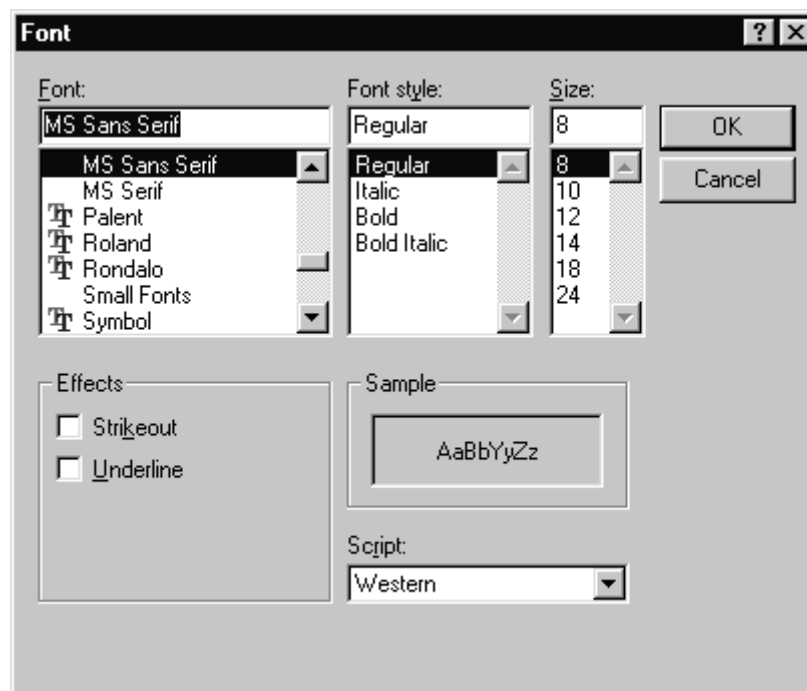
Pada waktu desain, anda dapat menentukan properti Font dengan dialog box, tetapi pada saat runtime, hal tersebut harus dilakukan dengan properti *Name*, *Size*, *Bold*, *Italic*, *Underline*, and *Strikethrough*, karena sebenarnya Font adalah suatu objek campuran.

Contoh :

```
Text1.Font.Name = "Tahoma"  
Text1.Font.Size = 12  
Text1.Font.Bold = True  
Text1.Font.Underline = True
```

Properti *Caption* dan *Text*

Caption adalah tulisan yang tampil dalam suatu control (atau title pada objek form) yang mana pemakai tidak dapat melakukan modifikasi langsung (control Label, CommandButton, Checkbox, Optionbutton, Data, dan Frame), sebaliknya Text adalah tulisan yang dapat dimodifikasi oleh pemakai (control Textbox, Listbox dan Combobox).



Gambar 2-8, pengaturan Font melalui dialog box Font dari jendela properti

Nb. Pada properti Caption, kita dapat menggunakan tanda & (ampersand) untuk menyatakan hotkey bagi control tersebut. Properti Text dapat diabaikan, karena merupakan properti default bagi suatu control.

Contoh :

```
' Kedua perintah ini adalah sama
Text2.Text = Text1.Text
Text2 = Text1                               'tanpa properti text
```

Properti *Enabled* dan *Visible*

Pada defaultnya semua control dan form adalah Enabled dan Visible. Anda dapat menyembunyikan suatu kontrol dengan menggunakan properti Visible = False. Kontrol yang properti Enabled = False tidak dapat diakses oleh pemakai, tetapi dapat diakses secara kode.

Programmer dapat juga menggunakan properti Locked = True untuk membuat suatu kontrol read only.

Properti *TabStop* dan *TabIndex*

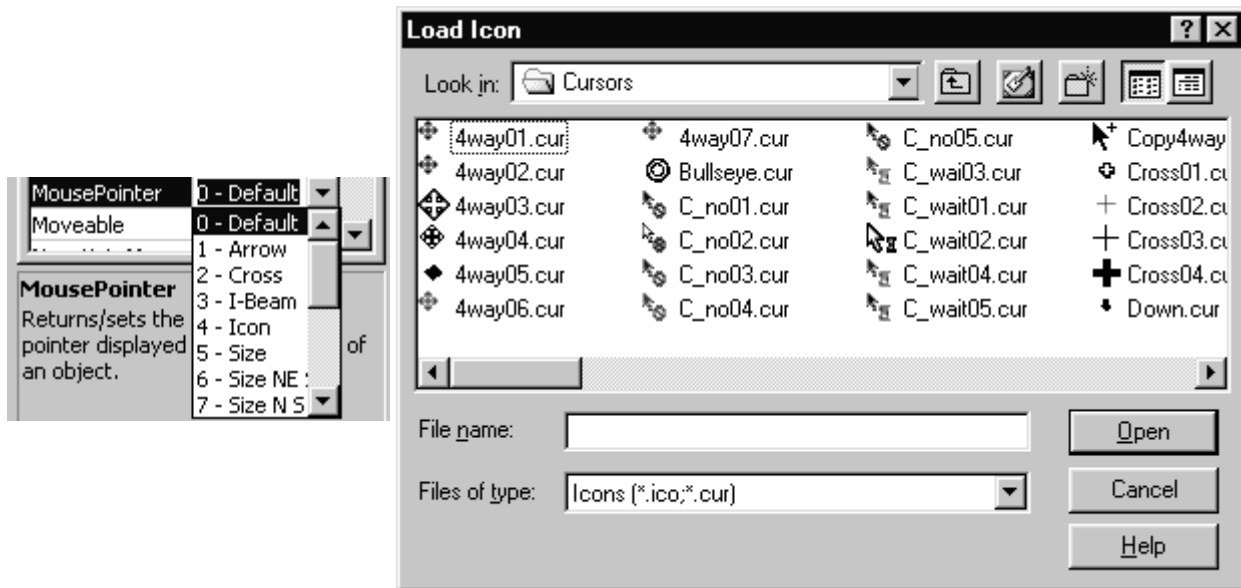
Suatu kontrol yang dapat menerima fokus input memiliki properti TabStop. Kontrol yang memiliki properti TabStop juga memiliki properti TabIndex. Properti TabStop menentukan apakah suatu kontrol dapat menerima fokus pada penekanan tombol Tab oleh pemakai, defaultnya adalah true, sedangkan properti TabIndex menentukan urutan fokus pada saat pemakai menekan Tab.

Properti *MousePointer* dan *MouseIcon*

Properti ini akan menentukan bentuk kursor mouse ketika berada diatas kontrol tersebut. Windows memperbolehkan kita mengatur tampilan mouse untuk setiap form dan kontrol dengan mengikuti aturan berikut :

- Jika properti *Screen.MousePointer* di set ke nilai yang bukan 0-vbDefault, kursor mouse akan mengikuti nilai ini, tetapi ketika mouse berada diatas aplikasi lain (atau pada desktop), tampilan kursor akan bergantung kepada kondisi aplikasi bersangkutan, bukan pada aplikasi anda.
- Jika *Screen.MousePointer* adalah 0 dan kursor mouse berada diatas suatu kontrol, Visual Basic akan memeriksa nilai properti *MousePointer* kontrol tersebut; jika nilainya bukan 0-vbDefault, kursor mouse akan di set ke nilai tersebut.
- Jika *Screen.MousePointer* adalah 0 dan mouse berada diatas permukaan suatu form atau berada diatas kontrol yang properti *MousePointer* adalah 0, Visual Basic akan menggunakan nilai yang tersimpan pada properti *Mousepointer* pada form.

Properti *MouseIcon* dapat digunakan untuk menampilkan bentuk pointer mouse sesuai dengan keinginan pemakai, tetapi sebelumnya properti *MouseIcon* harus di set ke 99-vbCustom, dan kemudian set suatu icon pada properti *MouseIcon*.



Gambar 2-9, pengaturan Mouse Pointer dan Mouse Pointer melalui jendela properti

Properti Tag

Semua kontrol mendukung properti Tag, tanpa kecuali, karena properti ini disediakan oleh Visual Basic, bukan oleh kontrol. Properti Tag digunakan sebagai kontainer untuk bagi data dari kontrol yang anda ingin simpan (misalnya anda dapat menggunakannya untuk menyimpan nilai awal dari textbox, sehingga pemakai dapat melakukan Undo dengan mengembalikan nilai awal tersebut dari properti Tag).

Beberapa Metode-metode umum

Berikut ini akan dibahas beberapa metode yang umum yang hampir terdapat pada semua intrinsic kontrol pada Visual Basic.

Metode Move

Jika suatu kontrol mendukung properti *Left*, *Top*, *Width*, dan *Height*, juga mendukung metode *Move*, dengannya dimana anda dapat mengubah beberapa atau semua properti dalam satu operasi tunggal. Contoh berikut mengubah tiga properti: *Left*, *Top*, dan *Width*.

Contoh :

'Mendoublekan lebar form, dan memindahkan ke sudut kiri atas layar
'Syntaxnya adalah: Move Left, Top, Width, Height.

```
Form1.Move 0, 0, Form1.Width * 2
```

Metode Refresh

Metode refresh menyebabkan suatu form akan digambar ulang. Pada dasarnya Visual Basic otomatis memanggil metode ini setiap ada kesempatan, tetapi anda dapat menggunakannya untuk mengupdate tampilan seketika.

Contoh :

```
For n = 1000 To 1 Step -1
    Labell.Caption = CStr(i)
    Labell.Refresh           ' Mengupdate label seketika.
Next
```

Metode SetFocus

Metode ini memindahkan fokus input ke kontrol tertentu. Sesuatu masalah yang sering terjadi adalah metode ini akan menyebabkan error ketika diterapkan pada kontrol yang sedang di Disable atau dalam keadaan Invisible. Untuk menghindari hal ini, metode SetFocus jangan digunakan pada bagian Form Load.

Contoh :

```
Private Sub cmdAdd_Click()
    Aksi = flAdd
    Call Buka
    Call Kosong
    txtCCode.SetFocus           'Memindahkan fokus ke kontrol txtCCode
End Sub
```

Metode ZOrder

Metode *ZOrder* memberikan efek tampilan kontrol yang saling menimpa. Anda menggunakan metode ini untuk memindahkan kontrol keatas kontrol yang lain. Gunakan argumen satu untuk sebaliknya.

Contoh :

```
Labell.ZOrder           'Memindahkan ke atas
Labell.ZOrder 1         'Memindahkan ke bawah
```

Pada saat design, anda dapat menggunakan Ctrl+J untuk memindahkan kontrol ke depan, dan Ctrl+K untuk memindahkan form ke belakang.



Gambar 2-10, contoh pemakaian metode ZOrder

Beberapa Event yang umum

Berikut ini akan dibahas beberapa metode yang umum yang hampir terdapat pada semua intrinsic kontrol pada Visual Basic.

Event *Click* dan *DbClick*

Event click terjadi ketika pemakai melakukan klik pada tombol kiri mouse. demikian juga Event *DbClick* terjadi karena pemakai melakukan klik dua kali.

```
Private Sub cmdUpdate_Click()
    JlhRec = JlhRec + 1
    DatPemakai.CCCode = txtCCCode.Text
    DatPemakai>Nama = txtNama.Text
    DatPemakai.Ekstension = txtEkstension.Text
    DatPemakai.Local = optLocal.Value
    DatPemakai.SLJJ = optSLJJ.Value
    DatPemakai.SLI = optSLI.Value
    Put #1, JlhRec, DatPemakai
    Posisi = JlhRec
End Sub
```

Pada kontrol-kontrol seperti *Checkbox* dan *OptionButton*, event *Click* juga dibangkitkan kalau terjadi perubahan nilai properti *Value* secara koding.

ListBox dan *Combobox* juga melakukan hal yang sama kalau properti *ListIndex*-nya berubah.

Adalah penting untuk menghindari pemakaian Event *Click* dan *DbClick* pada satu kontrol yang sama, karena tidak menjamin Event *DbClick* dibangkitkan setelah Event *Click* terjadi.

Event Change

Event Change dibangkitkan ketika isi dari suatu kontrol berubah. Tetapi pada CheckBox, dan OptionButton Event Click yang dibangkitkan.

Pada kontrol TextBox dan ComboBox dibangkitkan ketika pemakai mengetik sesuatu. (Tetapi pada kontrol ComboBox, kontrol membangkitkan event Click kalau pemakai memilih item dari list dari pada mengetiknya). Pada kontrol Scroll bar event Change terjadi ketika pemakai melakukan klik baik pada panah maupun scroll box. Event Change juga terdapat pada PictureBox, DriveListBox, dan kontrol DirListBox.

Event Change juga dibangkitkan oleh isi kontrol yang diubah melalui koding.

Event GotFocus dan LostFocus

GotFocus dibangkitkan ketika kontrol menerima fokus, dan *LostFocus* dibangkitkan ketika fokus meninggalkannya dan beralih ke kontrol lain. Pada awalnya event ini banyak digunakan untuk mekanisme validasi ketika kontrol kehilangan fokus. Pada Visual Basic 6 telah diperkenalkan suatu event *Validate* event, yang lebih sesuai untuk masalah yang sama.

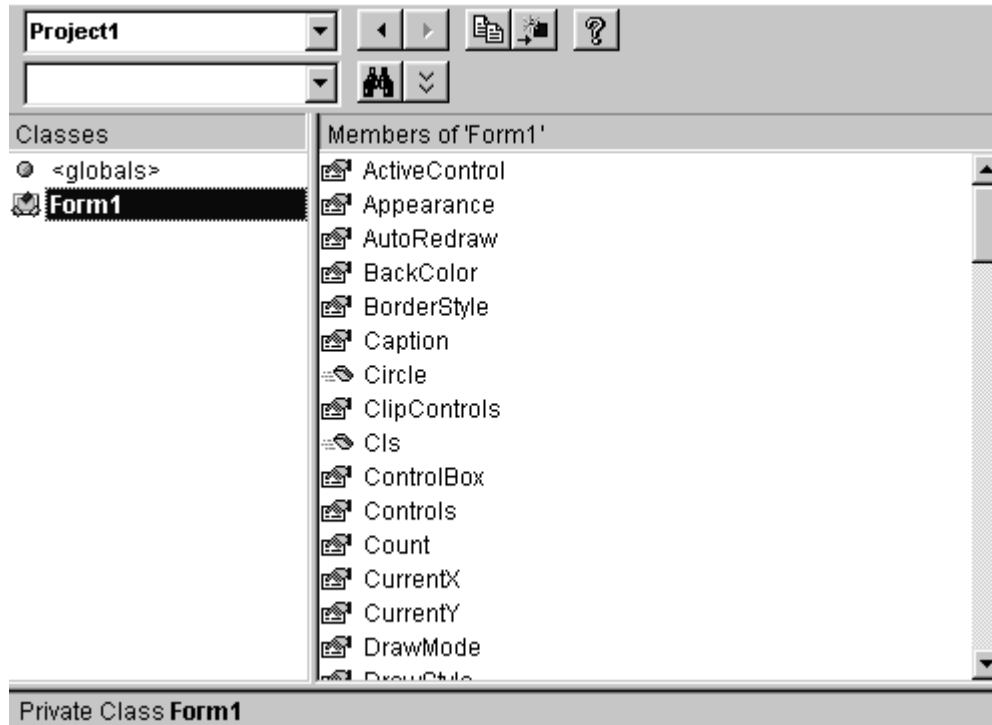
Event KeyPress, KeyDown, dan KeyUp

Event-event ini dibangkitkan ketika pemakai menekan tombol pada keyboard pada kontrol yang memiliki fokus. Urutan eventnya adalah : *KeyDown* (ketika pemakai menekan tombol), *KeyPress* (Visual Basic menterjemahkan tombol ke numerik ANSI code), dan *KeyUp* (ketika pemakai melepas tombol). Hanya tombol yang berkaitan dengan tombol kontrol (Ctrl+x, BackSpace, Enter, dan Escape) serta karakter yang dapat dicetak membangkitkan event *KeyPress* event. Tombol lainnya--termasuk tombol panah, tombol fungsi, kombinasi Alt+x, tidak menyebabkan event ini, hanya membangkitkan event *KeyDown* dan *KeyUp*.

Event *KeyPress* melewati ANSI code dari tombol yang ditekan. Anda dapat memanipulasinya untuk keperluan tertentu.

Objek Browser

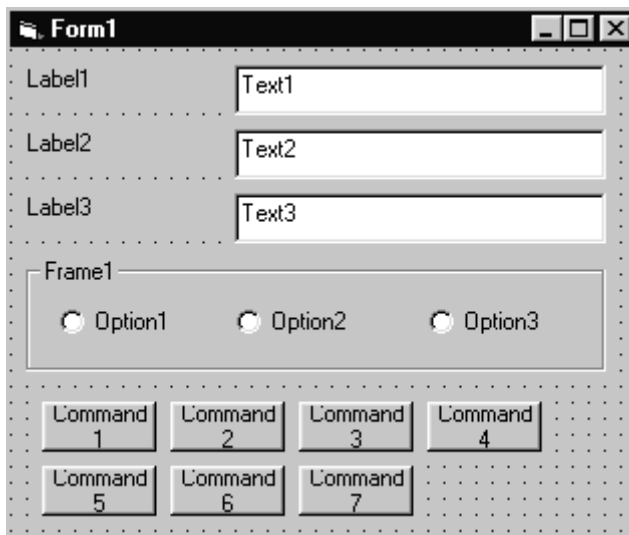
Anda dapat melihat properti-properti, metode-metode, dan event-event pada suatu objek dengan menggunakan fasilitas Objek Browser yang dapat diaktifkan dengan menggunakan menu View, Object Browser.



Gambar 2-11, jendela objek browser untuk melihat isi suatu objek

Modul 4, Mengenal Form

Form merupakan objek pertama yang anda hadapi begitu membuat suatu project dan merupakan tempat kita membentuk user interface. Pada form kita dapat menanamkan berbagai Kontrol seperti label, textbox, combobox, listbox, optionbutton, dll untuk membuat user interface.



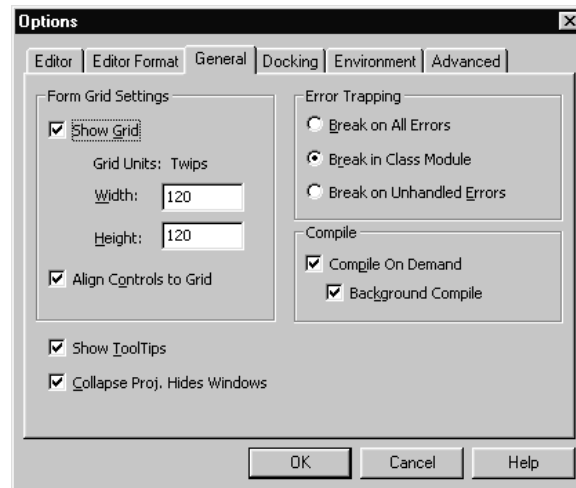
Gambar 2-12, user interface form dengan berbagai kontrol di atasnya

Catatan penulis :

Secara default Visual Basic akan memberikan nomor Tab Index pada setiap kontrol berdasarkan urutan penempatannya pada saat design. Urutan tab index ini akan menentukan urutan fokus pada saat keaktifan Form pertama kali, dan fokus berikutnya pada setiap penekanan tombol Tab.

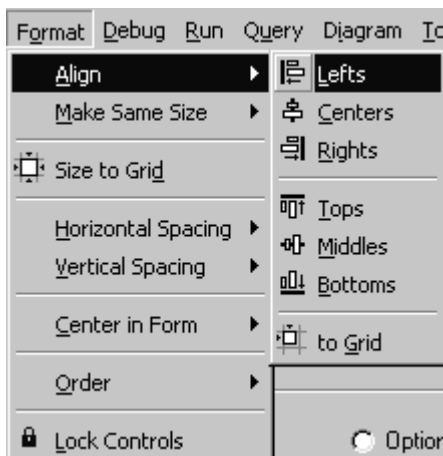
Anda dapat mengubah urutan Tab Index pada masing-masing kontrol dengan mengatur properti TabIndex.

Untuk memudah anda dalam menempatkan kontrol-kontrol pada form, periksalah option seperti Show Grid dan ukurannya, serta Align Control to Grid, pengaturan ini dapat dilakukan dengan menu Tools, Option, kemudian memilih tab General.



Gambar 2-13, jendela option untuk pengaturan form

Untuk memudahkan anda mengatur ukuran, jarak antar kontrol maupun pemerataan kontrol-kontrol yang anda tempatkan diatas form, anda dapat menggunakan kelompok menu format



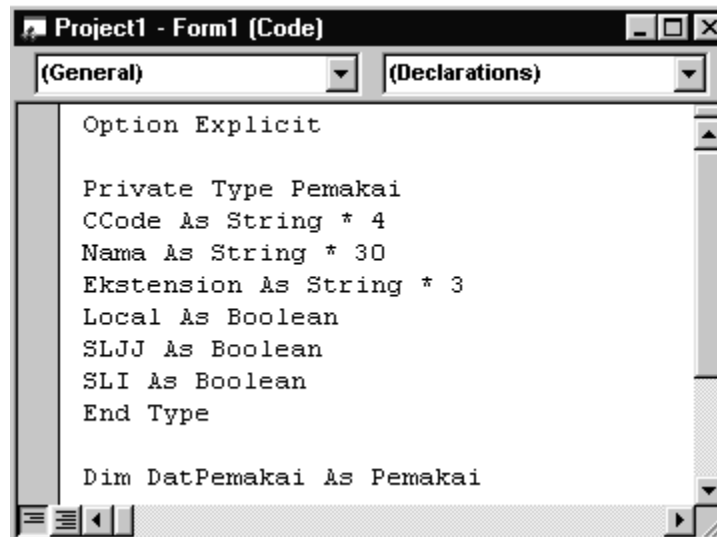
Gambar 2-14, menu format

Struktur kode pada Form

Memahami struktur kode pada jendela kode di suatu form akan sangat membantu programmer untuk menulis kode yang terstruktur. Adapun struktur kode pada form dapat dibagi atas tiga bagian yaitu :

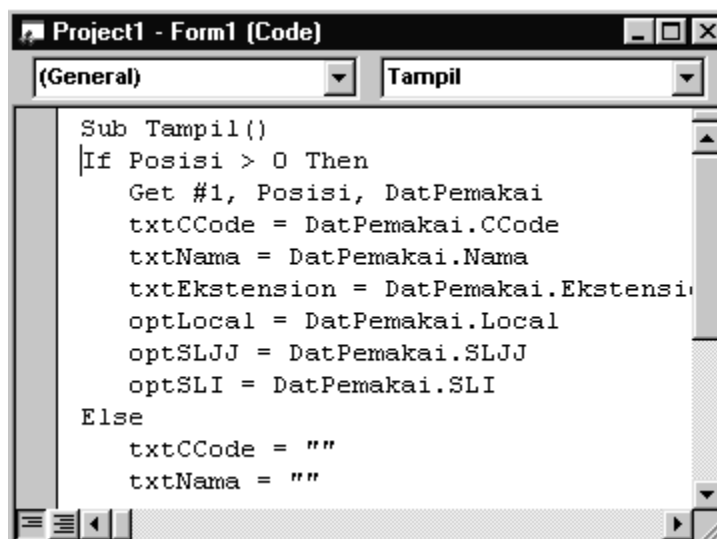
- Bagian General Declaration, bagian ini digunakan untuk deklarasi Option Explicit, Option Base, Type, dan variabel yang dapat digunakan pada form level (dikenal oleh semua subrutin pada form tersebut) baik pada tingkat Public

maupun Private. Bagian ini biasanya terletak pada bagian teratas pada jendela koding.



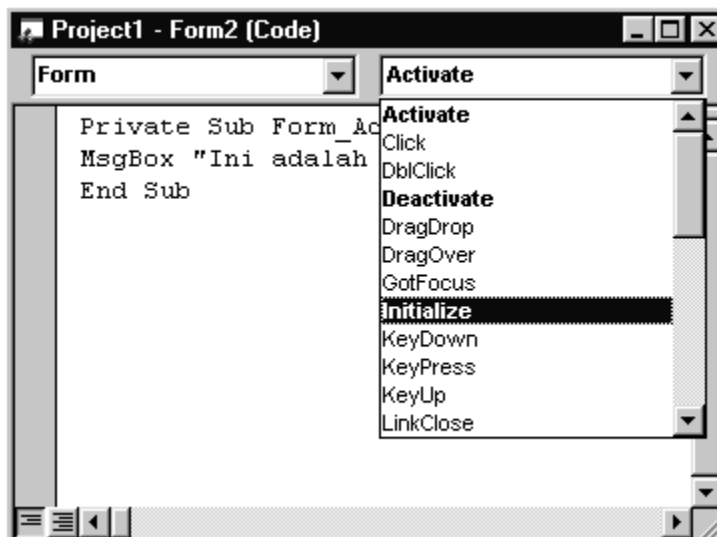
Gambar 2-15, bagian general declaration pada jendela koding form

- Bagian General Procedure, bagian ini merupakan subrutin maupun fungsi yang dibuat oleh pemakai. Bagian ini ditandai dengan (General), dan nama subrutin.



Gambar 2-16, bagian general procedure pada jendela koding form

- Bagian Event Procedure, bagian ini merupakan subrutin yang digunakan untuk menangani kejadian yang diakibatkan oleh aksi pemakai terhadap kontrol tertentu. Ditandai dengan nama kontrol dan jenis eventnya.



Gambar 2-17, bagian event procedure pada jendela koding form

Beberapa Event, Metoda, dan Perintah pada Form

Supaya anda dapat bekerja baik dengan objek form, anda perlu mengetahui dan mengerti Event-Event, Metode-Metode serta perintah yang berhubungan dengan pemakaian form.

Event Initialize

Event ini terjadi pertama kali ketika Form dibuat dari Class-nya dan hanya sekali selama keberadaan form tersebut, Event ini digunakan untuk menginisialisasi nilai awal variabel.

Event Terminate

Event ini terjadi ketika objek Form akan dimusnahkan dari memori.

Event Load

Event ini terjadi ketika suatu form di Load. Pada Event procedure ini anda letakkan statement yang digunakan untuk mengatur setting awal form anda, misalnya membuka file, menginisialisasi nilai awal variabel, dan mengatur properti kontrol-kontrol pada form.

Contoh :

```
Private Sub Form_Load()
Open "C:\Pemakai.Dat" For Random As #1 Len=Len(DatPemakai)
JlhRec = LOF(1) / Len(DatPemakai)
If JlhRec > 0 Then
    Posisi = 1
Else
```

```
    Posisi = 0
End If
Call Tampil
End Sub
```

Event QueryUnload

Event ini terjadi ketika suatu form akan tutup, dan dapat mempelajari bagaimana pemakai menutup form tersebut dengan mempelajari parameter UnloadMode.

Contoh :

```
Private Sub Form_QueryUnload(Cancel As Integer, _
    UnloadMode As Integer)
    Select Case UnloadMode
        Case vbFormControlMenu ' = 0
            ' Form diclose oleh pemakai.
        Case vbFormCode ' = 1
            ' Form diclose dengan kode.
        Case vbAppWindows ' = 2
            ' Session windows berakhir.
        Case vbAppTaskManager ' = 3
            ' Task manager mengakhiri program ini.
        Case vbFormMDIForm ' = 4
            ' Form ditutup oleh MDI.
        Case vbFormOwner ' = 5
            ' Form ditutup oleh Owner.
    End Select
End Sub
```

Parameter Cancel dapat digunakan untuk menentukan apakah proses dilanjutkan ke Event UnLoad atau tidak. (0 berarti proses dihentikan, 1 berarti proses dilanjutkan ke Event Unload)

Event UnLoad

Event ini terjadi ketika suatu form di unload dengan menggunakan command Close pada Control menu atau dengan statement UnLoad. Event ini terjadi setelah Event QueryUnload. Anda dapat mengetikkan kode-kode untuk menutup file-file yang terbuka pada Event ini.

Contoh :

```
Private Sub Form_Unload(Cancel As Integer)
    Close #1
End Sub
```

Parameter Cancel dapat digunakan untuk menentukan apakah proses UnLoad dilakukan atau tidak. (0 berarti proses UnLoad dilakukan, 1 berarti proses Unload dibatalkan)

Contoh :

```
Private Sub Form_Unload(Cancel As Integer)
Dim nPIL As Integer
nPIL = MsgBox("Apakah Anda Yakin", vbYesNo)
If nPIL = vbYes Then
    Close #1
    Cancel = 0
Else
    Cancel = 1
End If
End Sub
```

Event Resize

Event ini terjadi ketika form pertama kali ditampilkan atau ukuran dari suatu object berubah.

Contoh :

```
Private Sub Form_Resize ()
'Mengubah ukuran PictureBox menjadi sama dengan ukuran form yang
diresize.
    Picture1.Move 0,0, ScaleWidth, ScaleHeight
End Sub
```

Catatan :

ScaleWidth, properti yang mengembalikan nilai ukuran lebar interior dari suatu kontrol

ScaleHeight, properti yang mengembalikan nilai ukuran lebar interior dari suatu kontrol

Event Activate

Event Activate terjadi ketika suatu form menjadi jendela aktif, ketika user melakukan klik pada form, atau menggunakan metode SHOW atau SETFOCUS.

Event DeActivate

Event DeActivate terjadi ketika suatu form menjadi jendela yang tidak aktif, dimana ketika focus bergeser ke form lain. Event ini tidak terjadi pada saat UNLOAD.

Perintah Load

Melakukan Load Form, tetapi tidak membuatnya menjadi Visibel

Syntax :

Load form

Contoh :

```
Private Sub cmdLoad_Click()  
Load Form2  
End Sub
```

Metoda Show

Melakukan Load suatu Form, dan membuatnya menjadi Visible.

Syntax :

object.Show

Contoh :

```
Private Sub cmdShow_Click()  
Form2.Show  
End Sub
```

Metoda Hide

Menyembunyikan form, tetapi tidak melakukan unload

Syntax :

object.Hide

Contoh :

```
Private Sub cmdHide_Click()  
Form2.Hide  
End Sub
```

Perintah UnLoad

Melakukan UnLoad Form

Syntax :

UnLoad form

Contoh :

```
Private Sub cmdUnload_Click()  
Unload Form2
```

End Sub

Urutan Event pada Form

Secara garis besar Event pada Form adalah sebagai berikut :

Initialize, hanya sekali yaitu ketika form pertama kali dibuat dari Class-nya

Load, Activate

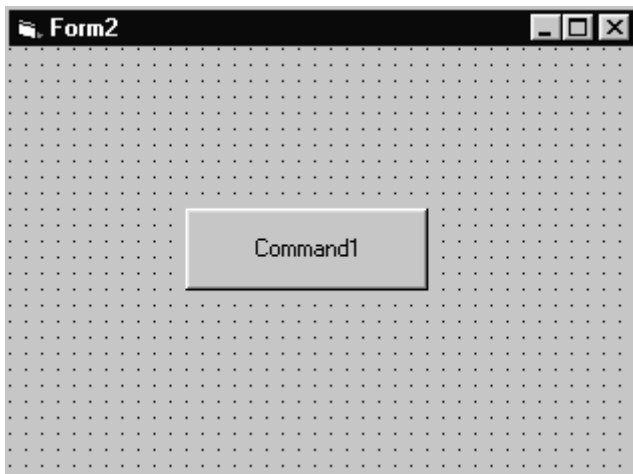
Deactivate

Terminate, hanya sekali yaitu ketika form dimusnahkan dari memori.

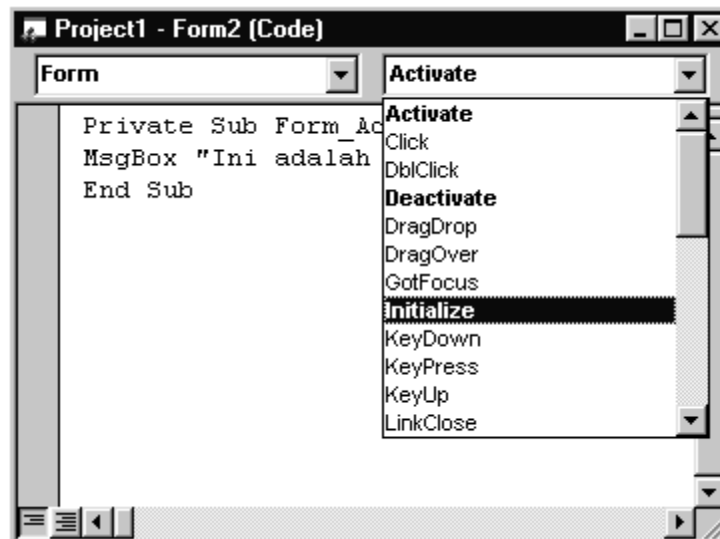
Latihan 4

Judul : Memahami urutan event, metoda, dan perintah pada Form

1. Tambahkan Form2 ke dalam project anda (caranya : Pada menu Project, pilih Add Form), Kemudian tanamkan Command1 pada Form2, sehingga menjadi sebagai berikut :



2. Aktifkan jendela koding Form2 dengan menekan F7, atau pada Menu **View**, pilih **Code**, atau klik icon **View Code** pada jendela Project Explorer. Pada DropDownList object, pilih **Form**, dan pada DropDownList procedure pilih **Initialize**.



Dan lakukan koding untuk Event Initialize untuk form2, sebagai berikut :

```
Private Sub Form_Initialize()  
MsgBox "Ini adalah initialize form2"  
End Sub
```

Dan dengan cara yang sama lakukan juga koding untuk Event-Event berikut :

Private Sub Form_Load()

MsgBox "Ini adalah Load Form2"

End Sub**Private Sub Form_Activate()**

MsgBox "Ini adalah Activate Form2"

End Sub**Private Sub Form_Deactivate()**

MsgBox "Ini adalah Deactivate Form2"

End Sub**Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)**

MsgBox "Ini adalah QueryUnload Form2"

End Sub**Private Sub Form_Unload(Cancel As Integer)**

MsgBox "Ini adalah Unload Form2"

End Sub**Private Sub Form_Terminate()**

MsgBox "Ini adalah terminate form2"

End Sub

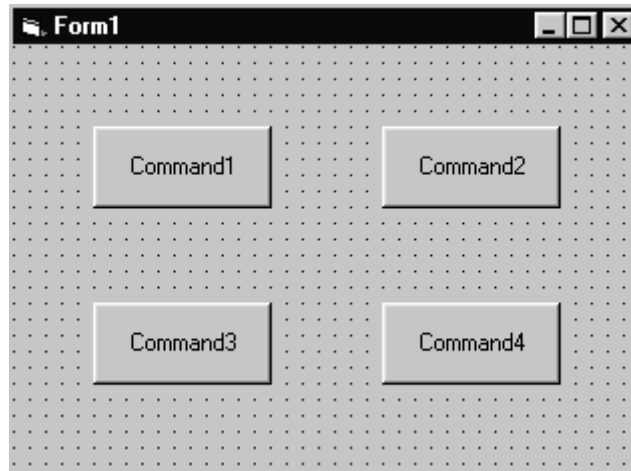
Dan akhirnya lakukan juga koding untuk Command1 pada Form2

Private Sub Command1_Click()

Unload Me

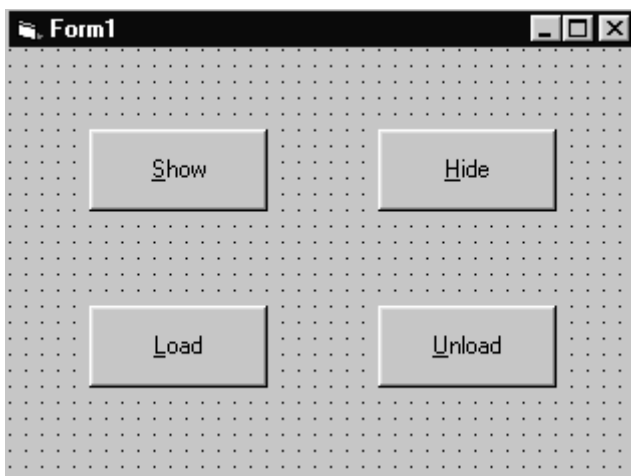
End Sub

3. Tampilkan kembali Form1. (caranya : Double Klik pada Form1 di jendela Project Explorer), dan kemudian tanamkan Command1 s/d 4 di atasnya, dan lakukan koding berikut :



Kemudian atur properti masing-masing menurut tabel berikut :

Kontrol	Properti	Value
Command1	Name Caption	cmdShow &Show
Command2	Name Caption	cmdHide &Hide
Command3	Name Caption	cmdLoad &Load
Command4	Name Caption	cmdUnLoad &UnLoad



Lakukan koding untuk masing-masing Command

```

Private Sub cmdShow_Click()
Form2.Show
End Sub
    
```

```
Private Sub Form_Resize()  
MsgBox "Ini adalah Resize Form2"  
End Sub
```

```
Private Sub cmdHide_Click()  
Form2.Hide  
End Sub
```

```
Private Sub cmdLoad_Click()  
Load Form2  
End Sub
```

```
Private Sub cmdUnload_Click()  
Unload Form2  
End Sub
```

4. Simpan Project anda FormEvent.vbp, Form1.frm, Form2.frm

5. Jalankan project anda

- Lakukan klik pada Show, dan perhatikan tulisan pada message box, yang menunjukkan urutan Event pada Form2, klik Ok untuk menutup masing-masing MsgBox.
- Alih keaktifan ke Form1 tanpa menutup Form2, perhatikan tulisan pada message box, yang menunjukkan Event yang terjadi.
- Aktifkan kembali ke Form2 dengan klik pada Form2 (bukan klik pada Command1 di Form1), perhatikan Event yang terjadi.
- Tutup Form2, dengan klik pada Command1, dan perhatikan urutan Event yang terjadi.
- Buka kembali Form2, dengan klik pada Command1 diatas Form1, dan perhatikan urutan Event yang terjadi (apakah Event Initialize dijalankan ?)
- Tutup Form2
- Tutup Form1, dan perhatikan Event yang terjadi (mengapa Event Terminate Form2 terjadi ?)

6. Jalankan project anda

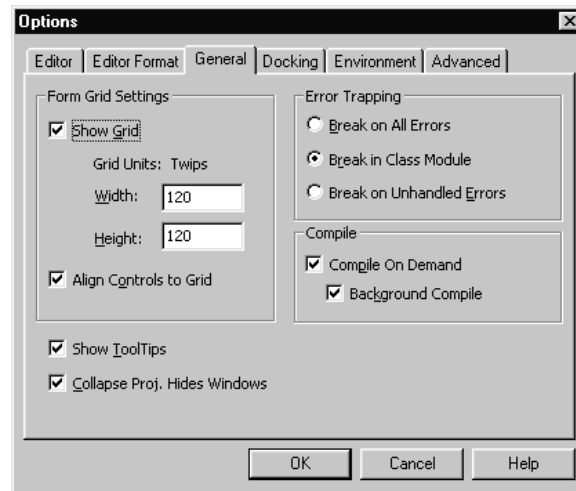
- Lakukan klik pada Load, dan perhatikan tulisan pada message box, yang menunjukkan urutan Event pada Form2, klik Ok untuk menutup masing-masing MsgBox.
- Lakukan klik pada Show, dan perhatikan Event yang terjadi.

7. Lakukan percobaan sendiri untuk menambah pengertian anda

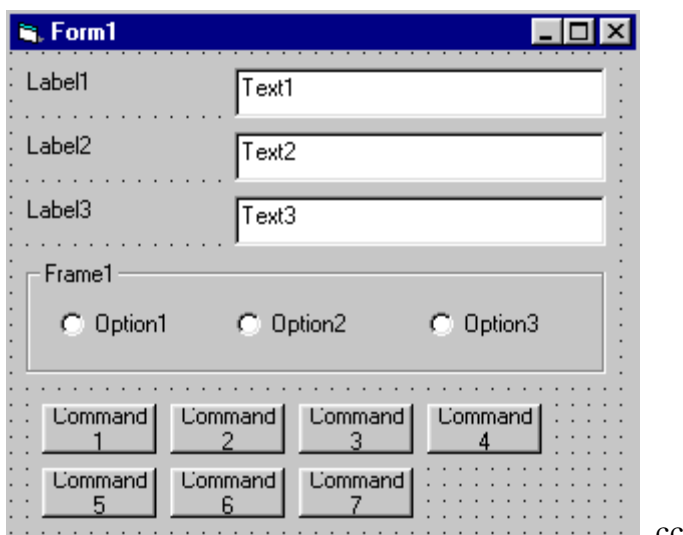
Latihan 5

Judul : Pengaturan Form, Properti, dan Event yang diterapkan pada pengolahan Random File

1. Buatlah suatu project baru, dan periksalah setting grid pada form anda, dengan menu Tools, Option, General. Adapun hal yang perlu diperhatikan adalah keaktifan **Show Grid**, dan **Align Control to Grid**

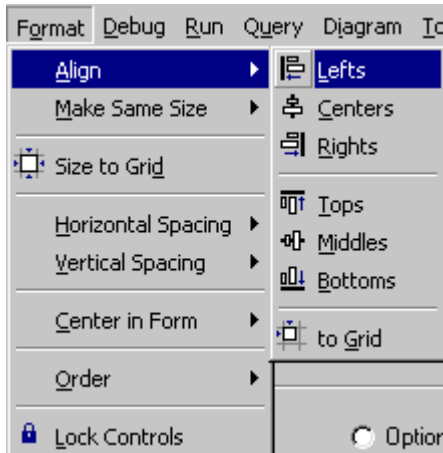


Kemudian tanamkan kontrol-kontrol berikut pada form1 sehingga membentuk tampilan sebagai berikut, anda harus menanamkan kontrol Frame terlebih dahulu baru diikuti dengan Option1 s/d 3 diatasnya, dalam hal ini Kontrol Frame1 bertindak sebagai kontainer bagi Option1 s/d 3.



CC

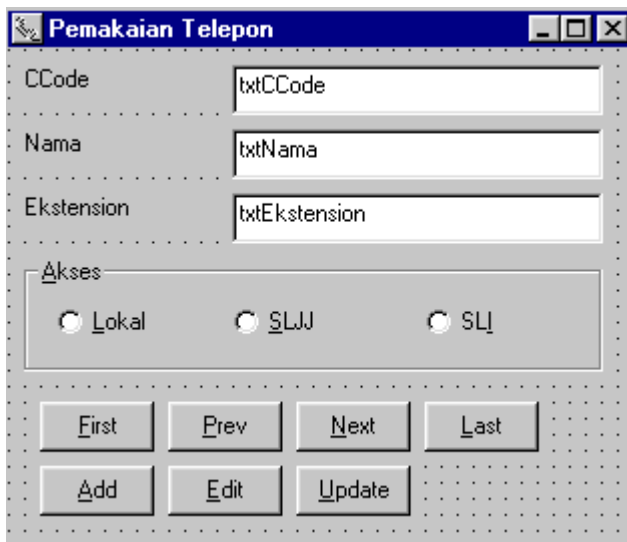
Pilihlah kontrol Label1, Label2, dan Label3 (caranya : klik pada Label1, kemudian tekan tombol Ctrl dan jangan di lepas, klik pada Label2, dan klik pada Label3, akhirnya tombol Ctrl di lepas), kemudian pada menu **Format** Pilih **Align** pilih **Left**.



Pilih Text1, Text2, dan Text3, dan, kemudian pada menu **Format**, pilih **Make Same Size**, pilih **Both**, dan juga dibuat rata Kiri.

Lakukan juga proses diatas untuk Option1 s/d 3, Command1 s/d 7

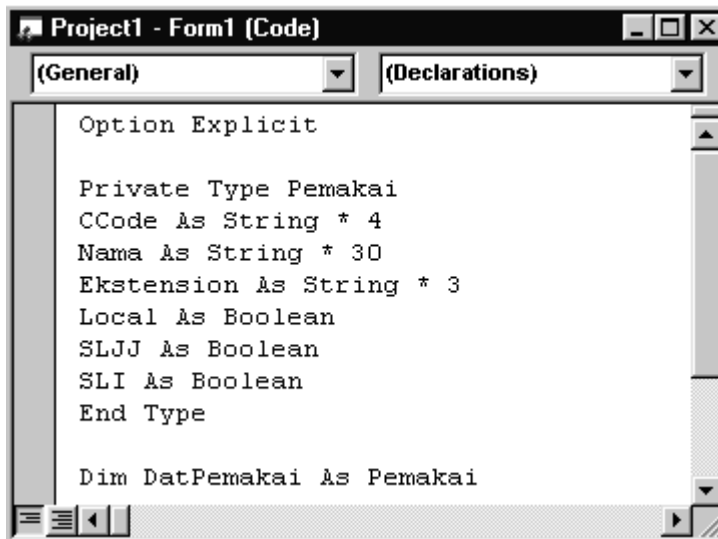
2. Aturlah properti dari masing-masing kontrol sehingga menjadi sebagai berikut, dan jangan lupa untuk mengatur tabindex dengan urutan txtCCode, txtNama,txtEkstension, fraAkses, optLokal, optSLJJ, optSLI, baru kemudian urutan untuk commandbutton :



Kontrol	Properti	Nilai
Form1	Name Caption	frmPemakai Pemakaian Telepon

	Icon	
Label1	Caption	CCode
Label2	Caption	Nama
Label3	Caption	Ekstension
Text1	Text Name	txtCCode txtCCode
Text2	Text Name	txtNama txtNama
Text3	Text Name	txtEkstension txtEkstension
Frame1	Caption Name	&Akses fraAkses
Option1	Caption Name	&Lokal cmdLokal
Option2	Caption Name	&SLJJ cmdSLJJ
Option3	Caption Name	SL&I cmdSLI
Command1	Caption Name	&First cmdFirst
Command2	Caption Name	&Prev cmdPrev
Command3	Caption Name	&Next cmdNext
Command4	Caption Name	&Last cmdLast
Command5	Caption Name	&Add cmdAdd
Command6	Caption Name	&Edit cmdEdit
Command7	Caption Name	&Update cmdUpdate

3. Aktifkan ke jendela Koding, dengan menekan tombol F7, atau pada jendela Project Explorer, klik pada frmPemakai, dan klik pada icon **View Code**, dan pilih General, Declaration



dan ketikkan kode berikut :

Option Explicit

'Deklarasi enumerated type

```
Private Enum Flag
flNone = 0
flAdd = 1           'Tambah data
flEdit = 2         'Perbaiki data
End Enum
```

'Struktur data untuk random file

```
Private Type Pemakai
CCode As String * 4       'Password PABX
Nama As String * 30      'Nama Pemakai
Ekstension As String * 3 'Ekstension
Local As Boolean         'Akses Lokal
SLJJ As Boolean          'Akses SLJJ
SLI As Boolean           'Akses SLI
End Type
```

'Deklarasi variabel Form Level

```
Dim DatPemakai As Pemakai 'Variabel untuk menampung data Pemakai
Dim Posisi As Integer     'Untuk mencatat nomor record yang sedang
ditampilkan
Dim JlhRec As Integer     'Untuk mencatat jumlah record dalam random file
Dim Aksi As Integer       'Untuk flag Tambah Data, Perbaiki Data, atau None
```

Sub Kunci()

```
'Mengunci kontrol Text, dan OptionButton
txtCCode.Locked = True
```

```
txtNama.Locked = True
txtEkstension.Locked = True
FraAkses.Enabled = False      'Kalau framenya disable, maka isinya tidak dapat
diakses
End Sub
```

Sub Buka()

```
txtCCode.Locked = False
txtNama.Locked = False
txtEkstension.Locked = False
FraAkses.Enabled = True
End Sub
```

Sub Tampil()

```
'Sub rutin ini digunakan untuk menampilkan isi record ke
'Masing-masing kontrol yang bersesuaian
If Posisi > 0 Then           'Jika posisi record lebih besar dari 0
    Get #1, Posisi, DatPemakai 'Baca record berdasarkan posisi
    txtCCode.Text = DatPemakai.CCode
    txtNama.Text = DatPemakai>Nama
    txtEkstension.Text = DatPemakai.Ekstension
    optLocal.Value = DatPemakai.Local
    optSLJJ.Value = DatPemakai.SLJJ
    optSLI.Value = DatPemakai.SLI
Else
    Call Kosong
End If
Aksi = flNone                'Tandai Flag Aksi adalah None
Call Kunci                   'Buat semua kontrol tidak dapat diperbaiki
End Sub
```

Sub Kosong()

```
'Subrutin ini akan mengosongkan semua nilai pada masing-masing kontrol
txtCCode.Text = ""
txtNama.Text = ""
txtEkstension.Text = ""
optLocal.Value = False
optSLJJ.Value = False
optSLI.Value = False
End Sub
```

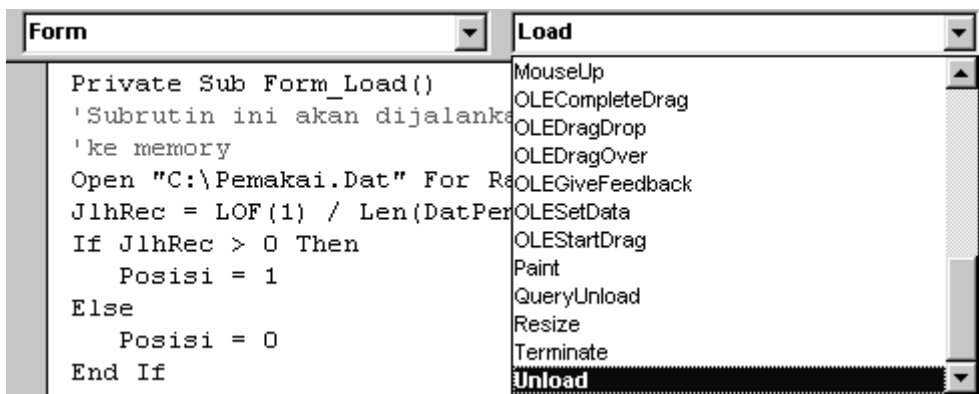
4. Kembali ke tampilan Form1 (gunakan Shift-F7 atau icon view Object pada Project Explorer), dan double klik pada frmPemakai, sehingga muncul Event Procedure Form_Load, dan ketiklah kode berikut :

Private Sub Form_Load()

```
'Subrutin ini akan dijalankan ketika form di load
'ke memory
Open "C:\Pemakai.Dat" For Random As #1 Len=Len(DatPemakai)
'Membuka random file sebagai file nomor 1

JlhRec = LOF(1) / Len(DatPemakai) 'Menghitung jumlah record
If JlhRec > 0 Then 'Jika jumlah record > 0
    Posisi = 1 'Posisi menunjuk ke record 1
Else
    Posisi = 0
End If
Call Tampil
End Sub
```

Beralih ke Event Procedure Unload, perhatikan gambar berikut



dan ketiklah kode berikut :

Private Sub Form_Unload(Cancel As Integer)

```
Dim nPil As Integer
nPil = MsgBox("Apakah Anda Yakin", vbYesNo) 'Konfirmasi penutupan
If nPil = vbYes Then 'Jika di jawab Yes
    Close #1 'Tutup file 1
    Cancel = 0 'Proses tutup form dilakukan
Else
    Cancel = 1 'Proses tutup form dibatalkan
End If
End Sub
```

dan lakukan juga koding untuk kontrol masing-masing :

Private Sub cmdFirst_Click()

```
If JlhRec > 0 Then
    Posisi = 1
    Call Tampil
```

End If

End Sub

Private Sub cmdPrev_Click()

If Posisi > 1 Then

 Posisi = Posisi - 1

 Call Tampil

End If

End Sub

Private Sub cmdNext_Click()

If Posisi < JlhRec Then

 Posisi = Posisi + 1

 Call Tampil

End If

End Sub

Private Sub cmdLast_Click()

Posisi = JlhRec

Call Tampil

End Sub

Private Sub cmdAdd_Click()

Aksi = flAdd

Call Buka

Call Kosong

txtCCode.SetFocus

End Sub

Private Sub cmdEdit_Click()

Aksi = flEdit

Call Buka

txtCCode.SetFocus

End Sub

Private Sub cmdUpdate_Click()

If Aksi = flAdd Or Aksi = flEdit Then

 If Aksi = flAdd Then

 JlhRec = JlhRec + 1

 Posisi = JlhRec

 End If

 DatPemakai.CCode = txtCCode.Text

 DatPemakai>Nama = txtNama.Text

 DatPemakai.Ekstension = txtEkstension.Text

 DatPemakai.Local = optLocal.Value

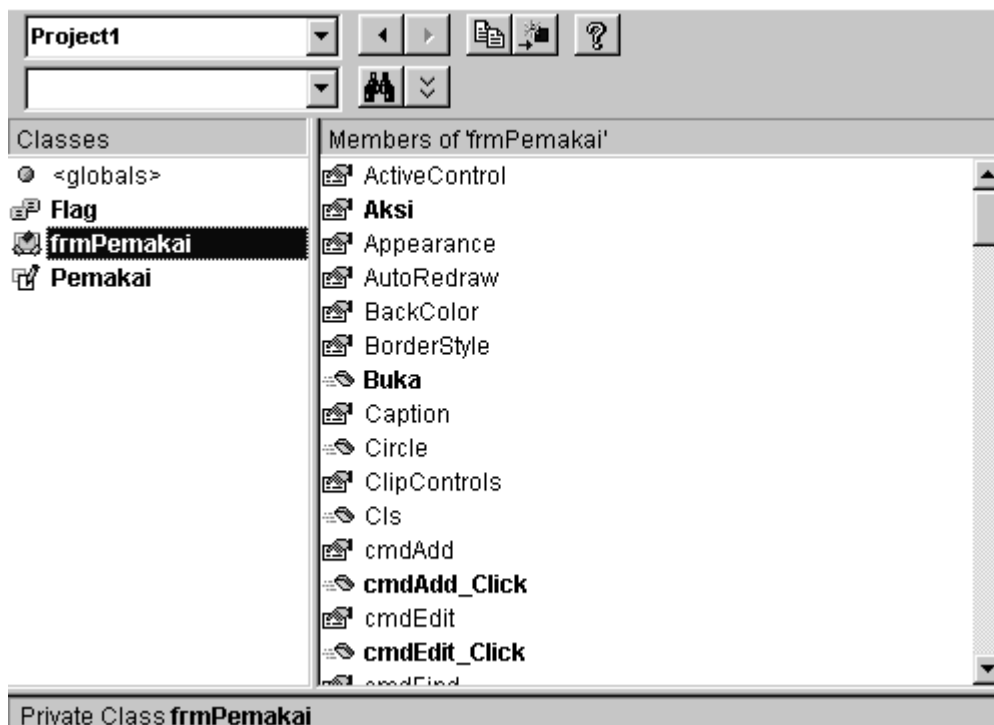
 DatPemakai.SLJJ = optSLJJ.Value

```
DatPemakai.SLI = optSLI.Value
Put #1, Posisi, DatPemakai
Call Kunci
End If
End Sub
```

5. Gunakan menu Format, Lock Controls, untuk mengunci semua kontrol yang berada pada form agar tidak dapat digeser dan diubah ukurannya, hal ini cocok dilakukan pada form yang telah selesai.

6. Simpan project diatas sebagai Pemakai.vbp, dan Pemakai.frm

7. Aktifkan Objek browser, dan amati objek-objek yang ada pada Project1, dengan menu View, Object Browser, kemudian pilih Project1.



Latihan 6

Judul : Tantangan pemrograman

Lengkapi program diatas dengan fasilitas Find, dimana jika tombol Find ditekan akan muncul suatu InputBox yang menanyakan nama yang dicari, jika ketemu, akan ditampilkan, dan jika tidak ketemu akan muncul message box data tidak ada.

```
Private Sub cmdFind_Click()  
Dim nama As String  
nama = InputBox("Masukkan nama :", "Pencarian")  
If nama <> "" Then  
    'lengkapi bagian ini  
  
End If  
End Sub
```

Modul 5, Kontrol Standard (bagian 1)



Kontrol standard terdapat pada semua versi Visual Basic, baik pada Learning Edition, Profesional Edition, maupun Enterprised Edition.


Kontrol-kontrol standard ini akan sering anda gunakan untuk pembentukkan user interface pada setiap project yang anda buat. Pada bagian ini kita akan membahas beberapa kontrol standard beserta properti-properti, event-event dan metoda-metoda yang bekerja pada masing-masing kontrol. Dengan memahami properti, event dan metoda akan sangat membantu anda dalam memanfaatkan kontrol-kontrol tersebut secara efektif dan efisien.


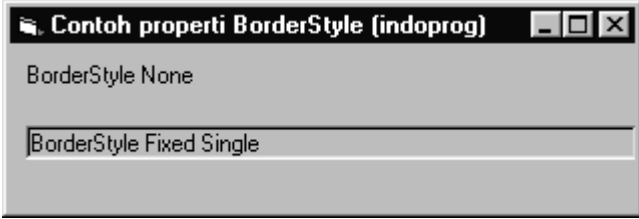
Pada Modul ini kita akan membahas Label, Textbox, Option, Check, Frame dan Command.

Label



Digunakan untuk menampilkan text tanpa bisa diubah oleh pemakai pada saat runtime. Beberapa properti pada label :

Properti	Fungsi
Alignment	Digunakan untuk menentukan pemerataan tulisan pada kontrol label (0 - Left Justify, 1 - Right Justify, 2 - Center) 
Appearance	Digunakan untuk menentukan tampilan dari kontrol label di cat atau tidak (0 - Flat, 1 - 3D)
AutoSize	Digunakan untuk menentukan apakah ukuran kontrol otomatis disesuaikan dengan ukuran tulisan (False, True)
BackColor	Digunakan untuk menentukan warna latar belakang (properti ini saling berkaitan dengan properti Appearance)
BackStyle	Digunakan untuk menentukan perilaku latar belakang kontrol label (0 - Transparent, 1 - Opaque), kalau transparan, maka BackColor menjadi tidak berarti, demikian juga Appearance.

	
BorderStyle	<p>Digunakan untuk menentukan bentuk border, apakah (0- None, 1 - Fixed Single)</p> 
Caption	Digunakan untuk menentukan tulisan pada kontrol label, mungkin ini adalah properti yang paling sering anda gunakan.
DragIcon	Digunakan untuk menentukan icon yang digunakan pada operasi drag-n-drop pada kontrol label
DragMode	Menentukan modus drag-n-drop (0 - manual, 1 - automatic)
Enabled	Menentukan apakah kontrol label dapat menerima event yang dibuat oleh pemakai, misalnya event Click, MouseDown, MouseUp, dll.
Font	Digunakan untuk menentukan jenis font, ukuran, style, dll
ForeColor	Digunakan untuk menentukan warna tulisan
Height	Digunakan untuk menentukan tinggi dari kontrol
Index	Digunakan untuk menentukan nomor index, jika kontrol tersebut merupakan kontrol array.
Left	Digunakan untuk menentukan jarak kiri kontrol dari kontainernya.
MouseIcon	Digunakan untuk menentukan icon mouse yang digunakan ketika pointer mouse berada diatas kontrol label tersebut, properti ini berarti bila properti MousePointer diset menjadi (99 - Custom)
MousePointer	Digunakan untuk menentukan bentuk pointer yang digunakan ketika mouse pointer berada diatas kontrol label tersebut.
Tag	Properti ini dapat digunakan sebagai tempat menyimpan data sementara yang berkaitan dengan kontrol label tersebut
ToolTipText	Digunakan untuk menentukan tip tulisan ketika mouse pointer berada diatas kontrol
Top	Digunakan untuk menentukan jarak atas kontrol dari kontainernya.
UseMnemonic	Digunakan untuk menentukan perlakuan terhadap tanda & ampersand sebagai access key, atau sebagai simbol & (False, True)
Visible	Digunakan untuk menentukan apakah kontrol label tersebut Visible atau tidak (False, True)
Width	Digunakan untuk menentukan lebar kontrol label
WordWrap	Digunakan untuk menentukan apakah tulisan dalam kontrol label dapat dilipat menjadi beberapa baris.



Gambar 3-1, Contoh properti pada kontrol Label

Adapun event-event yang efektif pada kontrol label adalah :

Event	Keterangan
Change	Event ini terjadi ketika nilai properti Caption dari kontrol
Click	Event ini terjadi ketika pemakai melakukan klik pada kontrol, event ini terjadi dengan urutan (MouseDown, MouseUp, Click)
DbClick	Event ini terjadi ketika pemakai melakukan double klik pada kontrol, event ini terjadi dengan urutan (MouseDown, MouseUp, Click, MouseDown, DbClick, MouseUp)
MouseDown	Event ini terjadi ketika pemakai melakukan penekanan tombol mouse pada kontrol, parameter yang dapat digunakan adalah Button (1 - Tombol kiri, 2 - Tombol kanan), Shift (1 - Shift, 2 - Ctrl, 3 - Alt), X (posisi kolom pointer), Y (posisi baris pointer)
MouseMove	Event ini terjadi ketika pamakai menggerakan mouse pada kontrol, parameter yang dapat digunakan adalah Button (1 - Tombol kiri, 2 - Tombol kanan), Shift (1 - Shift, 2 - Ctrl, 3 - Alt), X (posisi kolom pointer), Y (posisi baris pointer)
MouseUp	Event ini terjadi ketika pemakai melepaskan penekanan tombol mouse pada kontrol, parameter yang dapat digunakan adalah Button (1 - Tombol kiri, 2 - Tombol kanan), Shift (1 - Shift, 2 - Ctrl, 3 - Alt), X (posisi kolom pointer), Y (posisi baris pointer)


Adapun metoda yang efektif pada kontrol label adalah :

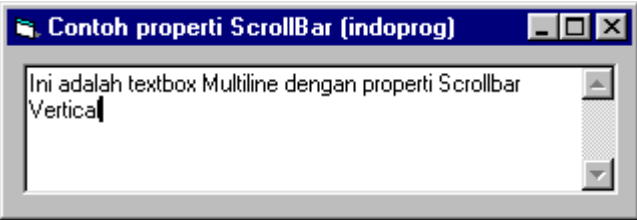
Metoda	Keterangan
Move	Metoda ini digunakan untuk memindahkan letak (koordinat Left, Top) dari kontrol label maupun ukurannya (ukuran Height, Width)
Refresh	Metoda ini digunakan untuk mencetak ulang kontrol label.
ZOrder	Metoda ini digunakan untuk menentukan order kontrol, apakah diatas atau dibawah kontrol lain.

Textbox



Digunakan untuk menampilkan text yang dapat diubah oleh pemakai pada saat runtime

Properti	Fungsi
Alignment	sda
Appearance	sda
BackColor	sda
BorderStyle	sda
CauseValidation	Digunakan untuk menentukan apakah event Validate di aktifkan ketika kontrol akan kehilangan fokus (False, True)
DragIcon	sda
DragMode	sda
Enabled	sda
Font	sda
ForeColor	sda
Height	sda
HideSelection	Digunakan untuk menentukan apakah selection disembunyikan ketika kontrol kehilangan fokus
Index	sda
Left	Digunakan untuk menentukan jarak kiri kontrol dari kontainernya.
Locked	Digunakan untuk menentukan apakah text dalam kontrol textbox dapat di perbaharui oleh pemakai atau tidak (False, True)
MaxLength	Digunakan untuk menentukan jumlah huruf maksimal yang dapat diketikkan dalam textbox (0 s/d 65535), 0 tidak dibatasi.
MouseIcon	Digunakan untuk menentukan icon mouse yang digunakan ketika pointer mouse berada diatas kontrol label tersebut, properti ini berarti bila properti MousePointer diset menjadi (99 - Custom)
MousePointer	Digunakan untuk menentukan bentuk pointer yang digunakan ketika mouse pointer berada diatas kontrol label tersebut.
MultiLine	Digunakan untuk menentukan apakah tulisan dalam textbox dapat ditampilkan dalam bentuk beberapa baris, dan mengenali Enter untuk memasuki baris baru. (False, True) 
PasswordChar	Digunakan untuk menentukan karakter yang digunakan untuk menyandikan tampilan huruf yang diketik oleh pemakai. Biasanya digunakan untuk pengisian password, dimana huruf yang ditekan ditampilkan dalam bentuk *
ScrollBars	Digunakan untuk menentukan penampilan ScrollBar pada textbox, properti ini efektif jika setting properti MultiLine adalah True (0 - None, 1 - Horizontal, 2 - Vertical, 3 - Both)

	
TabIndex	Digunakan untuk menentukan urutan tabindex, 0 adalah kontrol yang pertama mendapat fokus jika form dijalankan. Properti ini efektif jika setting properti TabStop adalah True
TabStop	Digunakan untuk menentukan apakah textbox dapat menerima fokus karena penekanan tombol TAB oleh pemakai (False, True)
Tag	sda
Text	Properti ini berisi tulisan yang berada dalam textbox
ToolTipText	sda
Top	sda
Width	sda



Gambar 3-2, Contoh properti pada kontrol Text

Adapun event-event yang efektif pada kontrol TextBox adalah :

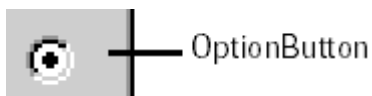
Event	Keterangan
Change	Event ini terjadi ketika nilai properti Text dari kontrol mengalami perubahan
Click	sda
DbClick	sda
GotFocus	Event ini terjadi ketika kontrol mendapatkan fokus
KeyDown	Event ini terjadi ketika pemakai mengetik karakter pada kontrol
KeyPress	Event ini terjadi setelah event KeyDown ketika pemakai mengetik karakter pada kontrol, parameter yang dapat digunakan adalah KeyAscii. Anda dapat memanipulasi tombol yang diketik dengan melakukan perubahan pada nilai KeyAscii. Contoh : Private Sub txtNama_KeyPress(KeyAscii As Integer)

	<pre>If KeyAscii = 13 Then SendKeys "{Tab}" End If End Sub</pre>
KeyUp	Event ini terjadi setelah event KeyPress ketika pemakai mengetik karakter pada kontrol dengan urutan (KeyDown, KeyPress, KeyUp, Change)
LostFocus	Event ini terjadi ketika kontrol kehilangan fokus
MouseDown	sda
MouseMove	sda
MouseUp	sda
Validate	Event ini terjadi ketika kontrol akan kehilangan fokus.

Adapun metoda yang efektif pada kontrol Textbox adalah :

Metoda	Keterangan
Move	sda
Refresh	sda
Setfocus	Metoda ini digunakan untuk memindahkan fokus ke kontrol yang bersangkutan
ZOrder	sda

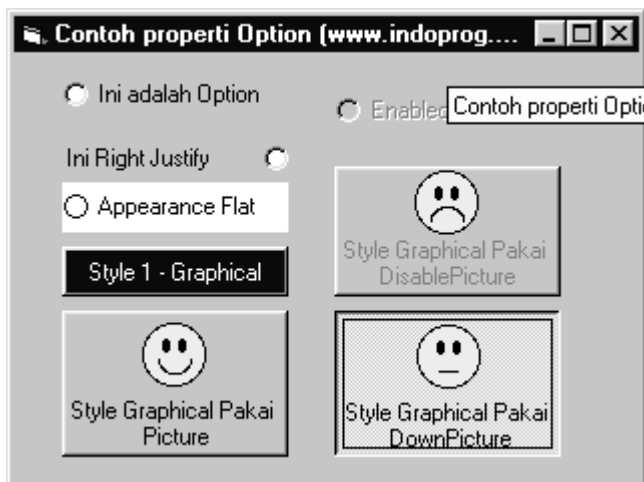
Option



Digunakan untuk menampilkan beberapa pilihan yang hanya dapat dipilih salah satu dalam suatu form, untuk

Properti	Fungsi
Alignment	sda
Appearance	sda
BackColor	sda
Caption	sda
CauseValidation	sda
DisablePicture	Digunakan untuk menentukan gambar yang ditampilkan ketika kontrol Option disable (properti Enabled = False), properti ini efektif jika setting properti Style adalah (1- Graphical)
DownPicture	Digunakan untuk menentukan gambar yang ditampilkan ketika kontrol Option dipilih (properti Value = True), properti ini efektif jika setting properti Style adalah (1- Graphical)
DragIcon	sda
DragMode	sda
Enabled	sda
Font	sda
ForeColor	sda
Height	sda
Index	sda
Left	Digunakan untuk menentukan jarak kiri kontrol dari kontainernya.
MaskColor	Digunakan untuk warna pada bitmap yang akan dijadikan transparan, properti ini efektif jika picture yang masukkan adalah (*.bmp), dan setting properti UseMaskColor adalah True

MouseIcon	sda
MousePointer	sda
Picture	Digunakan untuk menentukan gambar yang digunakan, properti ini efektif jika setting properti Style adalah (1 - Graphical)
Style	Digunakan untuk menentukan jenis style pada kontrol option (0 - Standard, 1 - Graphical)
TabIndex	sda
TabStop	sda
Tag	sda
ToolTipText	sda
Top	sda
UseMaskColor	Digunakan akan warna yang ditentukan pada MaskColor efektif atau tidak (False, True)
Value	Adalah nilai kontrol option button (True - terpilih, False - tidak terpilih)
Visible	Digunakan untuk menentukan apakah kontrol Visibel atau tidak.
Width	sda



Gambar 3-3, Contoh properti pada kontrol Option

Adapun event-event yang efektif pada kontrol option adalah :

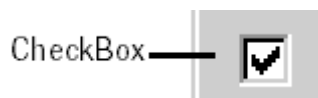
Event	Keterangan
Click	sda, pada option event ini juga dibangkitkan ketika kontrol menerima fokus, maupun ketika pemakai menekan space pada kontrol bersangkutan.
DblClick	sda
GotFocus	sda
KeyDown	sda
KeyPress	sda
KeyUp	sda
LostFocus	sda
MouseDown	sda
MouseMove	sda

MouseUp	sda
Validate	sda

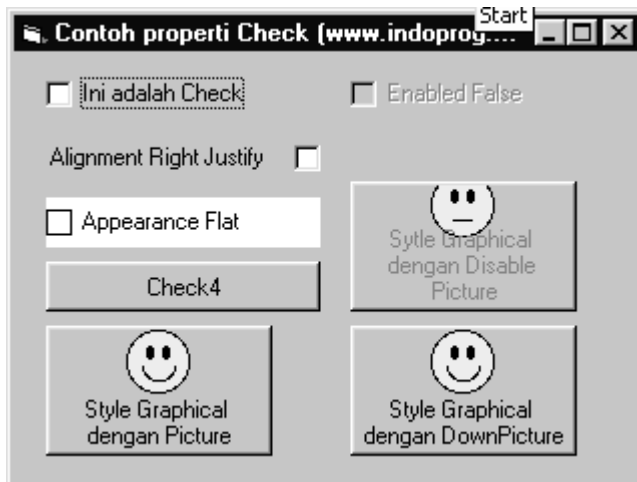
Adapun metoda yang efektif pada kontrol Option adalah :

Metoda	Keterangan
Move	sda
Refresh	sda
Setfocus	sda, pada kontrol option, metoda ini akan membangkitkan event Click.
ZOrder	sda

Check



Digunakan untuk menampilkan beberapa pilihan yang dapat dipilih lebih dari satu



Gambar 3-4, Contoh properti pada kontrol Check

Properti kontrol Check, maupun event dapat dilihat pada properti Option

Frame



Digunakan untuk mengelompokkan sekelompok kontrol. Pemakaian kontrol frame yang paling nyata adalah untuk mengelompokkan sejumlah option, sebagaimana kita ketahui,

pada suatu form, hanya 1 option yang dapat dipilih setiap saat, hal ini dapat diatasi dengan pemakaian frame, sehingga option dapat dipilih sesuai dengan konteks yang diwakili.

Properti	Fungsi
Appearance	sda
BackColor	sda
BorderStyle	sda
Caption	sda
CauseValidation	sda
DragIcon	sda
DragMode	sda
Enabled	sda
Font	sda
ForeColor	sda
Height	sda
Index	sda
Left	Digunakan untuk menentukan jarak kiri kontrol dari kontainernya.
MouseIcon	sda
MousePointer	sda
TabIndex	sda
Tag	sda
ToolTipText	sda
Top	sda
Visible	Digunakan untuk menentukan apakah kontrol Visibel atau tidak, jika Visible false, maka semua kontrol yang berada diatasnya menjadi tidak kelihatan.
Width	sda



Gambar 3-5, Contoh properti pada kontrol Frame

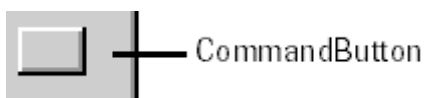
Adapun event-event yang efektif pada kontrol frame adalah :

Event	Keterangan
Click	sda
DbClick	sda
MouseDown	sda
MouseMove	sda
MouseUp	sda

Adapun metoda yang efektif pada kontrol Frame adalah :

Metoda	Keterangan
Move	sda
Refresh	sda
ZOrder	sda

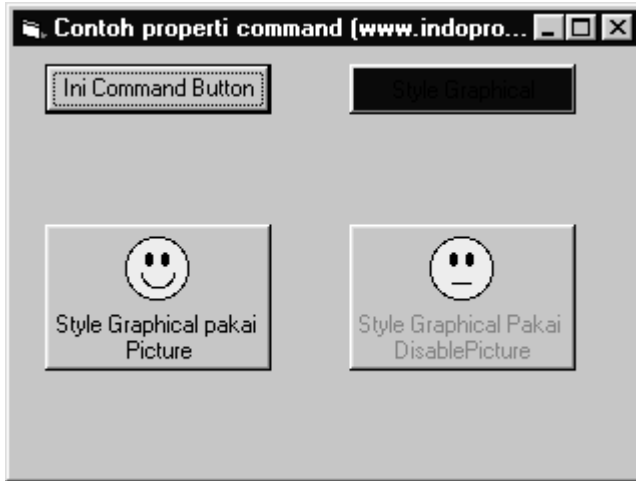
CommandButton



Digunakan untuk mendapatkan konfirmasi pemakai untuk pelaksanaan fungsi tertentu.

Properti	Fungsi
Alignment	sda
Appearance	sda
BackColor	sda
Cancel	Digunakan untuk menentukan apakah command button merupakan perintah Cancel untuk form tersebut , jika properti ini True, maka ketika pemakai menekan ESC, akan membangkitkan event Click untuk kontrol tersebut.
Caption	sda
CauseValidation	sda
Default	Digunakan untuk menentukan apakan command button merupakan Default command untuk form tersebut, jika properti ini True, maka ketika pemakai menekan Enter, akan membangkitkan event Click untuk kontrol tersebut.
DragIcon	sda
DragMode	sda
Enabled	sda
Font	sda
ForeColor	sda
Height	sda
Index	sda
Left	Digunakan untuk menentukan jarak kiri kontrol dari kontainernya.
MaskColor	Digunakan untuk warna pada bitmap yang akan dijadikan transparan, properti ini efektif jika picture yang masukkan adalah (*.bmp), dan setting properti UseMaskColor adalah True
MouseIcon	sda
MousePointer	sda
Picture	sda
Style	sda

TabIndex	sda
TabStop	sda
Tag	sda
ToolTipText	sda
Top	sda
UseMaskColor	sda
Visible	Digunakan untuk menentukan apakah kontrol Visibel atau tidak.
Width	sda



Gambar 3-6, Contoh properti pada kontrol Command

Adapun event-event yang efektif pada kontrol commandbutton adalah :

Event	Keterangan
Click	sda, pada commandbutton, event ini juga dibangkitkan ketika pemakai menekan space pada kontrol tersebut.
GotFocus	sda
KeyDown	sda
KeyPress	sda
KeyUp	sda
LostFocus	sda
MouseDown	sda
MouseMove	sda
MouseUp	sda

Adapun metoda yang efektif pada kontrol CommandButton adalah :

Metoda	Keterangan
Move	sda
Refresh	sda
Setfocus	sda
ZOrder	sda

Latihan 7

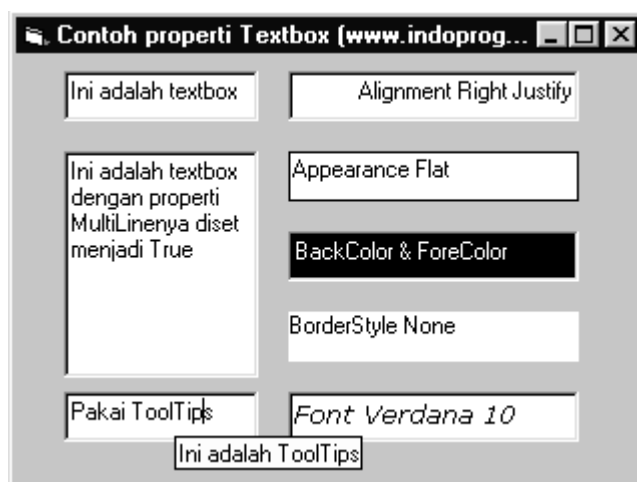
Judul : Mengenal Properti Label, Text, Option, Check, Frame dan Command

1. Buatlah project baru yang berisi form, form berikut :

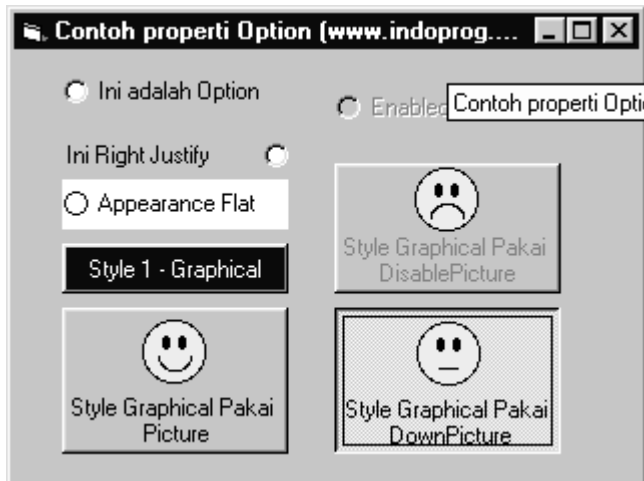
Form 1



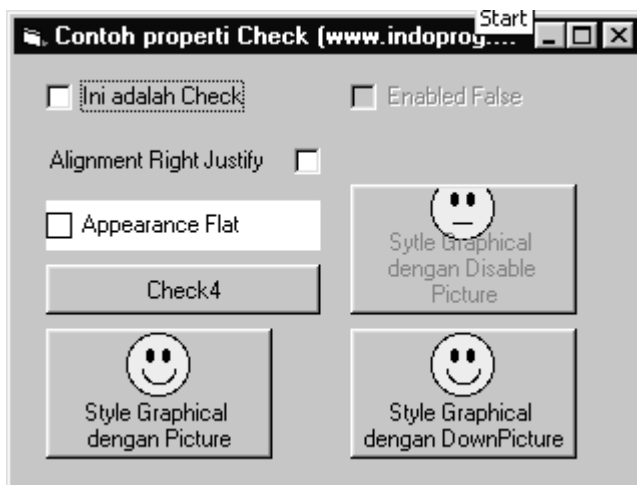
Form 2



Form 3



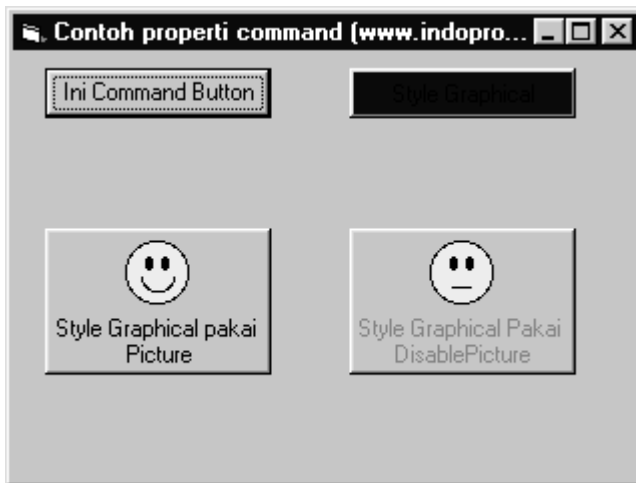
Form 4



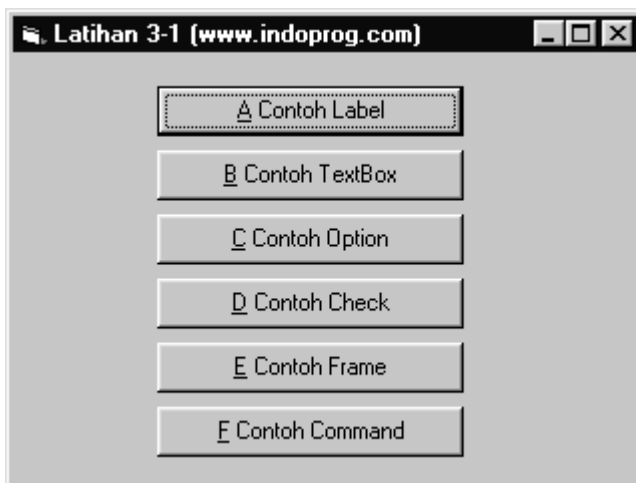
Form 5



Form 6



Form 7 (diset sebagai startup object)



Dengan fungsi masing-masing tombol untuk mengaktifkan masing-masing form yang bersesuaian.

Latihan 8

Judul : Memahami beberapa event yang umum pada kontrol.

1. Buatlah project baru, dengan Form1



Kontrol	Properti	Value
Text1	Name CauseValidation	txtUmur True
Text2	Name	txtCoba1
Text3	Name	txtCoba2
Command1	Name Caption Default	cmdDefault &Default True True
Command2	Name Caption Cancel	cmdCancel &Cancel True True

2. Lakukan Koding berikut :

```
Private Sub cmdCancel_Click()
MsgBox "cmdCancel Event Click"
End Sub
```

```
Private Sub cmdDefault_Click()
MsgBox "cmdDefault Event Enter"
End Sub
```

```
Private Sub txtCoba2_KeyDown(KeyCode As Integer, Shift As Integer)
Debug.Print "txtCoba2 Event Keydown dengan KeyCode :" & KeyCode & "
Shift : " & Shift
End Sub
```

```
Private Sub txtCoba2_KeyPress(KeyAscii As Integer)
```

```
Debug.Print "txtCoba2 Event KeyPressed dengan KeyAscii : " & KeyAscii
End Sub

Private Sub txtCoba2_KeyUp(KeyCode As Integer, Shift As Integer)
Debug.Print "txtCoba2 Event Keyup dengan KeyCode : " & KeyCode & "
Shift : " & Shift
End Sub

Private Sub txtCoba3_Click()
Debug.Print "txtCoba3 Event Click"
End Sub

Private Sub txtCoba3_DblClick()
Debug.Print "txtCoba3 Event DblClick"
End Sub

Private Sub txtCoba3_MouseDown(Button As Integer, Shift As Integer, X
As Single, Y As Single)
Debug.Print "txtCoba3 Event MouseDown dengan Button : " & Button & "
Shift : " & Shift & " X : " & X & " Y : " & Y
End Sub

Private Sub txtCoba3_MouseMove(Button As Integer, Shift As Integer, X
As Single, Y As Single)
Debug.Print "txtCoba3 Event MouseMove dengan Button : " & Button & "
Shift : " & Shift & " X : " & X & " Y : " & Y
End Sub

Private Sub txtCoba3_MouseUp(Button As Integer, Shift As Integer, X As
Single, Y As Single)
Debug.Print "txtCoba3 Event MouseUp dengan Button : " & Button & "
Shift : " & Shift & " X : " & X & " Y : " & Y
End Sub

Private Sub txtCoba1_GotFocus()
MsgBox "txtCoba1 menerima fokus"
End Sub

Private Sub txtCoba1_LostFocus()
MsgBox "txtCoba1 kehilangan fokus"
End Sub

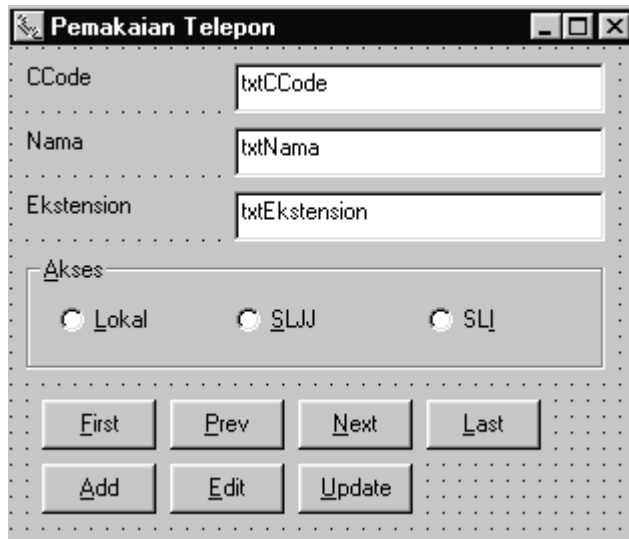
Private Sub txtCoba2_Change()
Debug.Print "txtCoba2 Change"
End Sub

Private Sub txtUmur_Validate(Cancel As Boolean)
If (Val(txtUmur) < 5) Or (Val(txtUmur) > 125) Then
    MsgBox "Masukkan umur 5 s/d 125" & vbCrLf & _
        "Event Validate"
    Cancel = True
End If
End Sub
```

Latihan 9

Judul : Tantangan melengkapi program data pemakai Telepon Latihan Bab2

1. Aktifkan program Latihan 2-2, modul sebelumnya.




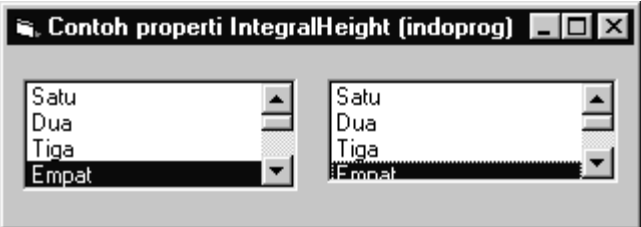
Lengkapi program tersebut dengan beberapa hal sebagai berikut :

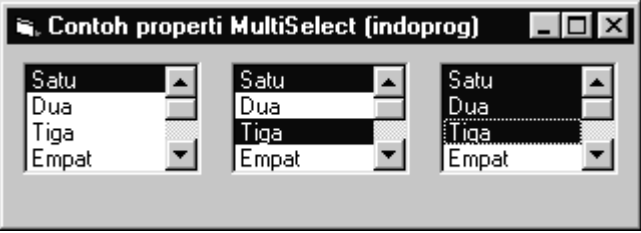
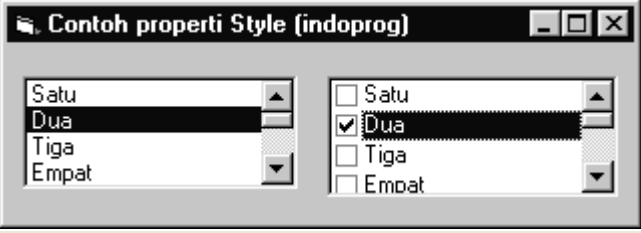
1. txtCCode, maksimal pengetikkan 4 huruf, dan tidak boleh kurang dari 4 huruf, dan yang dapat diketik hanya digit (0 s/d 9)
2. txtNama, maksimal pengetikkan 30 huruf, dan huruf pertama setiap suku kata otomatis besar, misalnya Hendra Wijaya, Susan Dewichan
3. txtEkstension, maksimal pengetikkan 3 huruf, dan yang dapat diketik hanya digit (0 s/d 9), dan nomor ekstension terkecil 100, terbesar 600.
4. Penekanan Enter dapat digunakan untuk berpindah antar textbox
5. Option SLI, hanya bisa dipilih untuk ekstension 100 s/d 199, SLJJ, hanya dipilih untuk ekstension 200 s/d 299
6. txtCCode, txtNama, txtEkstension, dan Akses tidak boleh kosong.

Modul 6, Kontrol Standard (bagian 2)

ListBox

Digunakan untuk menampilkan daftar pilihan yang dapat bergeser. Suatu listbox digunakan jika jumlah pilihan cukup banyak, sehingga menjadi tidak efektif kalau menggunakan Option maupun Check.

Properti	Fungsi
Appearance	Digunakan untuk menentukan tampilan dari kontrol (0 - Flat, 1 - 3D)
BackColor	Digunakan untuk menentukan warna latar belakang.
CauseValidation	Digunakan untuk menentukan apakah event Validate di aktifkan ketika kontrol akan kehilangan fokus (False, True)
Column	Digunakan untuk menentukan apakah daftar pilihan dalam listbox ditampilkan 1 kolom vertikal kebawah (0), atau kebawah dan menyamping sesuai dengan jumlah kolom yang ditentukan jika daftar pilihan cukup panjang. 
DragIcon	Digunakan untuk menentukan icon yang digunakan pada operasi drag-n-drop pada kontrol label
DragMode	Menentukan modus drag-n-drop (0 - manual, 1 - automatic)
Enabled	Menentukan apakah kontrol label dapat menerima event yang dibuat oleh pemakai, misalnya event Click, MouseDown, MouseUp, dll.
Font	Digunakan untuk menentukan jenis font, ukuran, style, dll
ForeColor	Digunakan untuk menentukan warna tulisan
Height	Digunakan untuk menentukan tinggi dari kontrol
Index	Digunakan untuk menentukan nomor index, jika kontrol tersebut merupakan kontrol array.
IntegralHeight	Digunakan untuk menentukan tinggi dari kontrol (Height) sedemikian rupa sehingga tidak ada item yang ditampilkan sebagian (False, True) 
ItemData	Adalah daftar yang dapat digunakan untuk menyimpan nilai numerik yang berkaitan dengan item-item yang terdapat dalam listbox. Berkoresponden satu-satu.
Left	Digunakan untuk menentukan jarak kiri kontrol dari kontainernya.
List	Merupakan properti yang berisi pilihan dalam Listbox, untuk mengetik itemnya anda harus menggunakan Ctrl-Enter untuk memasukkan item yang berikutnya.
MouseIcon	Digunakan untuk menentukan icon mouse yang digunakan ketika pointer mouse berada

	diatas kontrol label tersebut, properti ini berarti bila properti MousePointer diset menjadi (99 - Custom)
MousePointer	Digunakan untuk menentukan bentuk pointer yang digunakan ketika mouse pointer berada diatas kontrol label tersebut.
MultiSelect	Digunakan untuk menentukan apakah item dalam listbox dapat dipilih lebih dari satu atau tidak (0 - None, 1 - Simple, 2 - Extended) Perbedaan antara 1 - Simple dengan 2 - Extended, adalah pada Simple pemilihan beberapa item sekaligus tidak dapat menggunakan bantuan Shift. 
Sorted	Digunakan untuk menentukan apakah item dalam listbox otomatis diurut berdasarkan alfabetik atau tidak (False, True)
Style	Digunakan untuk menentukan bentuk tampilan item yang digunakan (0 - Standard, 1 - Checkbox) 
TabIndex	Digunakan untuk menentukan urutan tabindex, 0 adalah kontrol yang pertama mendapat fokus jika form dijalankan. Properti ini efektif jika setting properti TabStop adalah True
TabStop	Digunakan untuk menentukan apakah textbox dapat menerima fokus karena penekanan tombol TAB oleh pemakai (False, True)
Tag	Properti ini dapat digunakan sebagai tempat menyimpan data sementara yang berkaitan dengan kontrol label tersebut
ToolTipText	Digunakan untuk menentukan tip tulisan ketika mouse pointer berada diatas kontrol
Top	Digunakan untuk menentukan jarak atas kontrol dari kontainernya.
Visible	Digunakan untuk menentukan apakah kontrol label tersebut Visible atau tidak (False, True)
Width	Digunakan untuk menentukan lebar kontrol label

Menentukan isi ListBox

Untuk menentukan isi listbox anda dapat mengetikkannya pada waktu design dengan mengisinya di properti List. Misalnya daftar pilihan yang diinginkan adalah Satu, Dua, Tiga ..., Lima, maka :

- klik pada properti List,
- ketikkan Satu, tekan Ctrl-Enter
- ketikkan Dua, tekan Ctrl-Enter
- dst

Cara lain adalah dengan menggunakan metoda `AddItem(string, index)` pada Event Load pada Form, dimana string adalah item yang ingin ditambahkan ke ListBox, sedangkan index adalah posisi dimana item akan disisip, defaultnya adalah posisi terakhir. Contoh :

```
Private Sub Form_Load()  
    LstAngka.AddItem("Satu")  
    LstAngka.AddItem("Dua")  
    LstAngka.AddItem("Tiga")  
    LstAngka.AddItem("Empat")  
    LstAngka.AddItem("Lima")  
End Sub
```

Mendapatkan jumlah item dalam Listbox

Untuk mendapatkan jumlah item dalam Listbox anda dapat menggunakan properti `ListCount`.

Menggambil item yang dipilih oleh pemakai

Pada Listbox jenis `Multiselect = False`, maka item yang dapat dipilih oleh pemakai hanya satu, sehingga hal tersebut efektif diambil dengan menggunakan properti `Text` dari Listbox. Contoh berikut akan menampilkan item yang diklik oleh pemakai.

```
Private Sub LstAngka_Click()  
    MsgBox LstAngka.Text  
End Sub
```

Jika pada Listbox dengan `Multiselect = True`, maka pekerjaan menjadi sedikit lebih rumit, dimana kita perlu memeriksa item pada listbox satu persatu untuk memeriksa apakah item tersebut dalam keadaan terpilih (diperiksa dengan properti `selected(index)` atau tidak. Contoh berikut akan menampilkan kotak pesan yang berisi item-item yang terpilih.

```
Private Sub cmdPeriksa_Click()  
    For i = 0 To LstAngka.ListCount - 1  
        If LstAngka.Selected(i) Then  
            MsgBox LstAngka.List(i)  
        List  
        End If  
    Next i  
End Sub
```

Sesuatu hal yang perlu diingat bahwa nomor index item pada Listbox dimulai dari 0 s/d `ListCount - 1`

Mendapatkan nomor posisi item yang sedang difokus

Untuk mendapatkan posisi item yang sedang difokus pada Listbox dapat digunakan properti `ListIndex`

Menghapus Item pada Listbox

Untuk menghapus item pada Listbox, anda dapat menggunakan metoda `RemoveItem(index)`, dimana `index` adalah nomor index yang akan dihapus.

Contoh : Untuk memindahkan item yang terpilih pada ListBox kiri ke listbox kanan, dimana setting properti pada Listbox kiri `MultiSelect = False`.



Gambar 4-1, Contoh program yang menggunakan ListIndex dan RemoveItem

```
Private Sub cmdPindah_Click()
    If LstAngka.ListIndex > -1 Then
        LstPindah.AddItem (LstAngka.List(LstAngka.ListIndex))
        LstAngka.RemoveItem (LstAngka.ListIndex)
    End If
End Sub
```

Pada Listbox yang `MultiSelect = False`, properti `ListIndex` menunjukkan nomor index (mulai dari 0) item yang sedang terpilih, dan jika tidak ada yang terpilih nilainya -1.

Mengosongkan isi seluruh Listbox

Untuk mengosongkan isi seluruh Listbox, anda dapat menggunakan Metoda `Clear`.

Adapun event-event yang efektif pada Listbox adalah sebagai berikut :

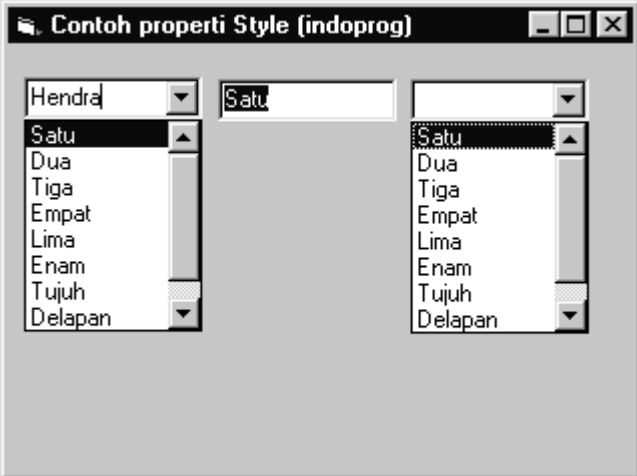
Event	Keterangan
Click	Event ini terjadi ketika pemakai melakukan klik pada kontrol, event ini terjadi dengan urutan (MouseDown, MouseUp, Click)
DbClick	Event ini terjadi ketika pemakai melakukan double klik pada kontrol, event ini terjadi dengan urutan (MouseDown, MouseUp, Click, MouseDown, DbClick, MouseUp)
GotFocus	Event ini terjadi ketika kontrol mendapatkan fokus
ItemCheck	Event ini terjadi ketika pemakai melakukan klik untuk menandai checkbox pada item Listbox, Event ini efektif pada ListBox yang <code>Style = Checkbox</code>
KeyDown	Event ini terjadi ketika pemakai mengetik karakter pada kontrol
KeyPress	Event ini terjadi setelah event <code>KeyDown</code> ketika pemakai mengetik karakter pada kontrol,

	parameter yang dapat digunakan adalah KeyAscii. Anda dapat memanipulasi tombol yang diketik dengan melakukan perubahan pada nilai KeyAscii.
KeyUp	Event ini terjadi setelah event KeyPress ketika pemakai mengetik karakter pada kontrol dengan urutan (KeyDown, KeyPress, KeyUp, Change)
LostFocus	Event ini terjadi ketika kontrol kehilangan fokus
MouseDown	Event ini terjadi ketika pemakai melakukan penekanan tombol mouse pada kontrol, parameter yang dapat digunakan adalah Button (1 - Tombol kiri, 2 - Tombol kanan), Shift (1 - Shift, 2 - Ctrl, 3 - Alt), X (posisi kolom pointer), Y (posisi baris pointer)
MouseMove	Event ini terjadi ketika pemakai menggerakkan mouse pada kontrol, parameter yang dapat digunakan adalah Button (1 - Tombol kiri, 2 - Tombol kanan), Shift (1 - Shift, 2 - Ctrl, 3 - Alt), X (posisi kolom pointer), Y (posisi baris pointer)
MouseUp	Event ini terjadi ketika pemakai melepaskan penekanan tombol mouse pada kontrol, parameter yang dapat digunakan adalah Button (1 - Tombol kiri, 2 - Tombol kanan), Shift (1 - Shift, 2 - Ctrl, 3 - Alt), X (posisi kolom pointer), Y (posisi baris pointer)
Scroll	Event ini terjadi ketika Listbox mengalami scroll.
Validate	Event ini terjadi ketika kontrol akan kehilangan fokus.

Combo Box

Jika dibandingkan dengan Listbox, maka ComboBox lebih menghemat pemakaian tempat pada form, dimana hasil pilihan pemakai ditampilkan dalam suatu textbox, dimana pilihan-pilihan dapat di drop-down dalam bentuk listbox. Pada Combobox pemakai juga dapat mengetik langsung pilihannya, tetapi hal ini sangat bergantung pada Style yang digunakan. Jika pada Listbox dimungkinkan pemakai melakukan MultiSelect, tetapi pada Combobox hal tersebut tidak dapat dilakukan.

Properti	Fungsi
BackColor	sda
CauseValidation	sda
DragIcon	sda
DragMode	sda
Enabled	sda
Font	sda
ForeColor	sda
Height	sda
Index	sda
IntegralHeight	sda
ItemData	sda
Left	sda
Locked	sda
MouseIcon	sda
MousePointer	sda
Sorted	sda
Style	Digunakan untuk menentukan bentuk tampilan combo yang digunakan (0 - Dropdown Combo, 1 - Simple Combo, 2 - Dropdown List), pada Dropdown Combo, pemakai diperbolehkan mengetik tulisan yang tidak ada di list, pada Simple Combo, pemakai boleh mengetik atau memilih dengan tombol keatas atau kebawah dan daftar tidak bisa terbuka, pada Dropdown List, tulisan hanya dapat dipilih dari daftar.

	
TabIndex	sda
TabStop	sda
Tag	sda
Text	sda
ToolTipText	sda
Top	sda
Visible	sda
Width	sda

Untuk pengolahan ComboBox tidak jauh berbeda dengan Listbox, untuk itu perhatikan kembali contoh-contoh pengolahan Listbox diatas dengan sedikit perbedaan karakteristik antara Listbox dan ComboBox.

Adapun event-event yang efektif pada ComboBox adalah sebagai berikut

Event	Keterangan
Click	sda
DbClick	sda
GotFocus	sda
KeyDown	sda
KeyPress	sda
KeyUp	sda
LostFocus	sda
MouseDown	sda
MouseMove	sda
MouseUp	sda
Validate	sda

Latihan 10

Judul : Memahami kontrol Listbox yang diaplikasikan untuk pengolahan nama negara dan propinsi

1. Buatlah program yang menyimpan nama-nama negara didalam sebuah text file Negara.txt, dimana setiap kali form diaktifkan, maka isi file Negara.txt akan dibaca dan ditampilkan dalam suatu listbox, kemudian pemakai dapat melakukan tambah, perbaiki maupun hapus item dalam listbox, dan isi daftar akan disimpan kembali ke file Negara.txt setiap pemakai melakukan klik pada Simpan ke file.



Kontrol	Properti	Nilai
Listbox1	Name MultiSelect Sorted	IstNegara 0 - None True
Text1	Name	IstItem
Command1	Name Caption	cmdTambah &Tambah
Command2	Name Caption	cmdPerbaiki &Perbaiki
Command3	Name Caption	cmdHapus &Hapus
Command4	Name Caption	cmdSimpankeFile &Simpan ke file
Command5	Name Caption	cmdSimpan &Simpan

Koding yang akan dilakukan :

Membuat deklarasi type dan variabel pada bagian General Declaration

```
Option Explicit                'Setiap variabel harus dideklarasikan

'Deklarasi enumerated type
Private Enum Flag
    flNone = 0
    flAdd = 1                'Tambah data
    flEdit = 2              'Perbaiki data
End Enum

'Deklarasi variabel form level
Dim Aksi As Integer        'Digunakan untuk menandai aksi yang dibuat pemakai
```

Membaca dari file Negara.txt dan mengisikannya kedalam lstNegara, serta membuat setting awal variabel pada Event Form Load

Private Sub Form_Load()

```
Dim sItem As String

If Dir("Negara.Txt") <> "" Then          'Jika file Negara.Txt telah
ada

    Open "Negara.txt" For Input As #1    'Buka text file untuk input

    Do While Not EOF(1)                 'Looping s/d Eof
        Input #1, sItem                 'baca 1 item
        LstNegara.AddItem (sItem)      'tambahkan ke Listbox
    Loop

    Close #1                            'Tutup text file

End If

Aksi = flNone                           'Mula-mula pemakai belum
melakukan aksi
```

End Sub

Melakukan koding untuk menangani event klik pada kontrol cmdTambah, adapun hal yang perlu dilakukan adalah menandai Aksi sebagai flAdd dan memindahkan fokus ke txtItem.

Private Sub cmdTambah_Click()

```
Aksi = flAdd                            'Aksi adalah flAdd
txtItem.SetFocus                        'pindah fokus ke txtItem
```

End Sub

Melakukan koding untuk menangani event klik pada kontrol cmdPerbaiki, adapun hal yang perlu dilakukan adalah memeriksa apakah pemakai ada memilih item yang diperbaiki, kemudian memasukkan item yang diperbaiki ke txtItem, kemudian menandai Aksi sebagai flEdit, kemudian memindahkan fokus ke txtItem.

Private Sub cmdPerbaiki_Click()

```
If LstNegara.Text = "" Then

    MsgBox "Anda harus memilih Item" & vbCrLf & _
        "yang akan diperbaiki", vbOKOnly + vbInformation

Else

    txtItem = LstNegara.Text
    txtItem.SetFocus
    Aksi = flEdit

End If
```

End Sub

Melakukan koding untuk menangani event klik pada kontrol cmdHapus, adapau hal yang perlu dilakukan adalah memeriksa Aksi, apakah flAdd atau flEdit, jika flAdd, maka tulisan dalam txtItem ditambahkan ke listbox, kalau flEdit maka item pada listbox dihapus, dan tulisan pada txtItem ditambahkan, kemudian Aksi di set ke flNone dan isi txtItem dikosongkan

```

Private Sub cmdSimpan_Click()
Dim Posisi As Integer

If Aksi = flAdd Then           'Jika Aksi adalah flAdd

    LstNegara.AddItem (txtItem.Text) 'Tambahkan txtItem ke Listbox

ElseIf Aksi = flEdit Then     'Jika Aksi adalah flEdit

    Posisi = LstNegara.ListIndex 'Ambil posisi aktif
    LstNegara.RemoveItem (Posisi) 'Hapus item posisi tersebut
    LstNegara.AddItem (txtItem.Text) 'Tambahkan txtItem ke Listbox

End If

Aksi = flNone                 'Aksi diset ke flNone
txtItem.Text = ""            'kosongkan txtItem

End Sub

```

Melakukan koding untuk menangani event klik pada cmdHapus, dengan memeriksa apakah ada item yang dipilih, kalau ada remove item tersebut,dan reset flag Aksi

```

Private Sub cmdHapus_Click()

If LstNegara.ListIndex < 0 Then 'Jika tidak ada yang dipilih

    MsgBox "Anda harus memilih Item" & vbCrLf & _
        "yang akan dihapus", vbOKOnly + vbInformation

Else 'Hapus item tersebut

    LstNegara.RemoveItem LstNegara.ListIndex

End If

Aksi = flNone

End Sub

```

Melakukan koding untuk menangani event klik pada cmdSimpankeFile, dengan looping dari item 0 s/d Listcount -1.

```

Private Sub cmdSimpanKeFile_Click()
Dim i As Integer

Open "Negara.txt" For Output As #1 'Buka file untuk ditulis

```

```

For i = 0 To LstNegara.ListCount - 1      'Dari item 0 s/d Listcount - 1
    Print #1, LstNegara.List(i)          'tuliskan ke textfile
Next i

Close #1                                  'tutup text file

MsgBox "Proses simpan selesai", vbOKOnly + vbInformation
    
```

End Sub

2. Buatlah program yang menyimpan nama-nama propinsi didalam sebuah text file Indonesia.txt, dimana setiap kali form diaktifkan, maka isi file Indonesia.txt akan dibaca dan ditampilkan dalam suatu listbox, kemudian pemakai dapat melakukan tambah, perbaiki maupun hapus item dalam listbox, dan isi daftar akan disimpan kembali ke file Indonesia.txt setiap pemakai melakukan klik pada Simpan ke file.



Latihan 11

Judul : Memahami kontrol ComboBox yang diaplikasikan untuk pengolahan data Peserta Indoprog-vb

1. Buatlah program untuk menyimpan data peserta indoprog-vb ke sebuah random file Peserta.Dat yang memiliki struktur sebagai berikut :

```
'Struktur data untuk random
file
Private Type Peserta
Email As String * 30
nama As String * 30
Alamat As String * 40
Kota As String * 20
Propinsi As String * 20
Negara As String * 20
Tempat As String * 20
TglLahir As Date
Pria As Integer
End Type
```

Pada masing-masing textbox dibatasi panjang pengetikkan terbatas sesuai dengan ukuran masing-masing field, seperti tampilan berikut :

Isi ComboBox cboNegara diisi dari file Negara.txt, dan Propinsi dari file Propinsi.txt, sehingga menghasilkan tampilan sebagai berikut :

Petunjuk, lakukan hal tersebut pada Event Form_Load, adapun potongan programnya adalah sebagai berikut :

Private Sub Form_Load()

```

Dim Item As String 'Variabel untuk membaca isi text
file

'Isi cboNegara dengan file Negara.txt
Open "Negara.txt" For Input As #1 'Buka text file Negara.txt
Do While Not EOF(1)
    Input #1, Item 'Baca ke item
    cboNegara.AddItem (Item) 'Tambahkan item ke cboNegara
Loop
Close #1 'Tutup text file
'Isi cboPropinsi dengan file Propinsi.txt
Open "Propinsi.txt" For Input As #1
Do While Not EOF(1)
    Input #1, Item
    cboPropinsi.AddItem (Item)
Loop
Close #1
'Buka Random file Peserta.Dat
Open "Peserta.Dat" For Random As #1 Len=Len(DatPeserta)
JlhRec = LOF(1) / Len(DatPeserta)
If JlhRec > 0 Then
    Posisi = 1
Else
    Posisi = 0
End If
    
```

Call Tampil

'Jalankan Sub Tampil

End Sub

Untuk koding pada event-event tombol harus anda buat sendiri dengan mempelajari dan mencontoh Latihan pada Modul-modul sebelumnya untuk kemajuan anda sendiri.

Catatan : Fasilitas Find adalah mencari berdasarkan Nama

Latihan 12

Judul : Tantangan penyempurnaan Program Peserta Indoprogram-vb

1. Buka kembali program Latihan1 diatas, jalankan program diatas, dan carilah kelemahan-kelemahan yang ada, serta lengkapi program diatas untuk menutupi lubang kelemahan tersebut.
2. Lengkapi program Latihan2 dengan kemampuan memeriksa apakah data yang dimasukkan telah ada pada random file atau tidak, adapun kunci yang digunakan adalah alamat Email pada saat penyimpanan.
3. Lengkapi program Latihan2 dengan kemampuan Find Next, dimana mencari data memenuhi syarat yang berikutnya.

The screenshot shows a Windows application window titled "Peserta Belajar Bersama Indoprogram-vb". The window contains a form with the following fields and values:

- Alamat Email: hendra@indoprogram.com
- Nama: Hendra Wijaya
- Alamat: Jl. Sudirman No. 22/10
- Kota: Medan
- Negara: Indonesia (dropdown menu)
- Propinsi: Sumatera Utara (dropdown menu)
- Tempat/Tgl Lahir: Medan, 6/19/01
- Kelamin: Pria


At the bottom of the form, there are several buttons: "First", "Prev", "Next", "Last", "Add", "Edit", "Update", "Find", and "Find Next".

4. Buatlah semua form-form diatas ditampilkan ditengah layar pada saat di jalankan.

Modul 7, Kontrol Standard (bagian 3)

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	Fungsi
BorderStyle	<p>Digunakan untuk menentukan jenis border yang digunakan (0 - None, 1 - Fixed Single)</p> 
DragIcon	Digunakan untuk menentukan icon yang digunakan pada operasi drag-n-drop pada kontrol label
DragMode	Menentukan modus drag-n-drop (0 - manual, 1 - automatic)
Enabled	Menentukan apakah kontrol dapat menerima event yang dibuat oleh pemakai, misalnya event Click, MouseDown, MouseUp, dll.
Height	Digunakan untuk menentukan tinggi dari kontrol
Index	Digunakan untuk menentukan nomor index, jika kontrol tersebut merupakan kontrol array.
Left	Digunakan untuk menentukan jarak kiri kontrol dari kontainernya.
MouseIcon	Digunakan untuk menentukan icon mouse yang digunakan ketika pointer mouse berada diatas kontrol label tersebut, properti ini berarti bila properti MousePointer diset menjadi (99 - Custom)
MousePointer	Digunakan untuk menentukan bentuk pointer yang digunakan ketika mouse pointer berada diatas kontrol label tersebut.
Stretch	Digunakan untuk menentukan apakah gambar disesuaikan dengan ukuran kontrol (gambar dapat mengalami pengecilan maupun pembesaran)

	
Picture	Digunakan untuk menentukan gambar dalam image, atau mengambil gambar dalam image.
Tag	Properti ini dapat digunakan sebagai tempat menyimpan data sementara yang berkaitan dengan kontrol picture tersebut
ToolTipText	Digunakan untuk menentukan tip tulisan ketika mouse pointer berada diatas kontrol
Top	Digunakan untuk menentukan jarak atas kontrol dari kontainernya.
Visible	Digunakan untuk menentukan apakah kontrol label tersebut Visible atau tidak (False, True)
Width	Digunakan untuk menentukan lebar kontrol label

Menampilkan gambar kedalam kontrol image

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

```
MyImage.Picture = LoadPicture("C:\Grafik\Indoprogram.gif")
```

Catatan :

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

Mengambil gambar dari Clipboard ke kontrol image

Anda dapat menggunakan metoda GetData(option) dari object Clipboard untuk mengambil isi Clipboard ke suatu kontrol image, contoh :

```
MyImage.Picture = Clipboard.GetData
```

Catatan :

Metoda GetData Mengembalikan suatu grafik dari objek Clipboard, format adalah suatu konstanta optional yang menentukan format grafik. Jika format adalah 0 (no) atau diabaikan, GetData secara otomatis.

argumen format Optional. Suatu konstanta atau nilai yang menunjukkan format grafik pada clipboard, Jika format 0 (no) atau diabaikan, GetData otomatis menggunakan format yang sesuai. Adapun nilai konstanta untuk argumen format Optional adalah sebagai berikut :

Konstanta	Nilai	Keterangan
vbCFBitmap	2	Bitmap (.bmp)
vbCFMetafile	3	Metafile (.wmf)
vbCFDIB	8	Device-independent bitmap (DIB)
vbCFPalette	9	Color palette

Mengosongkan kontrol image

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

```
MyImage.Picture = LoadPicture
```

Anda dapat juga menggunakan fungsi LoadPicture tanpa argumen untuk memeriksa apakah kontrol image dalam keadaan kosong atau tidak.

```
If MyImage.Picture = LoadPicture Then
    MsgBox "Gambar harus diisi"
Endif
```

Menyimpan gambar dalam kontrol image ke file

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

```
SavePicture MyImage.Picture, "C:\Grafik\Indoprogram.bmp"
```

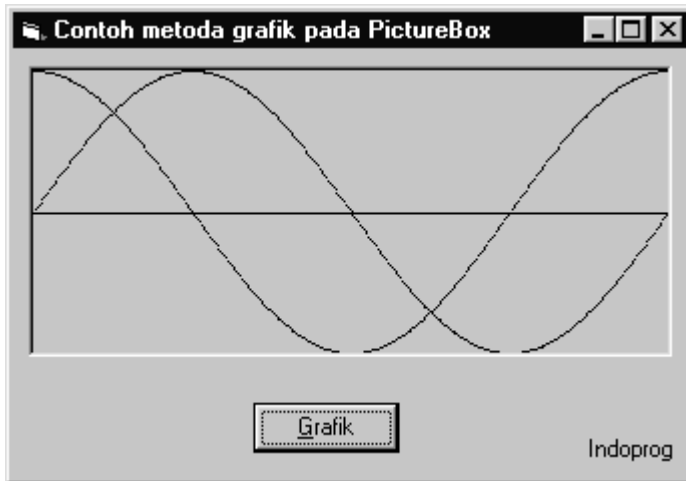
Catatan :

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

PictureBox

Kalau image digunakan untuk menampilkan gambar, demikian juga picturebox. Selain menampilkan gambar picture box mendukung berbagai metoda untuk operasi grafik, dan dapat berfungsi sebagai kontainer bagi kontrol-kontrol lain.

Contoh :

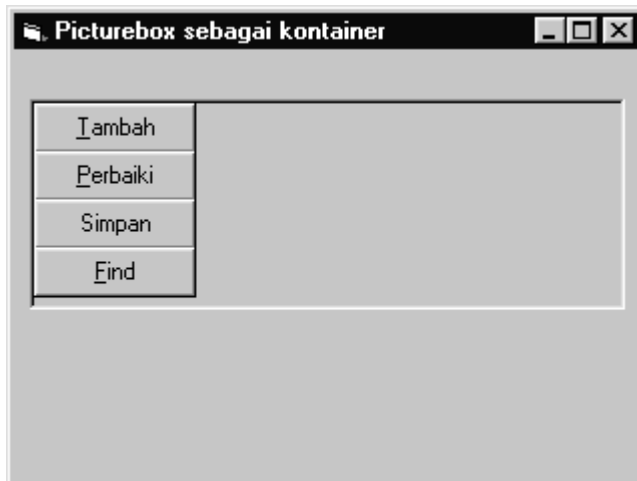


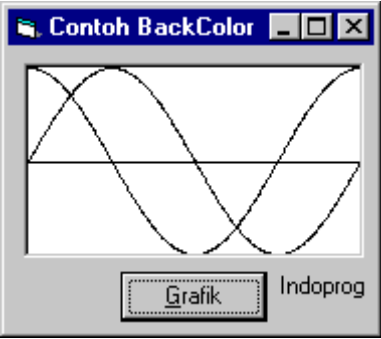
Dihasilkan oleh pemakaian metoda Line, dan Pset pada PictureBox

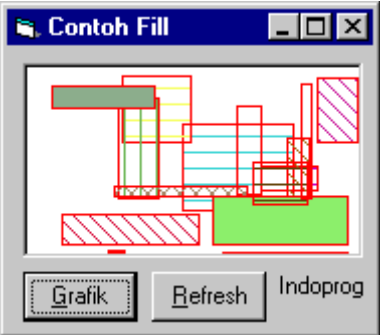
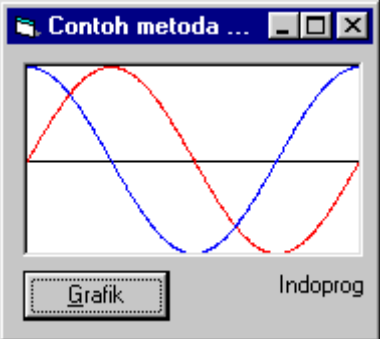
```


Private Sub cmdGrafik_Click()
MyPicture.ScaleMode = 0
MyPicture.ScaleWidth = 360
MyPicture.ScaleHeight = 2
MyPicture.ScaleLeft = 0
MyPicture.ScaleTop = -1
MyPicture.Line (0, 0)-(360, 0)
For i = 0 To 360
    MyPicture.PSet (i, -Sin(i * 3.14 / 180))
    MyPicture.PSet (i, -Cos(i * 3.14 / 180))
Next i
End Sub
    
```

Berikut ini adalah picturebox sebagai kontainer.



Properti	Fungsi
Align	Digunakan untuk menentukan bagaimana pemerataan objek pada form (0 - None, 1 - Align Top, 2 - Align Bottom, 3 - Align Left, 4 - Align Right)
Appearance	sda
BackColor	Digunakan untuk menentukan warna latar belakang. 
AutoRedraw	Digunakan untuk menentukan apakah grafik yang dihasilkan dengan metoda penggambaran (Line, Circle, PSet) akan dijadikan sebagai bitmap yang tetap.
AutoSize	Digunakan untuk menentukan apakah ukuran kontrol otomatis disesuaikan dengan ukuran objek picture.
BorderStyle	sda
CauseValidation	Digunakan untuk menentukan apakah event Validate di aktifkan ketika kontrol akan kehilangan fokus (False, True)
ClipControl	Digunakan untuk menentukan icon yang digunakan pada operasi drag-n-drop pada kontrol label
DragIcon	Digunakan untuk menentukan icon yang digunakan pada operasi drag-n-drop pada kontrol label
DragMode	Menentukan modus drag-n-drop (0 - manual, 1 - automatic)
DrawMode	Menentukan jenis modus yang digunakan pada saat penggambaran grafik dalam picture dimana merupakan interaksi antara warna latar belakang dengan warna garis sehingga menghasilkan warna baru.
DrawStyle	Menentukan style garis yang digunakan dalam menggambar garis.
DrawWidth	Menentukan ukuran garis penggambaran
Enabled	Menentukan apakah kontrol label dapat menerima event yang dibuat oleh pemakai, misalnya event Click, MouseDown, MouseUp, dll.
FillColor	Menentukan warna pengisian grafik

<p>FillStyle</p>	<p>Menentukan pola pengisian grafik</p> 
<p>Font</p>	<p>Digunakan untuk menentukan jenis font, ukuran, style, dll</p>
<p>ForeColor</p>	<p>Digunakan untuk menentukan warna tulisan</p>  <p>Contoh :</p> <pre> Private Sub cmdGrafik_Click() MyPicture.ScaleMode = 0 MyPicture.ScaleWidth = 360 MyPicture.ScaleHeight = 2 MyPicture.ScaleLeft = 0 MyPicture.ScaleTop = -1 MyPicture.ForeColor = vbBlack 'Set ForeColor jadi hitam MyPicture.Line (0, 0)-(360, 0) 'Garis hitam For i = 0 To 360 MyPicture.ForeColor = vbRed MyPicture.PSet (i, -Sin(i * 3.14 / 180)) 'Titik Merah MyPicture.ForeColor = vbBlue MyPicture.PSet (i, -Cos(i * 3.14 / 180)) 'Titik Biru Next i End Sub </pre>
<p>FontTransparant</p>	<p>Digunakan untuk menentukan apakah latar belakang tulisan transparant atau tidak.</p>

	 <pre data-bbox="480 583 1036 758"> Private Sub MyCommand_Click() MyPicture.FontTransparent = True MyPicture.Print "Transparent" MyPicture.FontTransparent = False MyPicture.Print "Tidak Transparent" End Sub </pre>
Height	Digunakan untuk menentukan tinggi dari kontrol
Index	Digunakan untuk menentukan nomor index, jika kontrol tersebut merupakan kontrol array.
Left	Digunakan untuk menentukan jarak kiri kontrol dari kontainernya.
MouseIcon	Digunakan untuk menentukan icon mouse yang digunakan ketika pointer mouse berada diatas kontrol label tersebut, properti ini berarti bila properti MousePointer diset menjadi (99 - Custom)
MousePointer	Digunakan untuk menentukan bentuk pointer yang digunakan ketika mouse pointer berada diatas kontrol label tersebut.
Picture	Digunakan untuk menentukan gambar dalam picture box.
ScaleMode	<p>Digunakan untuk menentukan unit skala yang digunakan (0 - User, 1 - Twip, 2 - Point, 3 - Pixel, 4 - Character, 5 - Inch, 6 - Milimeter, 7 - Centimeter)</p> <p>1 inchi = 1440 Twip 1 cm = 567 Twip</p> <p>1 inchi = 72 point</p> <p>1 character = (120 Twip untuk lebar, 240 untuk tinggi)</p> <p>1 cm = 1000 unit (Himetric)</p> <p>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.</p>
ScaleLeft	Digunakan untuk menentukan nilai koodinat horizontal paling kiri, anda dapat menggunakan properti ini untuk menentukan koordinat paling kiri dari suatu sumbu X.
ScaleTop	Digunakan untuk menentukan nilai koodinat vertikal paling atas, anda dapat menggunakan properti ini untuk menentukan koordinat paling atas dari suatu sumbu Y.
ScaleHeight	Digunakan untuk menentukan tinggi sumbu vertikal. Menentukan panjang sumbu Y.
ScaleWidth	Digunakan untuk menentukan panjang sumbu horizontal. Menentukan panjang sumbu X.
Tag	Properti ini dapat digunakan sebagai tempat menyimpan data sementara yang berkaitan dengan kontrol picture tersebut
ToolTipText	Digunakan untuk menentukan tip tulisan ketika mouse pointer berada diatas kontrol
Top	Digunakan untuk menentukan jarak atas kontrol dari kontainernya.
Visible	Digunakan untuk menentukan apakah kontrol label tersebut Visible atau tidak (False, True)
Width	Digunakan untuk menentukan lebar picturebox

Properti	Fungsi
CurrentX	Digunakan untuk mendapatkan/menentukan posisi X yang sedang aktif, digunakan untuk perintah Print
CurrentY	Digunakan untuk mendapatkan/menentukan posisi Y yang sedang aktif, digunakan untuk perintah Print

Pengolahan gambar dalam PictureBox dapat dicontoh dari contoh pada Image diatas.

Metoda Grafik pada PictureBox

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

Metoda	Keterangan
Circle(x,y),r,warna,awal,akhir,Aspek	Mengambarkan 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 tinggi dengan lebar.
Cls	Membersihkan PictureBox dengan warna BackColor
Line (x1,y1) - (x2, y2),warna	Mengambarkan garis tunggal dari koordinat x1,y1 sampai dengan x2,y2
Line (x1,y1) - (x2, y2),warna,B	Mengambarkan 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
PSet Step (x,y)	Mencetak dot (titik) pada koordinat relatif terhadap posisi dot sebelumnya.
Point(x,y)	Mendapatkan warna dot (titik) tertentu pada layar Contoh : Color& = MyPicture.Point(100,200)
Refresh	Menyebabkan kontrol picture dan bitmap tetapnya digambar ulang.

Event Paint Pada PictureBox

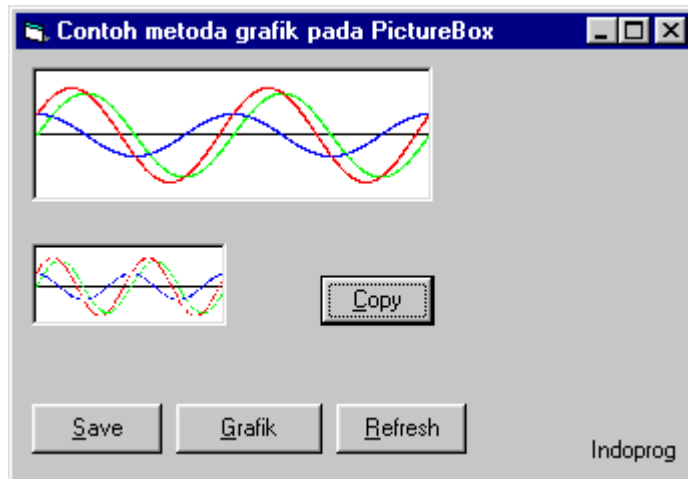
Event Paint akan diaktifkan ketika PictureBox mengalami penggambaran. Anda dapat menggunakan event ini untuk menjalankan fungsi-fungsi tertentu ketika PictureBox digambar.

Metoda PaintPicture Pada PictureBox

Metoda PaintPicture merupakan metoda yang sangat bermanfaat untuk melakukan operasi seperti pembesaran, pengecilan, duplikasi sebagian gambar berdasarkan koordinat tertentu. Adapun syntax penulisannya adalah sebagai berikut :

```

PictureBoxTujuan.PaintPicture PictureBoxAsal, posisiXtujuan,
posisiYtujuan, [lebarTujuan], _
    [tinggiTujuan], [koordinatXAsal], [KoordinatYAsal], [lebarAsal],
[tinggiAsal]
    
```



```
Private Sub cmdCopy_Click()  
Tujuan.PaintPicture MyPicture.Image, _  
    0, 0, Tujuan.Width, Tujuan.Height, _  
    0, 0, MyPicture.Width, MyPicture.Height  
End Sub
```

Menyimpan Grafik dalam PictureBox ke file

Untuk menyimpan ataupun mengambil grafik yang terdapat dalam suatu picturebox, baik yang di muat dari suatu file gambar, maupun yang dibuat dengan menggunakan metoda penggambaran, Anda dapat menggunakan properti image.

Contoh :

```
SavePicture MyPicture.Image, "C:\Grafik.bmp"
```

Latihan 12

Judul : Memanfaatkan kontrol image untuk menampilkan Foto peserta

Pada Latihan ini kita akan meningkatkan fasilitas project Modul 4 Latihan 1 sebelumnya dengan kemampuan menampilkan foto dan menyimpan foto peserta. Adapun foto dari masing-masing peserta akan disimpan pada file-file eksternal yang diberi nama Fotox.bmp, dimana x disesuaikan dengan nomor record data dalam random file. Misalnya pada record 1, maka nama file fotonya adalah Foto1.bmp, record 2 nama file fotonya adalah Foto2.bmp, dst.

1. Bukalah project Modul 4 Latihan 1, dan tambahkan kontrol image serta command sehingga menjadi sebagai berikut :



Kontrol	Properti	Value
image1	name stretch	imgFoto True
command1	name caption	cmdPaste &Paste

2. Pada bagian general declaration tambahkan deklarasi variabel cFileFoto yang akan digunakan untuk menyimpan nama file foto. Adapun penamaan file foto adalah Fotox.bmp, dimana x disesuaikan dengan nomor record data dalam random file. Misalnya

pada record 1, maka nama file fotonya adalah Foto1.bmp, record 2 nama file fotonya adalah Foto2.bmp, dst.

Dim cFileFoto As String

Sehingga menjadi seperti berikut :

```
'Deklarasi variabel Form Level
Dim DatPeserta As Peserta
Dim Posisi As Integer
Dim JlhRec As Integer
Dim Aksi As Integer
Dim cFileFoto As String           'Ini yang ditambahkan
```

3. Lakukan koding untuk event klik pada cmdPaste yang berfungsi untuk mengisi kontrol imgFoto dengan isi dari Clipboard.

```
Private Sub cmdPaste_Click()
imgFoto.Picture = Clipboard.GetData           'Paste data dari Clipboard ke
kontrol image
End Sub
```

4. Tambahkan koding pada bagian cmdUpdate klik dimana berfungsi menyimpan gambar dalam imgFoto ke media penyimpanan dengan nama Fotox.bmp, dimana x disesuaikan dengan nomor posisi record penulisan. Misalnya pada record 1, maka nama file fotonya adalah Foto1.bmp, record 2 nama file fotonya adalah Foto2.bmp, dst.

```
cFileFoto = "Foto" & Posisi & ".bmp"       'Buat nama file Fotox.bmp
If imgFoto.Picture <> LoadPicture Then     'Periksa apakah kontrol image
kosong
    SavePicture imgFoto.Picture, cFileFoto 'Simpan gambar ke file dengan
nama yang
End If                                     'telah dibuat
```

Sehingga menjadi sebagai berikut :

```
Private Sub cmdUpdate_Click()
If Aksi = flAdd Or Aksi = flEdit Then
    If Trim$(txtEmail.Text) = "" Then
        MsgBox "Alamat Email tidak bisa kosong"
        txtEmail.SetFocus
        Exit Sub
    End If
    If Trim$(txtNama.Text) = "" Then
        MsgBox "Nama tidak bisa kosong"
        txtNama.SetFocus
        Exit Sub
    End If
    If Aksi = flAdd Then
        JlhRec = JlhRec + 1
        Posisi = JlhRec
    End If
```

```

DatPeserta.Email = txtEmail.Text
DatPeserta.nama = txtNama.Text
DatPeserta.Aalamat = txtAlamat.Text
DatPeserta.Kota = txtKota.Text
DatPeserta.Negara = cboNegara.Text
DatPeserta.Propinsi = cboPropinsi.Text
DatPeserta.Tempat = txtTempat.Text
DatPeserta.TglLahir = txtTglLahir.Text
DatPeserta.Pria = chkPria.Value
Put #1, Posisi, DatPeserta
cFileFoto = "Foto" & Posisi & ".bmp"      'Bagian yang ditambahkan
If imgFoto.Picture <> LoadPicture Then
    SavePicture imgFoto.Picture, cFileFoto
End If
Call Kunci
End If
End Sub

```

5. Pada Sub Tampil tambahkan koding yang berfungsi membaca file foto dari disk dan menampilkan ke kontrol image. Adapun proses yang dilakukan adalah membentuk nama file foto berdasarkan posisi record yang ditampilkan, kemudian memeriksa keberadaan file foto tersebut dalam media penyimpanan, jika ada, maka baca dan tampilkan pada kontrol imgFoto.

```

cFileFoto = "Foto" & Posisi & ".bmp"      'Bentuk nama file yaitu
Fotox.bmp
If Dir(cFileFoto) <> "" Then              'Periksa keberadaan file
tersebut
    imgFoto.Picture = LoadPicture(cFileFoto) 'Baca dan tampilkan
Else
    imgFoto.Picture = LoadPicture
End If

```

Sehingga menjadi sebagai berikut :

Sub Tampil()

```

If Posisi > 0 Then
    Get #1, Posisi, DatPeserta
    txtEmail.Text = DatPeserta.Email
    txtNama.Text = DatPeserta.nama
    txtAlamat.Text = DatPeserta.Aalamat
    txtKota.Text = DatPeserta.Kota
    cboNegara.Text = DatPeserta.Negara
    cboPropinsi.Text = DatPeserta.Propinsi
    txtTempat.Text = DatPeserta.Tempat
    txtTglLahir.Text = DatPeserta.TglLahir
    chkPria.Value = DatPeserta.Pria
    cFileFoto = "Foto" & Posisi & ".bmp"      'Ini yang ditambahkan
    If Dir(cFileFoto) <> "" Then
        imgFoto.Picture = LoadPicture(cFileFoto)
    Else
        imgFoto.Picture = LoadPicture
    End If
    Call Kosong

```

```
End If  
Aksi = flNone  
Call Kunci  
End Sub
```

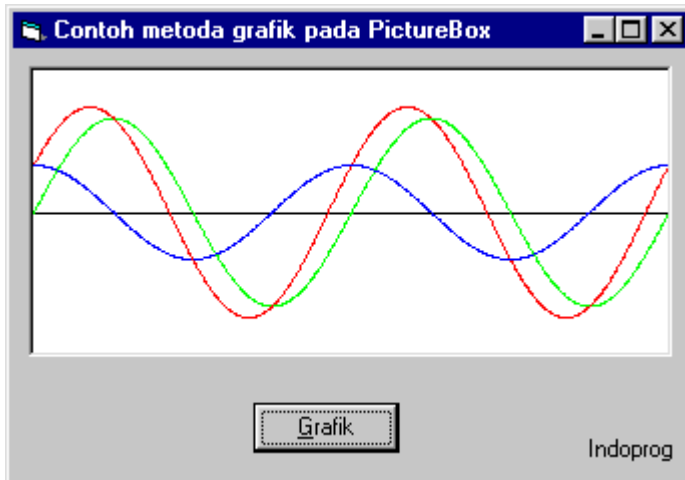
6. Jalankan software paintbrush, dan buka suatu file gambar, select suatu lokasi tertentu, kemudian klik Edit Copy.

7. Jalankan project diatas, dan klik pada Edit, kemudian klik pada Paste, dan Klik Update. Tutup program dan jalankan kembali, apakah foto masih ada disana ?

Latihan 13

Judul : Memahami properti dan metoda serta event pada pictureBox

1. Buatlah grafik $2 \sin x$ (warna hijau), $\cos x$ (warna biru), dan $2 \sin x + \cos x$ (warna merah), dengan sumbu Y dari -3 s/d 3, dan Sumbu X dari 0 sampai 720 derajat.



Solusi :

1. Atur ScaleMode PictureBox ke 0 - User

```
MyPicture.ScaleMode = 0
```

2. Atur ScaleWidth (panjang Sumbu X) PictureBox ke 720 (0 s/d 720 derajat)

```
MyPicture.ScaleWidth = 720
```

3. Atur ScaleHeight (panjang Sumbu Y) PictureBox ke 6 (-3 s/d 3)

```
MyPicture.ScaleHeight = 6
```

4. Atur ScaleLeft (koordinat paling kiri Sumbu X) PictureBox ke 0

```
MyPicture.ScaleLeft = 0
```

5. Atur ScaleTop (koordinat paling atas Sumbu Y) PictureBox ke -3 (Harus disadari bahwa perbedaan antara koordinat sumbu Y dimatematika dengan koordinat sumbu di PictureBox)

```
MyPicture.ScaleTop = -3
```

6. Gambarkan sumbu Y, dengan metode Line, mulai dari koordinat 0,0 s/d 720,0

```
MyPicture.Line (0, 0)-(720, 0)
```

7. Lakukan looping mulai dari 0 s/d 720 untuk menggambar titik-titik $2 \sin x$, $\cos x$, dan $2 \sin x + \cos x$.

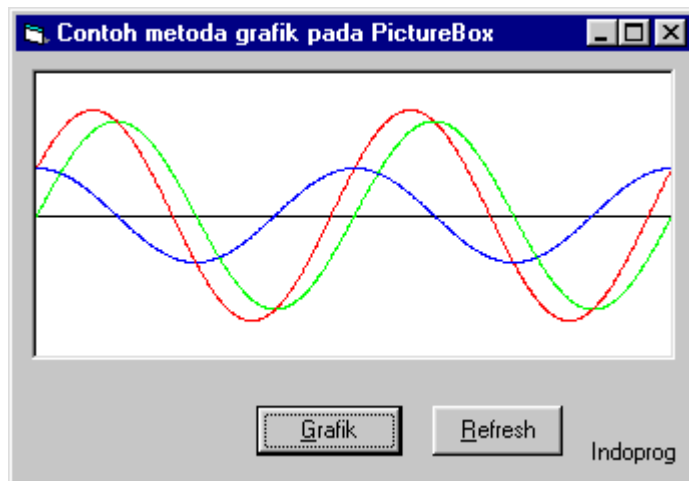
```
For i = 0 To 720
    MyPicture.ForeColor = vbGreen           'Warna Hijau
    MyPicture.PSet (i, -2 * Sin(i * 3.14 / 180))
    MyPicture.ForeColor = vbBlue           'Warna Biru
    MyPicture.PSet (i, -Cos(i * 3.14 / 180))
    MyPicture.ForeColor = vbRed           'Warna Merah
    MyPicture.PSet (i, -2 * Sin(i * 3.14 / 180) - Cos(i * 3.14 / 180))
Next i
```

Sehingga menjadi seperti berikut ini :

```
Private Sub cmdGrafik_Click()
MyPicture.ScaleMode = 0
MyPicture.ScaleWidth = 720
MyPicture.ScaleHeight = 6
MyPicture.ScaleLeft = 0
MyPicture.ScaleTop = -3
MyPicture.Line (0, 0)-(720, 0)
For i = 0 To 720
    MyPicture.ForeColor = vbGreen
    MyPicture.PSet (i, -2 * Sin(i * 3.14 / 180))
    MyPicture.ForeColor = vbBlue
    MyPicture.PSet (i, -Cos(i * 3.14 / 180))
    MyPicture.ForeColor = vbRed
    MyPicture.PSet (i, -2 * Sin(i * 3.14 / 180) - Cos(i * 3.14 / 180))
Next i
End Sub
```

2. Jalankan program dan klik pada command Grafik, dan perhatikan hasilnya.

3. **(Memahami metoda Refresh)** Tambahkan sebuah tombol refresh, dan lakukan koding sebagai berikut :



```
Private Sub cmdRefresh_Click()
MyPicture.Refresh
End Sub
```

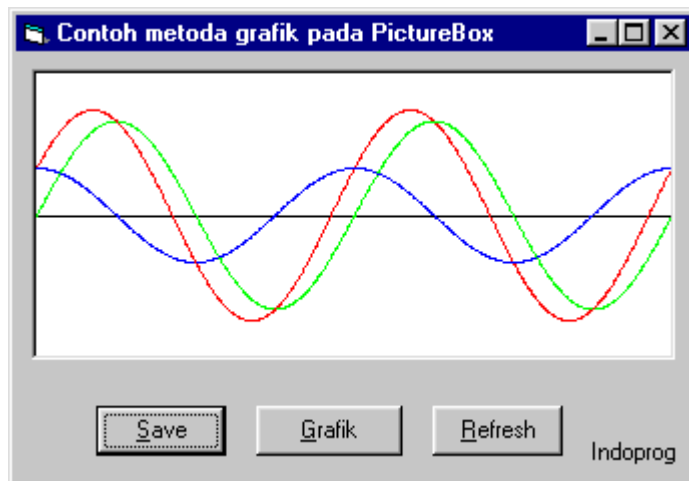
4. Jalankan kembali program, dan klik pada command Grafik, dan klik pada command Refresh, dan perhatikan hal yang terjadi.

5. **(Memahami properti AutoRedraw)** Lakukan perbaikan terhadap koding cmdGrafik sehingga menjadi sebagai berikut :

```
Private Sub cmdGrafik_Click()
MyPicture.ScaleMode = 0
MyPicture.ScaleWidth = 720
MyPicture.ScaleHeight = 6
MyPicture.ScaleLeft = 0
MyPicture.ScaleTop = -3
MyPicture.AutoRedraw = True 'Ini yang ditambahkan
MyPicture.Line (0, 0)-(720, 0)
For i = 0 To 720
    MyPicture.ForeColor = vbGreen
    MyPicture.PSet (i, -2 * Sin(i * 3.14 / 180))
    MyPicture.ForeColor = vbBlue
    MyPicture.PSet (i, -Cos(i * 3.14 / 180))
    MyPicture.ForeColor = vbRed
    MyPicture.PSet (i, -2 * Sin(i * 3.14 / 180) - Cos(i * 3.14 / 180))
Next i
MyPicture.AutoRedraw = False 'Ini yang ditambahkan
End Sub
```

6. Jalankan kembali program, dan klik pada command Grafik, dan klik pada command Refresh, dan perhatikan hal yang terjadi. Apa perbedaannya dengan yang sebelumnya.

7. **(Memanfaatkan properti image)** Tambahkan sebuah command Save pada project anda, dan lakukan koding jika tombol tersebut di klik, maka akan menyimpan grafik dalam pictureBox ke file "Grafik.bmp"



```
Private Sub cmdSave_Click()
SavePicture MyPicture.Image, "Grafik.bmp"
End Sub
```

8. **(Memahami event Paint)** Tanamkan pada Form anda suatu PictureBox berukuran 1550 Twips x 1550 Twips, kemudian lakukan koding berikut pada Event Paint, dan jalankan project tersebut, dan perhatikan picturebox anda.

```
Private Sub MyPicture_Paint ()
MyPic.Circle (775, 775), 500
MyPic.Circle (550, 650), 50
MyPic.Circle (1000, 650), 50
MyPic.Circle (775, 700), 400, , 4, 5.6
End Sub
```



Latihan 14

Judul : Tantangan membuat Mari Menggambar

Buatlah program mari yang mampu menggambar bentuk Lingkaran, Persegi, Garis, Titik, dan dapat menyimpan hasil ke suatu file bmp.



Berikut ini adalah contoh solusi untuk menggambar Lingkaran, yang lainnya adalah tugas anda untuk melengkapinya.

Private Enum Gambar

```
None = 0
Lingkar = 1
Persegi = 2
Garis = 3
Titik = 4
```

End Enum

```
Dim Aksi As Gambar 'digunakan untuk mengenali aksi pemakai
Dim Tahap As Integer 'digunakan untuk menentukan tahap
penggambaran
```

```
Dim Pusatx As Integer 'digunakan untuk Pusat Lingkaran
Dim Pusaty As Integer
```

Private Sub cmdCircle_Click()

```
Aksi = Lingkar 'tanda aksi sebagai menggambar lingkaran
Tahap = 1 'mulai dari tahap penentuan pusat
lingkaran
```

End Sub

Private Sub PicKanvas_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)



```
If Aksi = Lingkar Then 'jika aksi gambar lingkaran
    If Tahap = 1 Then 'jika tahap 1
        Pusatx = X 'simpan pusat lingkaran
```

```
        Pusaty = Y
        Tahap = 2           'tandaai tahap berikutnya yaitu penentuan
jari-jari
    End If
End If
End Sub
Private Sub PicKanvas_MouseMove(Button As Integer, Shift As Integer, X
As Single, Y As Single)
If Aksi = Lingkaran Then
    If Tahap = 2 Then
        PicKanvas.Refresh
        jari = Sqr((Y - Pusaty) ^ 2 + (X - Pusatx) ^ 2)
        PicKanvas.Circle (Pusatx, Pusaty), jari
    End If
End If
End Sub
Private Sub PicKanvas_MouseUp(Button As Integer, Shift As Integer, X As
Single, Y As Single)
If Aksi = Lingkaran Then
    If Tahap = 2 Then
        PicKanvas.Refresh
        PicKanvas.AutoRedraw = True
        jari = Sqr((Y - Pusaty) ^ 2 + (X - Pusatx) ^ 2)
        PicKanvas.Circle (Pusatx, Pusaty), jari
        PicKanvas.AutoRedraw = False
        Aksi = None
    End If
End If
End Sub
```

Modul 7, Kontrol Standard (bagian 4)

HScrollBar dan VScrollBar

Horizontal ScrollBar dan Vertical ScrollBar digunakan untuk memungkinkan pemakai melakukan pemasukkan data secara analog, dengan melakukan pengeseran ataupun penekanan tombol (kiri, kanan, atas, bawah). Anda dapat membayangkan Scrollbar sebagai pengatur Volume pada Radio.

Properti	Fungsi
CauseValidation	Digunakan untuk menentukan apakah event Validate di aktifkan ketika kontrol akan kehilangan fokus (False, True)
DragMode	Menentukan modus drag-n-drop (0 - manual, 1 - automatic)
DragIcon	Digunakan untuk menentukan icon yang digunakan pada operasi drag-n-drop pada kontrol.
Enabled	Menentukan apakah kontrol label dapat menerima event yang dibuat oleh pemakai, misalnya event Click, MouseDown, MouseUp, dll.
Height	Digunakan untuk menentukan tinggi dari kontrol
Index	Digunakan untuk menentukan nomor index, jika kontrol tersebut merupakan kontrol array.
LargeChange	Digunakan untuk menentukan besar perubahan Value ketika pemakai melakukan klik di antara tanda Arah dan Bar, atau ketika pemakai melakukan penekanan tombol PgUp dan PgDn pada saat aktif di ScrollBar.
Left	Digunakan untuk menentukan jarak kiri kontrol dari kontainernya.
Max	Digunakan untuk menentukan nilai Maksimal dari kontrol ScrollBar, ketika Bar rapat ke kanan atau ketika pemakai melakukan penekanan tombol End pada saat aktif di ScrollBar. 
Min	Digunakan untuk menentukan nilai Minimal dari kontrol ScrollBar, ketika Bar rapat ke kiri atau ketika pemakai melakukan penekanan tombol Home pada saat aktif di ScrollBar. 
MouseIcon	Digunakan untuk menentukan icon mouse yang digunakan ketika pointer mouse berada diatas kontrol label tersebut, properti ini berarti bila properti MousePointer diset menjadi (99 - Custom)
MousePointer	Digunakan untuk menentukan bentuk pointer yang digunakan ketika mouse pointer berada diatas kontrol label tersebut.
SmallChange	Digunakan untuk menentukan besar perubahan Value ketika pemakai melakukan klik pada tanda Arah, atau ketika pemakai melakukan penekanan tombol panah kiri dan kanan pada saat aktif di ScrollBar.
TabIndex	Digunakan untuk menentukan urutan tabindex, 0 adalah kontrol yang pertama mendapat fokus jika form dijalankan. Properti ini efektif jika setting properti TabStop adalah True

TabStop	Digunakan untuk menentukan apakah textbox dapat menerima fokus karena penekanan tombol TAB oleh pemakai (False, True)
Tag	Properti ini dapat digunakan sebagai tempat menyimpan data sementara yang berkaitan dengan kontrol label tersebut
Top	Digunakan untuk menentukan jarak atas kontrol dari kontainernya.
Value	Digunakan untuk mendapatkan / menentukan nilai dari ScrollBar
Visible	Digunakan untuk menentukan apakah kontrol label tersebut Visible atau tidak (False, True)
Width	Digunakan untuk menentukan lebar kontrol.

Event Pada ScrollBar

Pada ScrollBar ada dua event yang perlu diperhatikan, yaitu Event Change dan Event Scroll, dimana :

Event Change, akan dibangkitkan ketika terjadi perubahan Value pada ScrollBar, tetapi event ini tidak dibangkitkan ketika pemakai masih melakukan pengeseran terhadap Bar, kecuali kalau pemakai telah melepaskan tombol kiri Mouse.

Contoh :


```






Private Sub VScroll_Change ()
  ImgGambar.Top = -VScroll.Value      'Berlawanan arah
End Sub
Private Sub HScroll_Change ()
  ImgGambar.Left = -HScroll.Value     'Berlawanan arah
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 Change ketika pemakai melepas penekanan tombol mouse.

Jadi anda harus memanfaatkan kedua event tersebut untuk mendapatkan hasil yang baik dari pemakaian ScrollBar.

Line dan Shape

Properti	Fungsi
BackColor	Menentukan warna latarbelakang dari suatu Shape. Properti ini efektif jika setting Backstyle dari Shape adalah (1 - Opaque) 
BackStyle	Menentukan jenis latarbelakang dari suatu Shape, apakah (0 - Transparent, 1 - Opaque)

<p>BorderColor</p>	<p>Menentukan warna bingkai dari suatu Shape. Properti ini efektif jika setting BorderStyle bukan (0 - Transparent)</p> 
<p>BorderStyle</p>	<p>Menentukan jenis garis border yang digunakan. (0 - Transparent, 1 - Solid, 2 - Dash, 3 - Dot, 4 - Dash Dot, 5 - Dash Dot Dot, dst)</p> 
<p>BorderWidth</p>	<p>Menentukan ukuran ketebalan garis border.</p> 
<p>DrawMode</p>	<p>Menentukan jenis Drawmode yang digunakan. defaultnya adalah (13 - Copy pen)</p> 
<p>FillColor</p>	<p>Menentukan warna yang digunakan untuk menggambarkan pola isinya yang ditentukan pada properti FillStyle.</p> 
<p>FillStyle</p>	<p>Menentukan jenis arsiran yang digunakan untuk isian kontrol Shape. Lihat gambar pada properti FillColor.</p>
<p>Height</p>	<p>Digunakan untuk menentukan tinggi dari kontrol</p>
<p>Index</p>	<p>Digunakan untuk menentukan nomor index, jika kontrol tersebut merupakan kontrol array.</p>
<p>Left</p>	<p>Digunakan untuk menentukan jarak kiri kontrol dari kontainernya.</p>
<p>Shape</p>	<p>Digunakan untuk menentukan bentuk Shape (0 - Rectangle, 1 - Square, 2 - Oval, 3 - Circle, 4 - Rounded Rectangle, 5 - Rounded Square)</p>
<p>Tag</p>	<p>Properti ini dapat digunakan sebagai tempat menyimpan data sementara yang berkaitan</p>

	dengan kontrol label tersebut
Top	Digunakan untuk menentukan jarak atas kontrol dari kontainernya.
Visible	Digunakan untuk menentukan apakah kontrol label tersebut Visible atau tidak (False, True)
Width	Digunakan untuk menentukan lebar kontrol.

Timer

Properti	Fungsi
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 Timer adalah event yang dibangkitkan oleh kontrol timer berdasarkan interval waktu yang telah ditentukan.

Contoh :

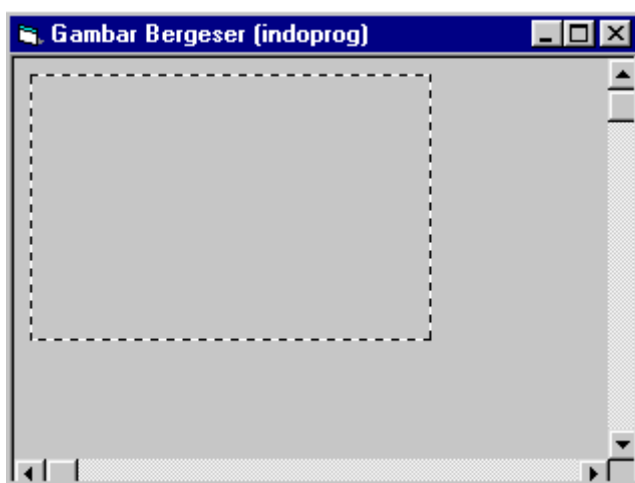
```

Private Sub Timer_Timer()
    Call AturJarum
End Sub
    
```

Latihan 15

Judul : Memanfaatkan HScrollBar dan VScrollBar untuk membuat gambar yang dapat bergeser.

Trik : Untuk membuat gambar yang dapat bergeser anda perlu menanamkan sebuah kontrol image, kontrol VscrollBar dan HScrollBar diatas sebuah kontrol Picture. Jadi dalam hal ini kontrol Picture bertindak sebagai kontainer.



Kontrol	Properti	Value
Picture1	Name	picBingkai
Image1	Name	imgGambar
HScroll1	Name	HScroll
VScroll1	Name	VScroll

Membuat Sub StatusBingkai yang berfungsi membuat ukuran PictureBox sama dengan ukuran Form sebelah dalam (ScaleWidth, ScaleHeight) dengan menggunakan Metoda Move. Dan membuat kontrol imgGambar rapat kiri atas terhadap kontainernya (0,0)

```
Private Sub StatusBingkai ()
PicBingkai.Move 0 , 0, Me.ScaleWidth , Me.ScaleHeight
ImgGambar.Move 0, 0
End Sub
```

Membuat Sub AturScrollBar yang berfungsi menentukan keaktifan tampilan dari VScroll dan HScroll. Adapun syarat penampilan dari kedua kontrol tersebut adalah :

HScroll, jika ukuran lebar gambar (imgGambar.Width) > ukuran tinggi bingkai (PicBingkai.Width)

VScroll, jika ukuran tinggi gambar (imgGambar.Height) > ukuran tinggi bingkai (PicBingkai.Height)

```

Private Sub AturScrollBar()
If ImgGambar.Width > PicBingkai.ScaleWidth Then      'Jika gambar lebih
lebar dari pada bingkai
    HScroll.Visible = True                            'aktifkan
horizontal scrollbar
Else
    HScroll.Visible = False
End If
If ImgGambar.Height > PicBingkai.ScaleHeight Then
    VScroll.Visible = True
Else
    VScroll.Visible = False
End If
End Sub

```

Membuat Sub StatusHScroll yang berfungsi mengatur penempatan HScroll pada picturebox (Left dan Top) dan ukuran dari HScroll (Width), serta Min, Max, SmallChange dan LargeChange.

```

Private Sub StatusHScroll()
If ImgGambar.Width > PicBingkai.ScaleWidth Then
    HScroll.Left = 0
    HScroll.Top = PicBingkai.ScaleHeight - HScroll.Height
    If ImgGambar.Height > PicBingkai.ScaleHeight Then
        HScroll.Width = PicBingkai.ScaleWidth - VScroll.Width
    Else
        HScroll.Width = PicBingkai.ScaleWidth
    End If
    HScroll.Min = 0
    HScroll.Max = (ImgGambar.Width - HScroll.Width) - 1
    HScroll.SmallChange = HScroll.Max \ 50
    HScroll.LargeChange = HScroll.Max \ 10
End If
End Sub

```

Membuat Sub StatusVScroll

```

Private Sub StatusVScroll()
If ImgGambar.Height > PicBingkai.ScaleHeight Then
    VScroll.Top = 0
    VScroll.Left = PicBingkai.ScaleWidth - VScroll.Width
    If ImgGambar.Width > PicBingkai.ScaleWidth Then
        VScroll.Height = PicBingkai.ScaleHeight - HScroll.Height
    Else
        VScroll.Height = PicBingkai.ScaleHeight
    End If
    VScroll.Min = 0
    VScroll.Max = (ImgGambar.Height - VScroll.Height) - 1
    VScroll.SmallChange = VScroll.Max \ 50
    VScroll.LargeChange = VScroll.Max \ 10
End If

```

End Sub

Melakukan Koding terhadap Event Change pada VScroll dan HScroll, yang berfungsi mengatur Top dan Left dari imgGambar terhadap kontainernya, dimana adalah jika pemakai melakukan geser kebawah, maka imgGambar perlu digeser keatas (- VScroll.Value), dan jika pemakai melakukan geser kekanan, maka imgGambar perlu digeser kekiri (-HScroll.Value)

```
Private Sub VScroll_Change()  
  ImgGambar.Top = -VScroll.Value      'Berlawanan arah  
End Sub  
Private Sub HScroll_Change()  
  ImgGambar.Left = -HScroll.Value     'Berlawanan arah  
End Sub
```

Melakukan koding terhadap Event Form Resize

```
Private Sub Form_Resize()  
  Call StatusBingkai  
  Call AturScrollBar  
  Call StatusVScroll  
  Call StatusHScroll  
End Sub
```

Untuk melakukan percobaan terhadap project ini anda perlu menentukan gambar yang berukuran besar pada imgGambar.

Latihan 16

Judul : Memanfaatkan Shape, Line, dan Timer untuk membuat Jam Analog

Untuk membuat jam analog beberapa hal yang perlu diketahui adalah :

1. Menghitung Besar sudut 1 jam, 1 menit dan 1 detik, dimana :

1 Lingkaran Jam = 360 derajat, maka

1 jam = 30 derajat (360/12)

1 menit = 6 derajat (360/60)

1 detik = 6 derajat (360/60)

2. Mengkonversi Jam menjadi Sudut, dimana :

Jam 1 = 60 derajat

Jam 2 = 30 derajat

Jam 3 = 0 derajat

Jam 4 = - 30 derajat atau 330 derajat

Jam 5 = - 60 derajat atau 300 derajat

Rumus deret aritmatika adalah $(n-1) * b + a$, dimana n adalah suku, a adalah nilai awal, b adalah beda, maka berdasarkan deret diatas :

$b = -30$

$a = 60$

Sehingga dapat dirumuskan menjadi :

$(\text{Jam} - 1) * -30 + 60$

Rumuskan sendiri untuk yang menit dan detik

$(\text{Menit} - 1) * -6 + 84$

(Detik - 1) * -6 + 84

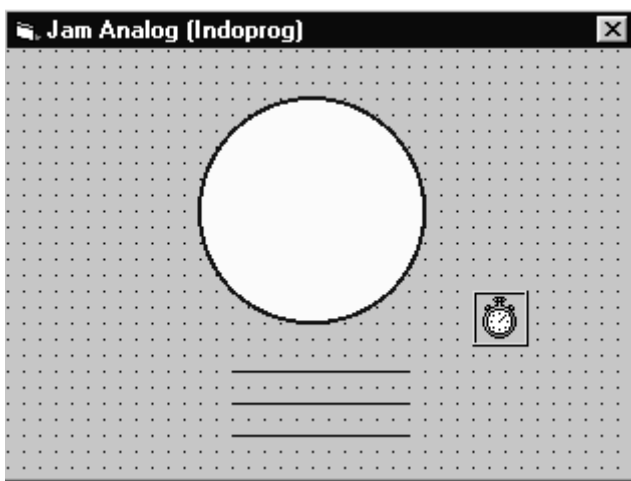
2. Menghitung koordinat Cartesius (x,y) untuk jarum jam, menit dan detik

Untuk menghitung koordinat Cartesius, kita perlu menguraikan suatu koordinat Polar menjadi komponen X dan Y dengan rumus :

$$X = R \text{ Cos sudut}$$

$$Y = R \text{ Sin sudut}$$

Tanamkan pada Form 1 Shape, 3 Line, dan 1 Timer, sehingga menjadi sebagai berikut :



Aturlah masing-masing properti menjadi :

Kontrol	Properti	Value
Form1	BorderStyle	1 - Fixed Single
Shape1	Name BackStyle BackColor BorderColor BorderWidth Visible	Bingkai 1 - Opaque Atur sendiri Atur sendiri Atur sendiri False
Line1	Name Visible	JarumJam False
Line2	Name Visible	JarumMenit False
Line3	Name Visible	JarumDetik False
Timer1	Name Interval Enable	Timer 1000 (1 detik) True

Pada General Declaration, deklarasikan variabel-variabel berikut :

Option Explicit

```
Dim PanjangJJam As Integer      'untuk panjang jarum jam
Dim PanjangJMenit As Integer
Dim PanjangJDetik As Integer
Dim PusatX As Integer          'titik pusat jam
Dim PusatY As Integer
```

Buatlah Sub sebagai berikut :

Sub HitungSkala, berfungsi membuat diameter Bingkai menjadi cocok pada form, menghitung panjang masing-masing jarum, dan menghitung titik pusat bingkai.

```
Sub HitungSkala()
Bingkai.Top = 0
Bingkai.Left = 0
Bingkai.Height = Me.ScaleHeight      'Hitung diameter bingkai
Bingkai.Width = Me.ScaleWidth
PanjangJJam = (4 / 10 * Me.ScaleHeight) \ 2      'Hitung panjang jarum
Jam (40% dari diameter)
PanjangJMenit = (6 / 10 * Me.ScaleHeight) \ 2    ' 60% dari diameter
PanjangJDetik = (8 / 10 * Me.ScaleHeight) \ 2    ' 80% dari diameter
PusatX = Me.ScaleWidth \ 2              'Hitung titik pusat
Bingkai
PusatY = Me.ScaleHeight \ 2
End Sub
```

Sub AturJarumJam, berfungsi menghitung sudut jarum jam berdasarkan waktu komputer, menghitung koordinat cartesius berdasarkan sudut, dan mengeser JarumJam berdasarkan koordinat.

```
Private Sub AturJarumJam()
Dim JamSekarang As Single
Dim SudutJam As Integer
Dim x, y
JamSekarang = (Now - Int(Now)) * 24      'Ambil waktu dengan
membuang bagian tanggal
SudutJam = (JamSekarang - 1) * -30 + 60    'Hitung sudut jam
x = PanjangJJam * Cos(SudutJam * 3.14 / 180) 'Hitung koordinat
Cartesius
y = PanjangJJam * Sin(SudutJam * 3.14 / 180)
JarumJam.X1 = PusatX                    'Atur jarum jam
JarumJam.Y1 = PusatY
JarumJam.X2 = PusatX + x
JarumJam.Y2 = PusatY - y
End Sub
```

```
Private Sub AturJarumMenit()
Dim WaktuSekarang As Single
Dim MenitSekarang As Integer
Dim SudutMenit As Integer
Dim x, y
WaktuSekarang = (Now - Int(Now)) * 24
MenitSekarang = (WaktuSekarang - Int(WaktuSekarang)) * 60
SudutMenit = (MenitSekarang - 1) * -6 + 84
x = PanjangJMenit * Cos(SudutMenit * 3.14 / 180)
y = PanjangJMenit * Sin(SudutMenit * 3.14 / 180)
```

```

JarumMenit.X1 = PusatX
JarumMenit.Y1 = PusatY
JarumMenit.X2 = PusatX + x
JarumMenit.Y2 = PusatY - y
End Sub
Private Sub AturJarumDetik()
Dim WaktuSekarang As Single
Dim DetikSekarang As Integer
Dim SudutDetik As Integer
Dim x, y
WaktuSekarang = (Now - Int(Now)) * 24 * 60
DetikSekarang = (WaktuSekarang - Int(WaktuSekarang)) * 60
SudutDetik = (DetikSekarang - 1) * -6 + 84
x = PanjangJDetik * Cos(SudutDetik * 3.14 / 180)
y = PanjangJDetik * Sin(SudutDetik * 3.14 / 180)
JarumDetik.X1 = PusatX
JarumDetik.Y1 = PusatY
JarumDetik.X2 = PusatX + x
JarumDetik.Y2 = PusatY - y
End Sub
Private Sub AturJarum()
Call AturJarumJam           'Atur Jarum Jam
Call AturJarumMenit         'Atur Jarum Menit
Call AturJarumDetik         'Atur Jarum Detik
End Sub

```

Melakukan koding pada Event Load dan Form Resize

```

Private Sub Form_Load()
Me.Height = 3000           'Atur tinggi Form
Me.Width = 3000           'Lebar
Call HitungSkala           'Hitung Skala
Me.Visible = True
Bingkai.Visible = True   'Buat jadi Visible
Call AturJarum            'Atur jarum
JarumJam.Visible = True   'Buat jadi Visible
JarumMenit.Visible = True
JarumDetik.Visible = True
End Sub
Private Sub Form_Resize()
Call HitungSkala           'Hitung skala ketika terjadi perubahan
ukuran Form
End Sub

```

Melakukan koding pada Event Timer

```

Private Sub Timer_Timer()
Call AturJarum           'Atur jarum jam setiap detik
End Sub

```

Tips : Membuat Form Berbentuk Bulat

Dengan menggunakan fungsi API, anda dapat membuat form berbentuk Bulat sesuai dengan ukuran Jam Analog Diatas, adapun perubahan yang perlu dilakukan adalah :

Atur properti BorderStyle menjadi 0 - None

Tambahkan sebuah module kedalam project anda dan lakukan koding sebagai berikut :

```
Public Declare Function CreateEllipticRgn Lib "gdi32" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
Public Declare Function SetWindowRgn Lib "user32" (ByVal hwnd As Long, ByVal hRgn As Long, ByVal bRedraw As Boolean) As Long

Public XP As Single
Public yp As Single
```

Lakukan perubahan koding pada event Form Load dan Form Resize

```
Private Sub Form_Load()
XP = Screen.TwipsPerPixelX 'Bagian ini yang ditambahkan
yp = Screen.TwipsPerPixelY
Me.Height = 3000 'Atur tinggi Form
Me.Width = 3000 'Lebar
Call HitungSkala 'Hitung Skala
Me.Visible = True
Bingkai.Visible = True 'Buat jadi Visible
Call AturJarum 'Atur jarum
JarumJam.Visible = True 'Buat jadi Visible
JarumMenit.Visible = True
JarumDetik.Visible = True
End Sub
Private Sub Form_Resize()
Dim mFormRegion As Long
Call HitungSkala
mFormRegion = CreateEllipticRgn(Bingkai.Left, Bingkai.Top,
Bingkai.Width / XP, Bingkai.Height / yp)
SetWindowRgn Me.hwnd, mFormRegion, False
End Sub
```

Latihan 17

Judul : Tantangan Pemrograman

1. Buka kembali Latihan 1, dan tambahkan fasilitas Zoom (+) kalau klik kiri, Zoom (-) kalau klik kanan pada Gambar. Adapun persentase pembesaran maupun pengecilan adalah 25%
2. Buka kembali Latihan 2, dan tambahkan fasilitas memindahkan jam Analog setelah form berbentuk Lingkaran, karena BorderStyle 0-None tidak dapat dipindah-pindahkan.
3. Buatlah program yang dapat mencatat Absensi Karyawan ke sebuah text file dengan format, pada Form tampil Jam Digital dan Analog, sebuah textbox untuk memasukkan No Induk Pegawai :

DDMMYYYYHHMMXXXXXXK

Dimana :

DDMMYYYY (Tanggal)

HHMM (Jam dan Menit)

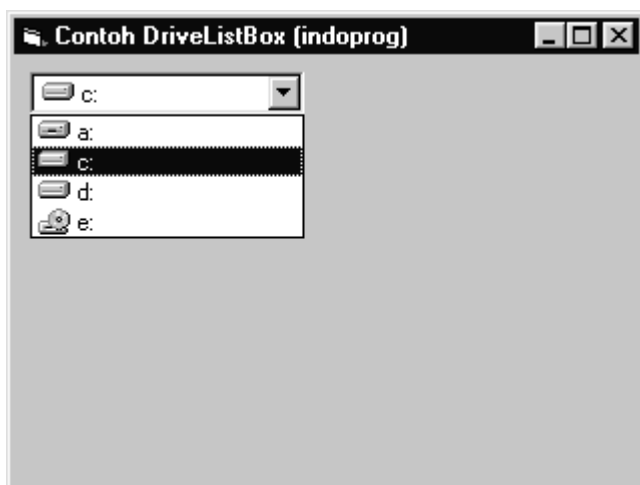
XXXXXX (No Induk Pegawai)

K (Kode 1 = Masuk 2 = Keluar)

Modul 8, Kontrol Standard (bagian 5)

DriveListBox

DriveListBox dapat digunakan untuk menampilkan drive yang terdapat pada sistem komputer



Properti	Fungsi
BackColor	Digunakan untuk menentukan warna latarbelakang kontrol
CauseValidation	Digunakan untuk menentukan apakah event Validate di aktifkan ketika kontrol akan kehilangan fokus (False, True)
DragMode	Menentukan modus drag-n-drop (0 - manual, 1 - automatic)
DragIcon	Digunakan untuk menentukan icon yang digunakan pada operasi drag-n-drop pada kontrol label
Enabled	Menentukan apakah kontrol label dapat menerima event yang dibuat oleh pemakai, misalnya event Click, MouseDown, MouseUp, dll.
Font	Digunakan untuk menentukan Font, Size, dan Style huruf yang digunakan pada kontrol DriveListBox
ForeColor	Digunakan untuk menentukan warna tulisan yang digunakan pada kontrol DriveListBox
Height	Digunakan untuk menentukan tinggi dari kontrol
Index	Digunakan untuk menentukan nomor index, jika kontrol tersebut merupakan kontrol array.
Left	Digunakan untuk menentukan jarak kiri kontrol dari kontainernya.
MouseIcon	Digunakan untuk menentukan icon mouse yang digunakan ketika pointer mouse berada diatas kontrol label tersebut, properti ini berarti bila properti MousePointer diset menjadi (99 - Custom)
MousePointer	Digunakan untuk menentukan bentuk pointer yang digunakan ketika mouse pointer berada diatas kontrol label tersebut.
TabIndex	Digunakan untuk menentukan urutan tabindex, 0 adalah kontrol yang pertama mendapat fokus jika form dijalankan. Properti ini efektif jika setting properti TabStop adalah True
TabStop	Digunakan untuk menentukan apakah textbox dapat menerima fokus karena penekanan tombol TAB oleh pemakai (False, True)
Tag	Properti ini dapat digunakan sebagai tempat menyimpan data sementara yang berkaitan dengan kontrol label tersebut
Top	Digunakan untuk menentukan jarak atas kontrol dari kontainernya.

Value	Digunakan untuk mendapatkan / menentukan nilai dari ScrollBar
Visible	Digunakan untuk menentukan apakah kontrol label tersebut Visible atau tidak (False, True)
Width	Digunakan untuk menentukan lebar kontrol.

Mendapatkan Drive yang aktif di DriveListBox

Untuk mendapatkan drive yang aktif di DriveListBox, anda dapat menggunakan properti Drive.

DirListBox

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



Properti	Fungsi
Appearance	Digunakan untuk menentukan penampilan dari kontrol (0 - Flat, 1 - 3D)
BackColor	Digunakan untuk menentukan warna latarbelakang kontrol
CauseValidation	Digunakan untuk menentukan apakah event Validate di aktifkan ketika kontrol akan kehilangan fokus (False, True)
DragMode	Menentukan modus drag-n-drop (0 - manual, 1 - automatic)
DragIcon	Digunakan untuk menentukan icon yang digunakan pada operasi drag-n-drop pada kontrol label
Enabled	Menentukan apakah kontrol label dapat menerima event yang dibuat oleh pemakai, misalnya event Click, MouseDown, MouseUp, dll.
Font	Digunakan untuk menentukan Font, Size, dan Style huruf yang digunakan pada kontrol.
ForeColor	Digunakan untuk menentukan warna tulisan yang digunakan pada kontrol.
Height	Digunakan untuk menentukan tinggi dari kontrol
Index	Digunakan untuk menentukan nomor index, jika kontrol tersebut merupakan kontrol array.
Left	Digunakan untuk menentukan jarak kiri kontrol dari kontainernya.
MouseIcon	Digunakan untuk menentukan icon mouse yang digunakan ketika pointer mouse berada diatas kontrol label tersebut, properti ini berarti bila properti MousePointer diset menjadi (99 - Custom)
MousePointer	Digunakan untuk menentukan bentuk pointer yang digunakan ketika mouse pointer berada

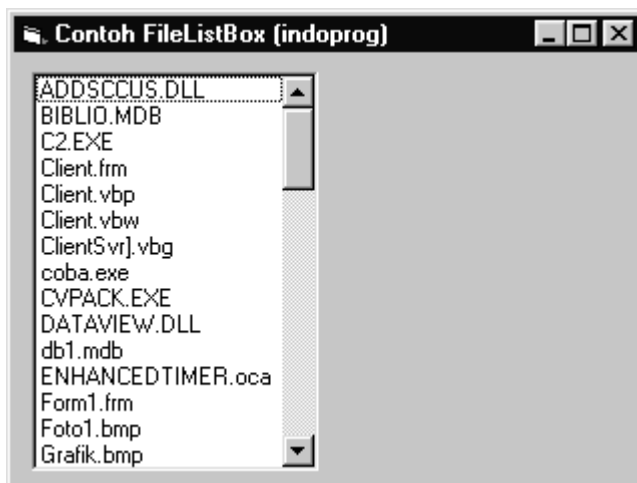
	diatas kontrol label tersebut.
TabIndex	Digunakan untuk menentukan urutan tabindex, 0 adalah kontrol yang pertama mendapat fokus jika form dijalankan. Properti ini efektif jika setting properti TabStop adalah True
TabStop	Digunakan untuk menentukan apakah textbox dapat menerima fokus karena penekanan tombol TAB oleh pemakai (False, True)
Tag	Properti ini dapat digunakan sebagai tempat menyimpan data sementara yang berkaitan dengan kontrol label tersebut
Top	Digunakan untuk menentukan jarak atas kontrol dari kontainernya.
Value	Digunakan untuk mendapatkan / menentukan nilai dari ScrollBar
Visible	Digunakan untuk menentukan apakah kontrol label tersebut Visible atau tidak (False, True)
Width	Digunakan untuk menentukan lebar kontrol.

Mendapatkan Path yang aktif pada DirListBox

Untuk mendapatkan Path yang sedang aktif pada DirListBox, anda dapat menggunakan properti Path.

FileListBox

DirListBox dapat digunakan untuk menampilkan file-file yang terdapat pada folder yang sedang aktif.



Properti	Fungsi
Appearance	Digunakan untuk menentukan penampilan dari kontrol (0 - Flat, 1 - 3D)
Archive	Digunakan untuk menentukan apakah FileListBox dapat menampilkan file dengan atribut Archive atau tidak.
BackColor	Digunakan untuk menentukan warna latarbelakang kontrol
CauseValidation	Digunakan untuk menentukan apakah event Validate di aktifkan ketika kontrol akan kehilangan fokus (False, True)
DragMode	Menentukan modus drag-n-drop (0 - manual, 1 - automatic)
DragIcon	Digunakan untuk menentukan icon yang digunakan pada operasi drag-n-drop pada kontrol label
Enabled	Menentukan apakah kontrol label dapat menerima event yang dibuat oleh pemakai, misalnya event Click, MouseDown, MouseUp, dll.

Font	Digunakan untuk menentukan Font, Size, dan Style huruf yang digunakan pada kontrol.
ForeColor	Digunakan untuk menentukan warna tulisan yang digunakan pada kontrol.
Hidden	Digunakan untuk menentukan apakah FileListBox dapat menampilkan file dengan atribut Hidden atau tidak.
Height	Digunakan untuk menentukan tinggi dari kontrol
Index	Digunakan untuk menentukan nomor index, jika kontrol tersebut merupakan kontrol array.
Left	Digunakan untuk menentukan jarak kiri kontrol dari kontainernya.
Normal	Digunakan untuk menentukan apakah FileListBox dapat menampilkan file dengan atribut Normal atau tidak.
MouseIcon	Digunakan untuk menentukan icon mouse yang digunakan ketika pointer mouse berada diatas kontrol label tersebut, properti ini berarti bila properti MousePointer diset menjadi (99 - Custom)
MousePointer	Digunakan untuk menentukan bentuk pointer yang digunakan ketika mouse pointer berada diatas kontrol label tersebut.
MultiSelect	Digunakan untuk menentukan apakah item dalam listbox dapat dipilih lebih dari satu atau tidak (0 - None, 1 - Simple, 2 - Extended) Perbedaan antara 1 - Simple dengan 2 - Extended, adalah pada Simple pemilihan beberapa item sekaligus tidak dapat menggunakan bantuan Shift.
Pattern	Digunakan untuk menentukan pola wildcard yang digunakan untuk membatasi file yang akan ditampilkan dalam kontrol (*. * semua file, *.bmp, ?A*.jpg)
ReadOnly	Digunakan untuk menentukan apakah FileListBox dapat menampilkan file dengan atribut ReadOnly atau tidak.
System	Digunakan untuk menentukan apakah FileListBox dapat menampilkan file dengan atribut System atau tidak.
TabIndex	Digunakan untuk menentukan urutan tabindex, 0 adalah kontrol yang pertama mendapat fokus jika form dijalankan. Properti ini efektif jika setting properti TabStop adalah True
TabStop	Digunakan untuk menentukan apakah textbox dapat menerima fokus karena penekanan tombol TAB oleh pemakai (False, True)
Tag	Properti ini dapat digunakan sebagai tempat menyimpan data sementara yang berkaitan dengan kontrol label tersebut
Top	Digunakan untuk menentukan jarak atas kontrol dari kontainernya.
Value	Digunakan untuk mendapatkan / menentukan nilai dari ScrollBar
Visible	Digunakan untuk menentukan apakah kontrol label tersebut Visible atau tidak (False, True)
Width	Digunakan untuk menentukan lebar kontrol.

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)

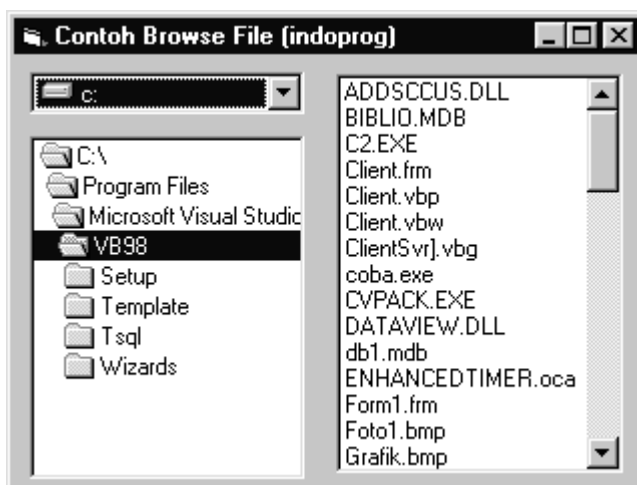
Untuk membatasi file berdasarkan atributnya, anda dapat menggunakan properti Archive, Hidden, Normal, ReadOnly dan System, dengan menentukan masing-masing menjadi True atau False.

Contoh :

```
MyFile.System = 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.



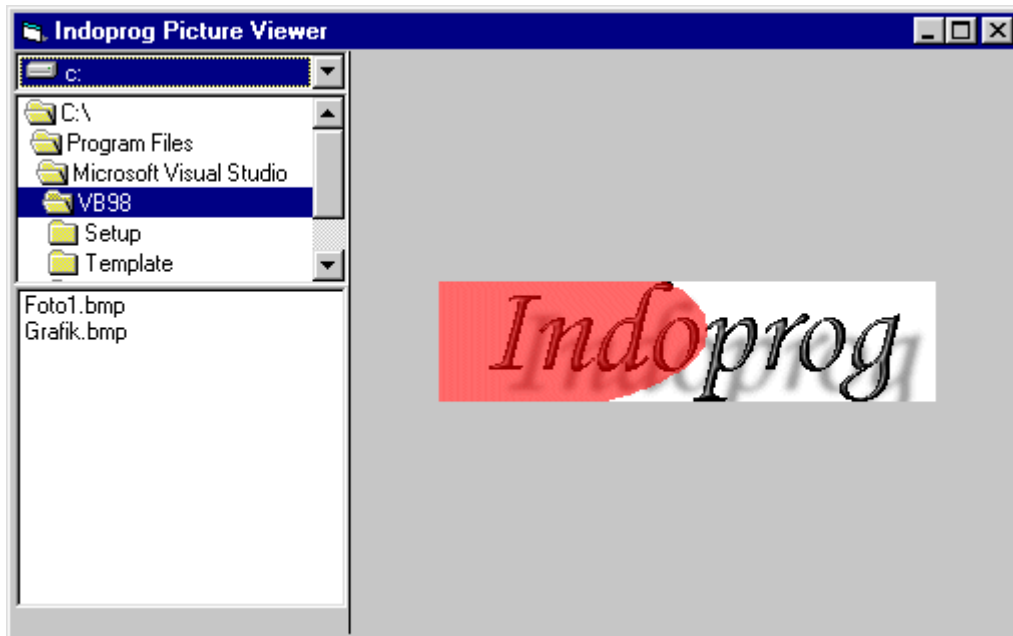
Maka Koding yang dapat dilakukan adalah sebagai berikut :

```
Private Sub MyDrive_Change ()  
MyDir.Path = MyDrive.Drive  
End Sub  
Private Sub MyDir_Change ()  
MyFile.Path = MyDir.Path  
End Sub
```

Latihan 18

Judul : Membuat Picture Viewer

Tanamkan pada form anda kontrol-kontrol berikut : DriveListBox, DirListBox, FileListBox, Line, dan Image



Kita perlu mempersiapkan beberapa variabel Form Level yang dapat digunakan oleh semua Sub yang ada, adapun variabel tersebut adalah :

```
Dim LebarTempatGambar As Integer      'Untuk menyimpan lebar tempat
Gambar yang tersedia
Dim TinggiTempatGambar As Integer     'Tinggi Gambar
Dim PosisiKiriGambar As Integer       'Posisi kiri tempat Gambar
Dim PosisiAtasGambar As Integer       'Posisi atas tempat Gambar
```

Selanjutnya kita buat sebuah Sub untuk menentukan posisi dan ukuran masing-masing kontrol diatas form ketika ukuran Form berubah :

```
Private Sub HitungPosisi()
MyDrive.Top = 0                'Kontrol Drive Rapat
MyDrive.Left = 0              'Kiri atas Form
MyDrive.Width = 2500
MyDir.Top = MyDrive.Height    'Kontrol Dir berada dibawah
MyDir.Left = 0                'kontrol Drive dan rapat kiri
MyDir.Width = 2500
MyDir.Height = Me.ScaleHeight \ 3 'Tingginya 1/3 tinggi form
```

```

MyFile.Top = MyDir.Top + MyDir.Height      'Kontrol File berada dibawah
kontrol Dir
MyFile.Left = 0
MyFile.Width = 2500
MyFile.Height = Me.ScaleHeight - MyDrive.Height - MyDir.Height
'Tingginya adalah tinggi sisa setelah Drive dan Dir
Pemisah.Y1 = 0                            'Garis Pemisah Antara Drive, Dir,
File dengan Gambar
Pemisah.Y2 = Me.ScaleHeight
Pemisah.X1 = MyDrive.Width + 1            'Berada disamping kanan kontrol
Drive
Pemisah.X2 = MyDrive.Width + 1
PosisiAtasGambar = 0                      'Hitung Tempat sisa untuk Gambar
PosisiKiriGambar = Pemisah.X1 + 50
TinggiTempatGambar = Me.ScaleHeight
LebarTempatGambar = Me.ScaleWidth - PosisiKiriGambar
End Sub

```

Kita juga perlu membuat sebuah Sub yang berfungsi membuat tampilan Gambar berada ditengah kalau ukurannya lebih kecil dari tempat gambar yang tersedia.

```

Private Sub BuatGambarKeTengah()
If Gambar.Width < LebarTempatGambar Then      'Jika lebih sempit
    Gambar.Left = PosisiKiriGambar + (LebarTempatGambar - Gambar.Width)
    \ 2
Else
    Gambar.Left = PosisiKiriGambar            'Rapat kiri
End If
If Gambar.Height < TinggiTempatGambar Then
    Gambar.Top = PosisiAtasGambar + (TinggiTempatGambar - Gambar.Height)
    \ 2
Else
    Gambar.Top = PosisiAtasGambar
End If
End Sub

```

Selanjutnya kita lakukan koding pada Event Load untuk membatasi jenis file yang ditampilkan pada FileListBox:

```

Private Sub Form_Load()
'Batasi hanya file Grafik
MyFile.Pattern = "*.bmp;*.gif;*.jpg;*.wmf;*.ico"
End Sub

```

Lakukan koding untuk Event Resize pada Form untuk memposisikan ulang semua kontrol kalau terjadi perubahan ukuran Form :

```

Private Sub Form_Resize()
Call HitungPosisi                'Hitung ulang posisi dan ukuran
Call BuatGambarKeTengah          'Buat grafik ke tengah kalau lebih kecil
dari tempat gambar
End Sub

```

Lakukan koding pada Event DriveListBox, DirListBox, FileListBox

```
Private Sub MyDrive_Change()  
MyDir.Path = MyDrive.Drive  
End Sub  
Private Sub MyDir_Change()  
MyFile.Path = MyDir.Path  
End Sub  
Private Sub MyFile_Click()  
Dim fName As String  
If Right(MyFile.Path, 1) = "\" Then      'Kalau sudah ada backslash  
    fName = MyFile.Path & MyFile.FileName  
Else  
    fName = MyFile.Path & "\" & MyFile.FileName  'Tambahkan \  
End If  
Gambar.Visible = False                  'Sembunyikan Gambar  
Gambar.Picture = LoadPicture(fName)     'GantiGambar  
Call BuatGambarKeTengah                  'Buat ketengah kalau lebih  
kecil dari tempat gambar  
Gambar.Visible = True                    'Tampilkan  
End Sub
```

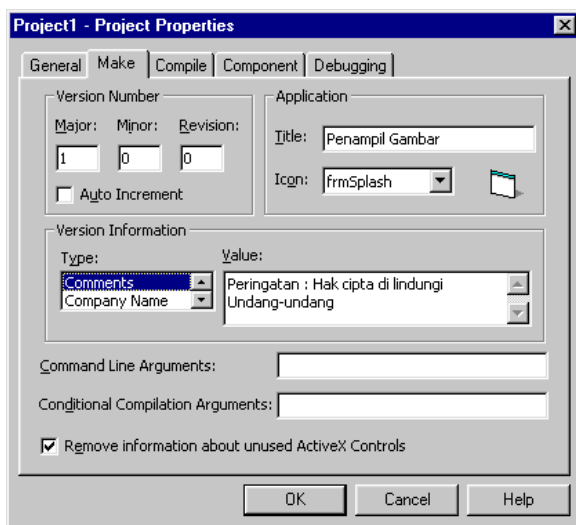
Latihan 19

Judul : Membuat Splash Screen

Aktifkan kembali project Latihan 1

1. Ganti name dari Form1 menjadi frmViewer
2. Pada menu Project, pilih Add Form, kemudian pilih Splash Screen.

Kemudian pada menu Project, pilih Project Properties, dan pilih tab Make



Aturlah properti berikut :

Properti	Value
Title	Penampil Gambar
Icon	frmSplash
Comments	Peringatan : Hak cipta di lindungi Undang-undang
Company Name	indoprogram@yahoogroups.com
Legal Copyright	Hak Cipta dan Distribusi oleh :
ProductName	Just4U

Dan lakukan koding berikut :

```
Private Sub Form_Load()  
    lblLicenseTo.Caption = "Izin pakai untuk Indoprogram-vb"
```

```

    lblPlatform.Caption = "Windows 95/NT"
    lblVersion.Caption = "Versi" & App.Major & "." & App.Minor & "." &
App.Revision
    lblCompanyProduct.Caption = App.Title
    lblProductName.Caption = App.ProductName
    lblCopyright.Caption = App.LegalCopyright
    lblCompany.Caption = App.CompanyName
    lblWarning.Caption = App.Comments
End Sub

```

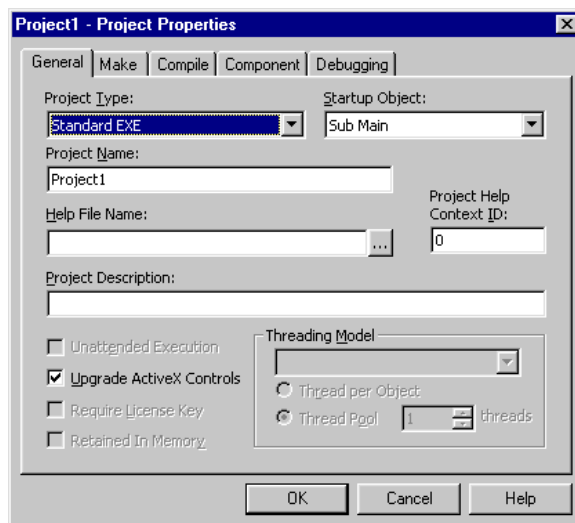
4. Selanjutnya tambahkan sebuah modul, dengan menu Project AddModule, dan lakukan koding berikut :

```

Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
Sub Main()
frmSplash.Show           'Tampilkan Splash Screen
frmSplash.Refresh        'Pastikan Splash Screen ditampilkan sempurna
Sleep (1000)              'Berhenti 1 detik
frmViewer.Show           'Tampilkan frmViewer
Unload frmSplash         'Unload frmSplash
End Sub

```

Pada menu Project, Project Properties, tentukan StartUp Object menjadi SubMain



Latihan 20

Judul : Tantangan pemrograman

Aktifkan kembali Latihan Sebelumnya, dan :

- Aplikasi tidak bisa dijalankan lebih dari 1 kali
- Lengkapi kemampuan Scroll kalau ukuran gambar lebih besar dari tempat yang disediakan.
- Lengkapi dengan kemampuan Zoom kalau pemakai Klik kiri (+) dan Klik kanan (-)
- Lengkapi fasilitas cetak gambar ke Printer.

Modul 9, MDI Form, dan Menu

Mengenal Multiple Document Interface

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.

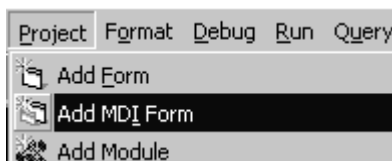
Sedangkan aplikasi seperti Microsoft Word menggunakan MDI, yaitu terdiri dari suatu MDIForm, dan didalamnya merupakan form-form anak (MDIchild).

Ada beberapa hal yang harus diperhatikan dalam penggunaan MDIForm adalah :

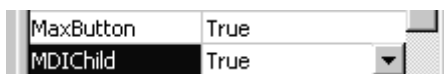
1. Didalam satu project hanya dapat terdiri dari satu MDIForm
2. Anda tidak dapat menempatkan kontrol-kontrol secara langsung pada MDIForm, kecuali kontrol yang memiliki properti Alignment, atau menempatkannya diatas kontainer seperti PictureBox.
3. Anda tidak dapat menggunakan metode penggambaran (Print,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

Membuat suatu Aplikasi MDI pada Visual Basic

1. Membuat suatu MDI form
Dari menu Insert, Pilih Add MDI Form. (Suatu aplikasi hanya dapat terdiri dari satu MDI form)

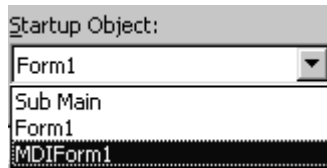


2. Membuat suatu Form menjadi MDI Child
Dari properti Form tersebut, ubah nilai properti MDI Child-nya menjadi True.



2. Menentukan StartUp Objek

Pada Menu Tools, pilih Command Options, pilih Tab Project, Tentukan StartUp Object ke MDI form yang telah ditambahkan.



Karakteristik dari MDI Form

- Semua child 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.

Mendapatkan MDI Child yang sedang aktif.

Untuk mendapatkan form-form yang sedang aktif didalam 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
```

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
```

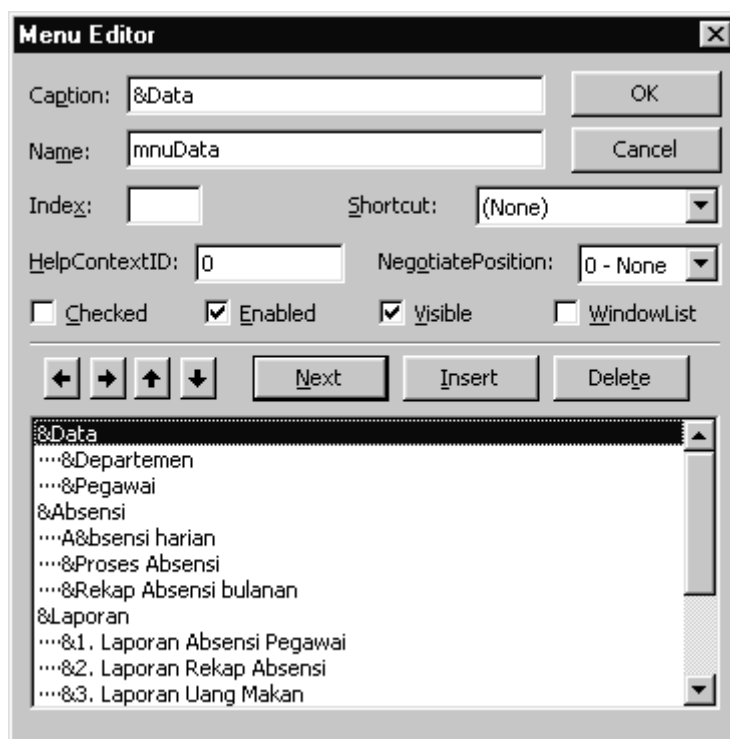
```
Private Sub mnuArrangeIcons_Click()
```

```
    Arrange vbArrangeIcons
```

```
End Sub
```

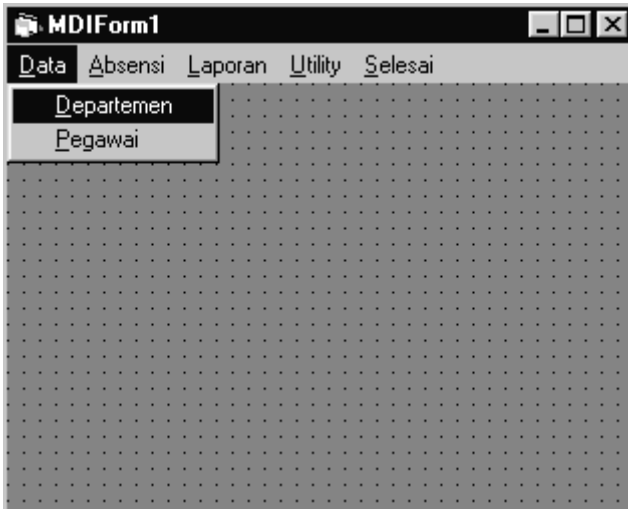
Membuat Menu Pada Visual Basic

Pembuatan menu pada Visual Basic dapat dilakukan dengan bantuan Menu Editor yang terdapat pada Menu Tools.



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 mengetikkan Caption dan Name, selanjutnya klik pada Next, dan ketikkan menu yang berikutnya, sampai selesai. Selanjutnya adalah membuat Sub Menu dengan melakukan klik pada panah kanan dan sebaliknya. Anda dapat membuat sampai 5 Sub Menu (6 kalau termasuk Menu Utama).

Pada contoh diatas akan menghasilkan menu sebagai berikut :

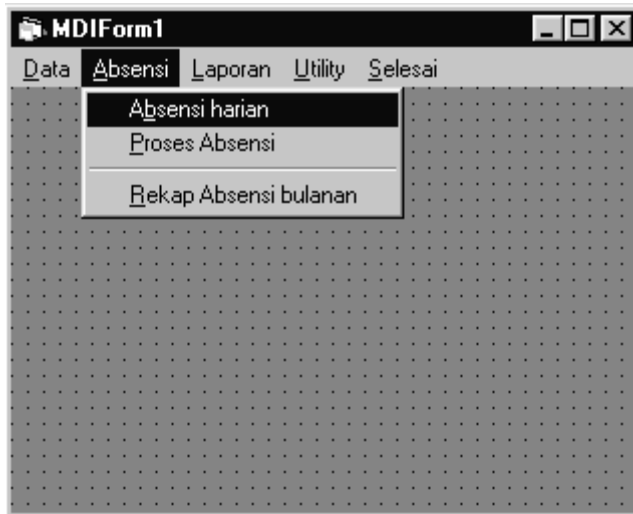


Membuat Garis Pemisah antar Menu

Anda dapat menggunakan tanda - (karakter hyphen) untuk membuat garis pemisah antar menu pada Caption.



Sehingga akan menghasilkan menu sebagai berikut :



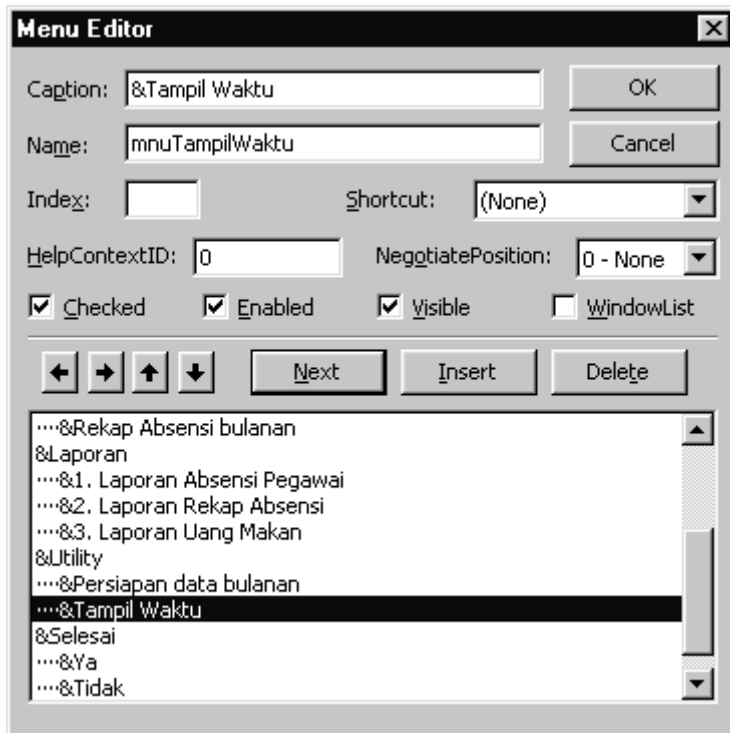
Melakukan Koding terhadap Menu

Untuk melakukan koding terhadap item menu, anda dapat membuka menu pada saat design dan akan menghasilkan Event Click untuk item menu tersebut, contoh :

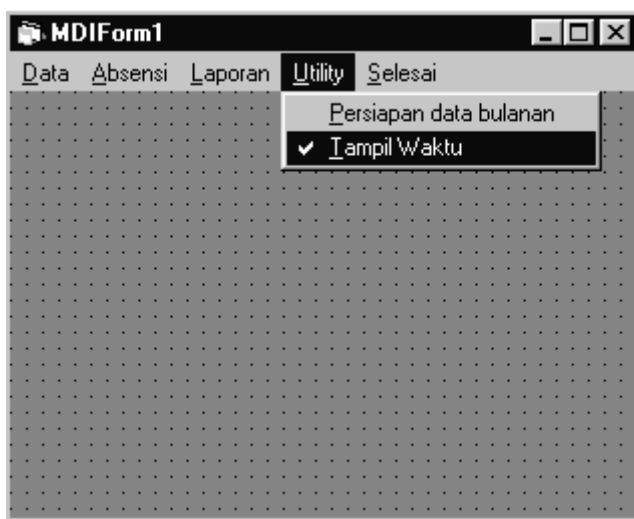
```
Private Sub mnuDataDepartemen_Click()  
    frmDepartemen.Show  
End Sub
```

Membuat Menu yang memiliki tanda Check

Anda dapat membuat menu yang memiliki tanda Check dengan memanfaatkan option Checked pada Menu Editor.



Sehingga akan menghasilkan menu sebagai berikut :



Anda dapat mengatur keaktifan tanda Check dari item menu tersebut dengan koding sebagai berikut :

```

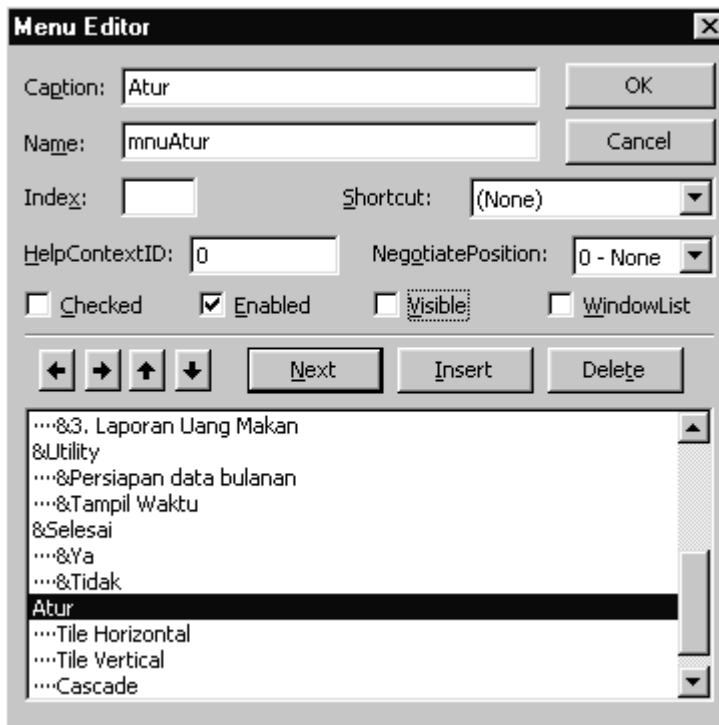
Private Sub mnuTampilWaktu_Click()
mnuTampilWaktu.Checked = Not mnuTampilWaktu.Checked
If mnuTampilWaktu.Checked Then
    frmWaktu.Show
Else
    Unload frmWaktu
End If
    
```

End Sub

Anda dapat memanfaatkan option Visible untuk menyembunyikan menu ataupun menampilkan suatu item menu berdasarkan kondisi tertentu, dan option Enabled untuk membuat suatu menu dapat dipilih atau tidak.

Membuat Menu Pop Up

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



Kemudian lakukan koding pada Event MouseDown pada MDIform

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

Sehingga kalau dilakukan klik kanan pada form akan menampilkan suatu Pop Up menu yang berupa Sub Menu dari mnuAtur.



Latihan 21

Judul : Membuat MDIform dan Menu

1. Aktifkan kembali Latihan Modul 4
2. Tambahkan MDIForm pada project tersebut, dengan menu Project, kemudian pilih Add MDIform, kemudian aturlah properti sebagai berikut :

Kontrol	Properti	Value
MDIForm1	Name WindowState	frmMain vbMaximize

3. Jadikan form-form lain yang berada pada project tersebut menjadi MDIChild, dengan membuat properti MDIChild pada masing-masing form menjadi True.
4. Kemudian tambahkan menu berikut pada MDIForm anda, dengan menu Tools, Menu Editor, dan ketikkan data berikut.

Data	mnuData
- Peserta	mnuDataPeserta
- Negara	mnuDataNegara
- Propinsi	mnuDataPropinsi
- Garis pemisah	mnuPemisah1
- Ujian	mnuDataUjian
Laporan	mnuLaporan
- Daftar Peserta	mnuLaporanDaftarPeserta
- Peserta Ujian dan Nilai	mnuLaporanPesertaUjian
Cetak	mnuCetak
- Cetak Sertifikat	mnuCetakSertifikat
- Cetak Label Amplop	mnuCetakLabelAmplop
Selesai	mnuSelesai
- Ya	mnuYa
- Tidak	mnuTidak

Atur hirarki yang bersesuaian dengan tampilan diatas.

6. Kemudian Lakukan koding untuk menampilkan Form berdasarkan masing-masing pilihan pada menu, Contoh :

```
Private Sub mnuDataPeserta_Click()  
    frmPeserta.Show
```

```
End Sub
Private Sub mnuDataNegara_Click()
frmNegara.Show
End Sub
Private Sub mnuDataPropinsi_Click()
frmPropinsi.Show
End Sub
Private Sub mnuYa_Click()
Unload Me
End Sub
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
```

Modul 9, System Objek

Memahami Objek Printer

Untuk memcetak hasil aplikasi anda ke printer, anda dapat menggunakan objek Printer yang disediakan oleh Visual Basic. Pada objek printer tersedia berbagai metoda pencetakan text dan penggambaran grafik (Line, Circle, Pset, dll) seperti yang tersedia pada form maupun kontrol PictureBox.

Objek printer juga mendukung properti ScaleWidth, ScaleHeight, ScaleTop, dan ScaleLeft. Anda dapat memeriksa kembali fungsi properti-properti ini pada control Picture.

Mencetak Data ke Objek Printer

Anda dapat menggunakan metoda print untuk mengirim tulisan yang akan dicetak ke objek printer. Anda dapat juga menggunakan metoda Newpage untuk membuat printer mencetak tulisan berikutnya ke halaman baru. Metoda EndDoc untuk memulai pencetakan yang sebenarnya, sedangkan KillDoc untuk membatalkan pencetakan. Anda juga dapat menggunakan properti Page untuk mendapatkan nomor halaman.

Contoh :

```
Printer.Print "Indoprog-vb, forum diskusi dan belajar bersama Visual  
Basic"  
Printer.NewPage  
Printer.Print "Ini akan dicetak pada halaman kedua"  
Printer.EndDoc
```

Untuk menentukan jumlah rangkap pencetakan dapat menggunakan properti Copies.

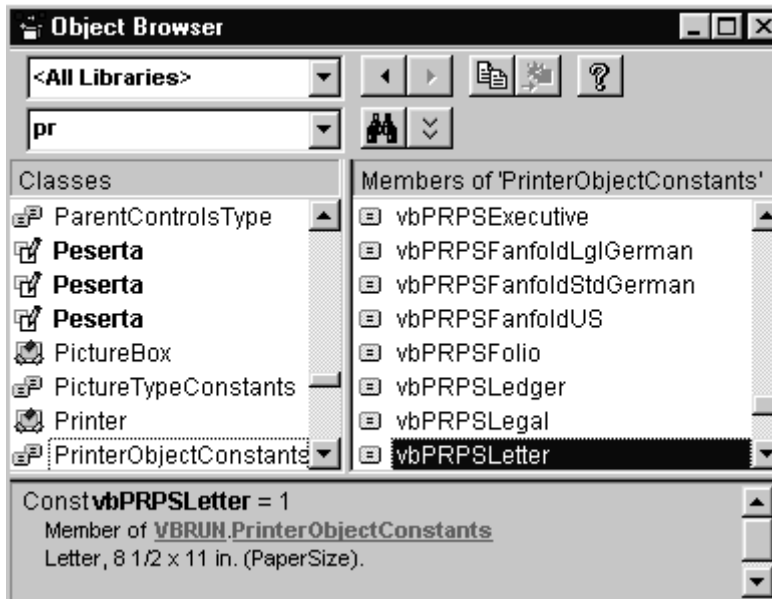
Contoh :

```
Printer.Copies = 2  
Printer.Print "Indoprog-vb, forum diskusi dan belajar bersama Visual  
Basic"  
Printer.NewPage  
Printer.Print "Ini akan dicetak pada halaman kedua"  
Printer.EndDoc
```

Anda dapat juga menggunakan metode penggambaran grafik seperti Line, Circle, PSet, dan PaintPicture seperti yang terdapat pada Form dan PictureBox.

Menentukan ukuran kertas

Properti PaperSize dapat digunakan untuk menentukan ukuran kertas, misalnya vbPRPSLetter untuk 8 1/2 x 11 (Letter). Untuk konstanta jenis ukuran kertas dapat anda lihat dengan Object Browser pada PrinterObjectConstants.



Contoh :
`Printer.PaperSize = vbPRPSLetter`

Menentukan posisi pencetakan

Anda dapat mengatur/mendapatkan posisi pencetakan dengan menggunakan properti CurrentX dan CurrentY.

Contoh :
`Printer.CurrentX = 0.5*1440 '1/2 inch`
`Printer.CurrentY = 1*1440 '1 inch`
`Printer.Print "Hello Indoprog-vb"`

Catatan : Ukuran dinyatakan dalam Twips, dimana 1 inch = 1440 Twips

Menentukan font pencetakan

Anda dapat mengatur font pencetakan dengan menggunakan objek font. Pada objek font anda dapat mengatur setting seperti :

Properti	Keterangan
Bold	Mengaktifkan cetak tebal (True, False)
Name	Menentukan nama font yang digunakan, misalnya Verdana, Times New Roman, Arial, Courier New, dll
Italic	Mengaktifkan cetak miring (True, False)
Size	Menentukan ukuran font

StrikeThrough	Mengaktifkan cetak coret (True, False)
Underline	Mengaktifkan cetak garis bawah (True, False)

```
Contoh :
Printer.Font.Name = "Verdana"
Printer.Font.Size = 14
Printer.Font.Bold = True
Printer.Print "Hello Indoprog-vb"
```

Menentukan orientasi pencetakan

Anda dapat menggunakan properti Orientation untuk menentukan orientasi pencetakan Portrait atau Landscape (1-vbPRORPortrait, 2-vbPRORLandscape).

Contoh :

```
Printer.Orientation = vbPRORLandscape
```

Mendapatkan informasi printer yang terinstalasi pada sistem

Untuk mendapatkan jumlah printer yang terinstalasi pada sistem, anda dapat menggunakan properti Count, sedangkan nama printer dapat diperoleh dengan DeviceName, dan DriverName untuk mendapatkan nama Driver. Berikut ini adalah contoh mengisi nama Device dan Driver dari printer yang terinstalasi pada sistem pada suatu ComboBox.

Contoh :

```
For i = 0 To Printers.Count - 1
    cboPrinters.AddItem Printers(i).DeviceName & " [" & _
        Printers(i).DriverName & "]"
Next
```

Anda dapat juga mendapatkan informasi dengan properti Port (misalnya, LPT1:), memeriksa apakah printer tertentu dapat melakukan pencetakan berwarna dengan properti ColorMode (1-vbPRCMMonochrome or 2-vbPRCMColor).

Menentukan Printer yang akan digunakan

Jika anda tidak menentukan printer yang akan digunakan, maka printer yang akan digunakan adalah printer default pada sistem anda.

Untuk mencetak dengan menggunakan printer yang tertentu anda dapat melakukan setting printer ke printer yang bersesuaian. Contoh berikut akan melakukan setting printer ke nomor printer tertentu berdasarkan pilihan pemakai dari ComboBox.

Contoh :

```
Private Sub cmdMakeCurrent_Click()
```

```
Set Printer = Printers(cboPrinters.ListIndex)
End Sub
```

Mencetak Isi Form ke Printer

Untuk mencetak isi Form ke Printer anda dapat menggunakan Metoda PrintForm pada object Form.

Contoh :

```
Private Sub Command1_Click()
Me.PrintForm
End Sub
```

Memahami Objek App

Objek App disediakan oleh Visual Basic library dan mewakili aplikasi yang sedang dijalankan. Objek App memiliki banyak properti dan metoda, yang tidak akan dibahas secara keseluruhan disini.

Mendapatkan Path dimana aplikasi berada

Salah satu masalah dalam pengaktifan file adalah menentukan path dimana file tersebut berada, sebagaimana kita ketahui, pada aplikasi Windows umumnya di instalasi pada folder Program Files, tetapi ketika kita mengembangkan program tersebut, tentu saja kita lakukan pada folder sendiri, sehingga kadang-kadang menjadi masalah ketika program tersebut didistribusikan. Untuk mendapatkan Path dimana aplikasi berada kita dapat menggunakan properti Path pada objek App

Contoh :

```
cFolder = App.Path
Open cFolder & "\Pegawai.Dat" For Random As #1 Len = Len(DatPegawai)
```

Dengan potongan program diatas, kita senantiasa yakin bahwa file Pegawai.Dat akan berada di Path yang sama dengan aplikasi.

Mendapatkan nama file Executable

Anda dapat menggunakan properti EXEName untuk mendapatkan nama file Executable anda.

Contoh :

```
Private Sub Form_Load()
If UCase$(App.ExeName) <> "INDOPROG" Then
```

```
MsgBox "Anda tidak berhak mengganti nama program tanpa izin  
programmer"  
End  
End If  
End Sub
```

Memeriksa apakah aplikasi telah dijalankan

Kadang-kadang aplikasi kita hanya boleh dijalankan sekali pada komputer yang sama. Untuk memeriksa apakah instance yang sama dari aplikasi telah berjalan pada sistem, anda dapat menggunakan properti `PrevInstance`.

Contoh :

```
Private Sub Form_Load()  
If App.PrevInstance Then  
MsgBox "Aplikasi yang sama telah aktif"  
End  
End If  
End Sub
```

Tidak menampilkan aplikasi pada task list ketika sedang berjalan

Anda dapat menggunakan properti `TaskVisible` untuk menentukan apakah aplikasi ditampilkan di Task List ketika dijalankan. Jendela Task List akan dimunculkan ketika anda menekan `Ctrl + Alt + Del`.

Contoh :

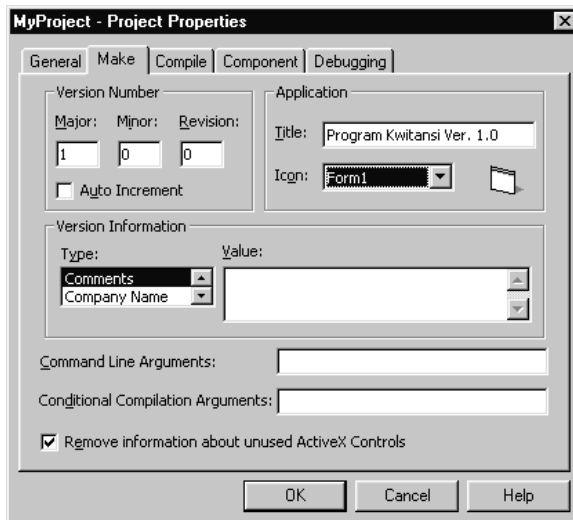
```
Sub Main()  
App.TaskVisible = False  
End Sub
```

Sedangkan properti `Title` dapat digunakan untuk menentukan judul aplikasi pada Task List.

Contoh :

```
Sub Main()  
App.Title = "Program Kwitansi Ver 1.0"  
End Sub
```

atau anda dapat juga menentukannya pada saat design, pada menu Project Properties



Sedangkan properti seperti *Major*, *Minor*, dan *Revision* mengembalikan informasi tentang versi dari aplikasi yang sedang dijalankan. *Comments*, *CompanyName*, *FileDescription*, *LegalCopyright*, *LegalTrademarks*, dan *ProductName* akan berguna ketika anda membuat Splash Screen ataupun dialog box About

Memahami Objek Screen

Pada contoh maupun latihan modul-modul sebelumnya kita banyak menggunakan Objek screen, terutama untuk mendapatkan Tinggi (Height) dan Lebar (Width) dari Layar komputer untuk menempatkan suatu Form ke posisi di Tengah layar. Contoh :

Contoh :

Sub Tengah (x)

```
x.Move (Screen.Width - x.Width) \ 2, _
        (Screen.Height - x.Height) \ 2
```

End Sub

Properti Height dan Width mengembalikan ukuran Tinggi dan Lebar layar dalam ukuran Twips, tetapi untuk fungsi-fungsi API kita membutuhkan ukuran dalam pixel. Untuk mendapatkan jumlah Twips per pixel kita dapat menggunakan properti TwipsPerPixelY dan TwipsPerPixelX.

Contoh :

```
'Mendapatkan ukuran tinggi dan lebar layar dalam Pixel
scrWidth = Screen.Width / Screen.TwipsPerPixelX
scrHeight = Screen.Height / Screen.TwipsPerPixelY
```

Dengan menggunakan objek Screen kita juga dapat memperoleh nama-nama font yang tersedia dilayar dengan menggunakan properti Font dan FontCount.

Contoh :

```
' Mengambil semua font pada objek Screen dan memasukkannya ke kontrol  
ListBox.  
Dim i As Integer  
For i = 0 To Screen.FontCount - 1  
    lstFonts.AddItem Screen.Fonts(i)  
Next
```

Anda dapat juga mengubah mouse pointer dengan menggunakan properti `MousePointer`.

Contoh :

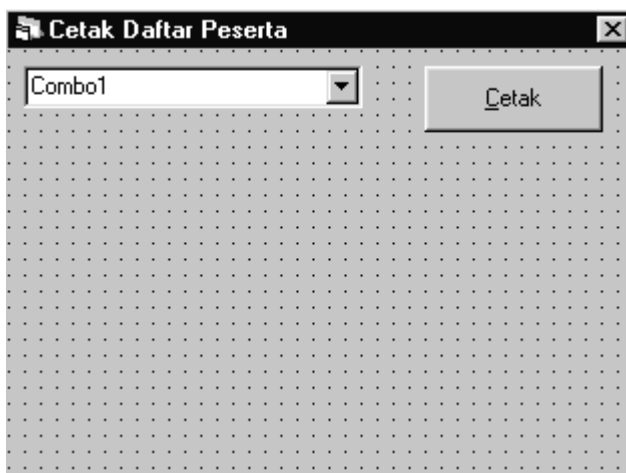
```
Screen.MousePointer = vbHourglass
```

Catatan : Efek mouse pointer hanya berlaku pada aplikasi anda.

Latihan 22

Judul : Mencetak data peserta ke Form (Print Preview)

1. Tambahkan pada project Latihan 1 sebuah form, dan tanamkan sebuah ComboBox dan Command sebagai berikut :



Kontrol	Properti	Value
Form1	Name Caption BorderStyle MDIChild	frmDaftarPeserta Cetak Daftar Peserta 3 - Fixed Dialog True
Combo1	Name	cboPrinter
Command1	Name Caption	cmdCetak &Cetak

2. Tambahkan lagi sebuah form dengan nama frmPreview, dan atur properti form sebagai berikut

Kontrol	Properti	Value
Form1	Name BorderStyle MDIChild	frmPreview 0 -None True

3. Lakukan Koding pada frmDataPeserta sebagai berikut :

'General Declaration

```
'Struktur data untuk random file
Private Type Peserta
Email As String * 30
nama As String * 30
Alamat As String * 30
Kota As String * 20
```

```

Propinsi As String * 20
Negara As String * 20
Tempat As String * 20
TglLahir As Date
Pria As Integer
End Type
'Deklarasi variabel form level
Dim DatPeserta As Peserta
Private Function InchToTwips(x)
InchToTwips = x * 1440
End Function
Private Sub cetak(x, y, z)
frmPreview.CurrentX = InchToTwips(x) 'Set posisi pencetakan
berdasarkan x
If Not z Then
    frmPreview.Print y;
Else
    frmPreview.Print y
    'Akhiri pencetakan dengan CrLf
End If
End Sub

Private Sub Form_Load()
Dim i As Integer
For i = 0 To Printers.Count - 1
    cboPrinters.AddItem Printers(i).DeviceName & " [" & _
        Printers(i).DriverName & "]"
Next
cboPrinters.ListIndex = 0
End Sub
Private Sub cmdCetak_Click()
Load frmPreview
frmPreview.Top = 0
frmPreview.Left = 0
frmPreview.Width = InchToTwips(8) 'Mengatur lebar form menjadi 8
inch
frmPreview.Height = InchToTwips(11) 'Mengatur tinggi form menjadi 11
inch

frmPreview.AutoRedraw = True 'Membuat pencetakan dengan metoda
menjadi permanen

Open "Peserta.Dat" For Random As #1 Len = Len(DatPeserta)
JlhRec = LOF(1) / Len(DatPeserta)

For i = 1 To JlhRec
    If frmPreview.CurrentY = 0 Then 'Jika halaman baru, maka set
margin atas
        frmPreview.CurrentY = InchToTwips(0.75)
        frmPreview.Font.Name = "Courier New"
        frmPreview.Font.Size = 14 'Atur font untuk judul pencetakan
        frmPreview.Font.Bold = True
        Call cetak(0.75, "Laporan Data Peserta Indoprog-vb", True)
        frmPreview.Line (0.75 * 1440, frmPreview.CurrentY)-(7.25 * 1440,
frmPreview.CurrentY)
        frmPreview.Font.Name = "Courier New"
        frmPreview.Font.Size = 10
        frmPreview.Font.Bold = False
    
```

```
        Call cetak(0.75, "Email                               Nama
Kota", True)
        frmPreview.Line (0.75 * 1440, frmPreview.CurrentY)-(7.25 * 1440,
frmPreview.CurrentY)
        frmPreview.Print
    End If
    Get #1, i, DatPeserta
    Call cetak(0.75, DatPeserta.Email & Space(2) & _
                DatPeserta.nama & Space(2) & _
                DatPeserta.Kota, True)
    If i = JlhRec Then
        frmPreview.Line (0.75 * 1440, frmPreview.CurrentY)-(7.25 * 1440,
frmPreview.CurrentY)
        Call cetak(0.75, "By indoprog-vb", True)
    End If
    'If frmPreview.CurrentY >= frmPreview.Height - InchToTwips(0.75)
Then
        'Printer.NewPage                                'Sementara
dimatikan
        'End If
Next i
frmPreview.AutoRedraw = False
frmPreview.Visible = True
'Printer.EndDoc                                'Sementara dimatikan
Close #1
End Sub
```

4. Pada menu di MDIForm lakukan koding untuk mnuDaftarPeserta.

```
Private Sub mnuDaftarPeserta_Click()
frmDaftarPeserta.Show
End Sub
```

Latihan 23

Judul : Tantangan Penyempurnaan Program

1. Ubah program print preview diatas menjadi cetak ke Printer.
2. Buatlah program Cetak Kwitansi menggunakan Printer.
3. Lengkapi program Latihan Modul 7 dengan fasilitas cetak Grafik ke Printer.
4. Buatlah program Cetak Sertifikat menggunakan Printer.
5. Tambahkan SplashScreen untuk project diatas.

Fungsi Bantu pada Visual Basic

Fungsi Bantu Konversi

IsNumeric(ekspresi)

Mengembalikan nilai boolean yang menunjukkan suatu ekspresi dapat dievaluasi sebagai numerik

IsEmpty(ekspresi)

Mengembalikan nilai boolean yang menunjukkan suatu variabel telah diinitaliasi, biasanya fungsi ini digunakan untuk memeriksa variabel yang dideklarasikan sebagai Variant.

Contoh :

```
Dim A as Variant
If IsEmpty(A) Then
    MsgBox "Variabel A Empty"           'Message akan muncul karena A
    belum terinisialisai
End If
A = 1
If IsEmpty(A) Then
    MsgBox "Variabel A masih Empty"    'Tidak akan muncul lagi karena A
    telah                               'terinisialisasi sebagai Integer
End If
Dim B as Integer
If IsEmpty(B) Then
    MsgBox "Variabel B Empty"         'Tidak akan muncul karena B
    adalah Integer
End If
```

IsNull(ekspresi)

Mengembalikan nilai boolean yang menunjukkan suatu ekspresi mengandung data yang tidak valid, biasanya digunakan untuk memeriksa isi field recordset.

IsArray(varname)

Mengembalikan suatu nilai boolean yang menunjukan suatu variabel adalah suatu array.

IsDate(ekspresi)

Mengembalikan suatu nilai boolean yang menunjukan suatu ekspresi dapat dikonversi ke date.

IsError(ekspresi)

Mengembalikan suatu nilai boolean yang menunjukkan suatu ekspresi adalah nilai error

IsObject(ekspresi)

Mengembalikan suatu nilai boolean yang menunjukkan suatu ekspresi mengacu pada suatu OLE Automation object.

IsMissing(argname)

Mengembalikan suatu nilai boolean yang menunjukan suatu argumen optional pada procedure ada dilewatkan atau tidak

CBool(ekspresi)

Konversi suatu ekspresi ke boolean

Contoh :

```
A = 5: B = 5                                ' Inisialiasi variabel
Check = CBool(A = B)                        ' Check berisi nilai True

A = 0
Check = CBool(A)                            ' Check berisi False.
```

CByte(ekspresi)

Konversi ekspresi ke Byte

Contoh :

```
MyDouble = 125.5678                          ' MyDouble adalah suatu nilai
Double
MyByte = CByte(MyDouble)                     ' MyByte berisi 126.
```

CCur(ekspresi)

Konversi suatu ekspresi ke Currency

Contoh :

```
MyDouble = 543.214588                        ' MyDouble adalah suatu nilai
Double.
MyCurr = CCur(MyDouble * 2)                 ' Konversi hasil MyDouble * 2
                                              ' (1086.429176) menjadi suatu
                                              ' Currency (1086.4292).
```


CDate(date)

Konversi suatu ekspresi ke date

Contoh :

```
MyDate = "February 12, 1969"      ' Definisi waktu dalam string
MyShortDate = CDate(MyDate)      ' Konversi ke type data Date
MyTime = "4:35:47 PM"           ' Definisi waktu dalam string
MyShortTime = CDate(MyTime)     ' Konversi ke type data Date
```

CDBl(ekspresi)

Konversi suatu ekspresi ke Double

Contoh :

```
MyCurr = CCur(234.456784)       ' MyCurr adalah nilai Currency.
MyDouble = CDBl(MyCurr * 8.2 * 0.01) ' Konversi hasil ke suatu Double.
```

CInt(ekspresi)

Konversi suatu ekspresi ke Integer

Contoh :

```
MyDouble = 2345.5678            ' MyDouble adalah nilai Double.
MyInt = CInt(MyDouble)         ' MyInt berisi nilai 2346.
```

CLng(ekspresi)

Konversi suatu ekspresi ke Long

Contoh :

```
MyVal1 = 25427.45: MyVal2 = 25427.55 ' MyVal1, MyVal2 adalah Double
MyLong1 = CLng(MyVal1)              ' MyLong1 berisi nilai 25427.
MyLong2 = CLng(MyVal2)              ' MyLong2 berisi nilai 25428.
```

CSng(ekspresi)

Konversi suatu ekspresi ke single

Contoh :

```
' MyDouble1, MyDouble2 adalah Double.
MyDouble1 = 75.3421115: MyDouble2 = 75.3421555
MySingle1 = CSng(MyDouble1)        ' MySingle1 berisi nilai 75.34211.
MySingle2 = CSng(MyDouble2)        ' MySingle2 berisi nilai 75.34216.
```

CStr(ekspresi)

Konversi suatu ekspresi ke string

Contoh :

```
MyDouble = 437.324           ' MyDouble adalah Double
MyString = CStr(MyDouble)    ' MyString berisi nilai "437.324".
```

CVar(ekspresi)

Konversi suatu ekspresi ke Variant

Contoh :

```
MyInt = 4534                 ' MyInt adalah suatu Integer.
MyVar = CVar(MyInt & "000") ' MyVar berisi nilai string
4534000.
```

Asc(string)

Mengembalikan kode character dari huruf pertama di suatu string.

Contoh :

```
MyNumber = Asc("A")         ' Mengembalikan 65.
MyNumber = Asc("a")         ' Mengembalikan 97.
MyNumber = Asc("Apple")     ' Mengembalikan 65.
```

Chr(charcode)

Mengembalikan karakter dari suatu kode karakter

Contoh :

```
MyChar = Chr(65)            ' Mengembalikan A.
MyChar = Chr(97)            ' Mengembalikan a.
MyChar = Chr(62)            ' Mengembalikan >.
MyChar = Chr(37)            ' Mengembalikan %.
```

Format(ekspresi[, format[, hariPertamaDariMinggu[, mingguPertamaDariTahun]]])

Memformat suatu ekspresi berdasarkan ekspresi format

Contoh :

```
MyTime = #17:04:23#
MyDate = #January 27, 1993#
```

```
' Mengembalikan waktu sistem dalam format long time
MyStr = Format(Time, "Long Time")

' Mengembalikan tanggal sistem dalam format long date
MyStr = Format(Date, "Long Date")

MyStr = Format(MyTime, "h:m:s")           ' Mengembalikan "17:4:23".
MyStr = Format(MyTime, "hh:mm:ss AMPM") ' Mengembalikan "05:04:23
PM".
MyStr = Format(MyDate, "dddd, mmm d yyyy") ' Mengembalikan "Wednesday,
Jan 27 1993".

'Jika format tidak tersedia, suatu string dikembalikan.
MyStr = Format(23) ' Mengembalikan "23".

' Format User-defined
MyStr = Format(5459.4, "##,##0.00")      ' Mengembalikan "5,459.40".
MyStr = Format(334.9, "###0.00")        ' Mengembalikan "334.90".
MyStr = Format(5, "0.00%")              ' Mengembalikan "500.00%".
MyStr = Format("HELLO", "<")           ' Mengembalikan "hello".
MyStr = Format("This is it", ">")      ' Mengembalikan "THIS IS
IT".
```

Hex(number)

Oct(number)

Mengembalikan string yang mewakili Octal atau Hexa dari suatu bilangan

Contoh :

```
MyHex = Hex(5)           ' Mengembalikan 5.
MyHex = Hex(10)          ' Mengembalikan A.
MyHex = Hex(459)         ' Mengembalikan 1CB.
MyOct = Oct(4)           ' Mengembalikan 4.
MyOct = Oct(8)           ' Mengembalikan 10.
MyOct = Oct(459)        ' Mengembalikan 713.
```

Str(number)

Mengembalikan string yang mewakili suatu angka.

Contoh :

```
MyString = Str(459)      ' Mengembalikan " 459".
MyString = Str(-459.65) ' Mengembalikan "-459.65".
MyString = Str(459.001) ' Mengembalikan " 459.001".
```

Val(string)

Mengembalikan angka yang terkandung dalam suatu string.

Contoh :

```
MyVal =Val(" 1615 198th Street N.E.")      ' Mengembalikan 1615198
MyVal =Val("&HFFFF")                        ' Mengembalikan -1
MyVal =Val("2457")                          ' Mengembalikan 2457.
MyVal =Val(" 2 45 7")                       ' Mengembalikan 2457.
MyVal =Val("24 and 57")                     ' Mengembalikan 24.
```

Fungsi Bantu Waktu

Now

Mengembalikan suatu Variant (Date) yang menunjukkan tanggal dan waktu berdasarkan sistem komputer.

Time

Mengembalikan waktu sistem sekarang

Timer

Mengembalikan suatu bilangan yang menunjukan jumlah detik sejak tengah malam

Date

Mengembalikan tanggal sistem sekarang

Time = Time

Date = Date

Mengatur waktu atau tanggal sistem

Untuk sistem yang menjalankan Microsoft Windows 95, tanggal yang dibutuhkan harus berupa tanggal dari 1 Jan 1998 sampai 31 Des 2099. Untuk sistem yang menjalankan Microsoft Windows NT, tanggal yang dibutuhkan harus berupa tanggal dari 1 Jan 1980 sampai 31 Desember 2079.

Hour(time)

Minute(time)

Second(time)

Mengembalikan suatu Variant (Integer) berupa bilangan 0 s/d 23 untuk jam, 0 s/d 59 untuk menit, dan 0 s/d 59 untuk detik.

Day(date)

Month(date)

Year(date)

Mengembalikan suatu Variant (Integer) berupa bilangan 1 s/d 31 untuk bulan, 1 s/d 12 untuk bulan, dan tahun.

Weekday(date, [hariPertamaDariMinggu])

Mengembalikan suatu Variant (Integer) yang menunjukkan hari dari minggu

hariPertamaDariMinggu Optional. suatu konstanta yang menunjukkan hari pertama dari minggu, jika tidak ditentukan, maka vbSunday diasumsikan.

Konstanta	Nilai	Keterangan
vbUseSystem	0	Menggunakan setting NLS API
vbSunday	1	Minggu (default)
vbMonday	2	Senin
vbTuesday	3	Selasa
vbWednesday	4	Rabu
vbThursday	5	Kamis
vbFriday	6	Jumat
vbSaturday	7	Sabtu

Nilai yang dikembalikan :

Konstanta	Nilai	Keterangan
vbSunday	1	Minggu
vbMonday	2	Senin
vbTuesday	3	Selasa
vbWednesday	4	Rabu
vbThursday	5	Kamis
vbFriday	6	Jumat
vbSaturday	7	Sabtu

DateAdd(interval, number, date)

Mengembalikan suatu Variant (Date) yang mana merupakan hasil penjumlahan.

Bagian	Keterangan
interval	Ekspresi string yang menunjukkan interval waktu yang ingin dijumlahkan (Perlu)
number	Ekspresi numerik yang merupakan jumlah dari interval yang ingin dijumlahkan. Nilai Positif menunjukan (tanggal berikutnya), Nilai Negatif menunjukan (tanggal sebelumnya) (Perlu)
date	Suatu Variant (Date) atau literal yang merupakan tanggal dimana interval akan dijumlahkan padanya. (Perlu)

Setting argumen untuk interval :

Setting	Keterangan
yyyy	Tahun
q	Triwulan
m	Bulan
y	Hari dari tahun
d	Hari
w	Hari dari minggu
ww	Minggu
h	Jam
n	Menit
s	Detik

Contoh :

`DateAdd("m", 1, "31-Jan-95")` 'mengembalikan 28-Feb-95

`DateDiff(interval, date1, date2)`

Mengembalikan suatu Variant (Long) untuk menunjukkan bilangan dari interval waktu antara dua tanggal.

FAQ

Saya adalah pemula, dapatkan anda memberikan informasi tentang edisi Visual Basic ?

Microsoft membagi Visual Basic menjadi tiga edisi, yaitu Learning, Professional dan Enterprised, dimana memiliki harga sesuai dengan feature masing-masing, berikut ini penjelasan dari Microsoft :

Visual Basic tersedia dalam tiga versi, yang masing-masing sesuai dengan kebutuhan pengembangan.

Visual Basic Learning edition memungkinkan para programmer membuat aplikasi yang powerful untuk Microsoft Windows dan Windows NT®. Menyediakan semua kontrol intrinsic, ditambah grid, tab, dan kontrol data-bound. Dokumentasi yang tersedia pada edisi ini meliputi CD Learn VB Now ditambah dengan CD Microsoft Developer Network (MSDNT) Library yang mengandung dokumentasi lengkap online.

Professional edition menyediakan bagi professional komputer sejumlah feature lengkap dan tool-tool untuk solusi pengembangan bagi pihak ketiga. Meliputi semua feature pada Learning edition, ditambah dengan kontrol ActiveX, Internet Information Server Application Designer, integrated Visual Database Tools dan Data Environment, Active Data Objects, serta Dynamic HTML Page Designer. Dokumentasi yang tersedia pada edisi ini meliputi buku Visual Studio Professional Features ditambah CD Microsoft Developer Network yang mengandung dokumentasi lengkap online.

Enterprise edition memungkinkan para professional untuk membuat aplikasi tersebar, meliputi semua features pada Professional edition, ditambah Back Office tools seperti SQL Server, Microsoft Transaction Server, Internet Information Server, Visual SourceSafe, SNA Server, dan lain-lain. Dokumentasi cetak yang terdapat pada Enterprise edition meliputi buku Visual Studio Enterprise Features ditambah CD Microsoft Developer Network yang mengandung dokumentasi lengkap online.

Bagaimana menampilkan tanda ampersand (&) pada menu atau label?

Tanda ampersand adalah simbol khusus untuk windows untuk menandai shortcut key (ALT <key>) untuk mengakses melalui keyboard. Untuk menampilkan tanda ampersand anda dapat mengetikkan dua kali &&, misalnya Tutorial && programming.

Bagaimana cara termudah memori ulang urutan tab index kontrol?

Cara yang termudah memori tab index kontrol adalah, klik pada kontrol yang ingin anda buat sebagai tabstop terakhir, nomor tab indexnya ke 0, kemudian klik kontrol terakhir berikutnya dan nomor tab indexnya ke 0, dan seterusnya sampai kontrol yang pertama, sehingga semua kontrol ternomori dengan benar.

Bagaimana saya membuat accelerator pada label, sehingga textbox disampingnya mendapatkan fokus, kalau dilakukan Alt <key> ?

Pada Label kontrol buat tanda ampersand pada huruf yang mau dijadikan accelerator, kemudian berikan tab index textbox disampingnya sebagai nomor berikutnya.

Bagaimana saya membuat penekanan Enter pada suatu textbox dapat memindahkan fokus ke kontrol berikutnya ?

Untuk membuat penekanan Enter pada Textbox memiliki efek pemindahan fokus ke kontrol berikutnya, anda dapat memanipulasi KeyAscii yang terdapat pada Event KeyPress pada Textbox tersebut, contoh :

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    SendKeys "{Tab}"
    KeyAscii = 0
End If
End Sub
```

Bagaimana saya memeriksa apakah picturebox atau image dalam keadaan kosong ?

Untuk memeriksa apakah suatu picturebox atau image kosong, anda dapat membandingkannya dengan fungsi LoadPicture tanpa argumen, contoh :

```
Private Sub Command1_Click()
If Image1.Picture = LoadPicture Then
    MsgBox "Image1 kosong"
```

```
End If  
End Sub
```

Bagaimana saya menyimpan isi PictureBox ke file ?

Untuk menyimpan isi dari suatu PictureBox ke file, anda dapat menggunakan perintah SavePicture terhadap properti Image dari PictureBox tersebut, contoh :

```
Private Sub Command1_Click()  
SavePicture PicGambar.Image, "MyPicture.bmp"  
End Sub
```

Bagaimana saya memindahkan isi Clipboard ke suatu PictureBox ?

Untuk memindahkan isi Clipboard ke suatu PictureBox, anda dapat menggunakan metoda GetData yang terdapat pada objek Clipboard, contoh :

```
Private Sub Command1_Click()  
PicGambar.Picture = Clipboard.GetData  
End Sub
```

Bagaimana mendeklarasikan variabel yang dapat digunakan pada semua procedure dan forms ?

Deklarasikan variabel tersebut pada suatu modul dengan menempatkan kata Public didepannya.

```
Public cFolder As String
```

Bagaimana saya mengambil command-line parameters?

Anda dapat mengambilnya dengan fungsi bantu Command\$.

Bagaimana membuat looping 1 kali satu detik pada perintah for...next ?

Anda dapat mengkonbinasikannya dengan fungsi Timer, perhatikan koding berikut :

```
For i = 1 To 100  
    Debug.Print "PentaCom"  
    sekarang = Timer  
    Do While Timer < sekarang + 1  
        x = DoEvents()  
    Loop  
Next i
```

Bagaimana menjalankan perintah DOS dari program?

Anda dapat menggunakan perintah Shell dengan contoh sebagai berikut :

```
Private Sub Command1_Click()  
x = Shell("c:\windows\command.com /c dir", vbNormalFocus)  
End Sub
```

Bagaimana pindah keaktifan ke aplikasi yang sedang berjalan, atau menjalankannya ?

Deklarasikan FindWindow dan BringWindowToTop APIs pada modul, dan menggunakannya :

```
Public Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName As String, ByVal lpWindowName As String) As Long
```

```
Public Declare Function BringWindowToTop Lib "user32" Alias "BringWindowToTop" (ByVal hwnd As Long) As Long
```

```
Private Sub cmdBuka_Click()  
Dim hCalcWnd As Long  
Dim x As Long
```

```
hCalcWnd = FindWindow("SciCalc", "Calculator")
```

```
If hCalcWnd = 0 Then
```



```
x = Shell("CALC.EXE", vbNormalFocus)
Else
BringWindowToTop (hCalcWnd)
End If
End Sub
```

Bagaimana cara termudah mendapatkan jumlah form yang terbuka pada aplikasi saya?

Anda dapat menggunakan properti Count pada Forms, atau nilai yang dikembalikan oleh DoEvents

```
JlhFormTerbuka = Forms.Count
```

atau

```
JlhFormTerbuka = DoEvents
```

Bagaimana saya dapat mengetahui jumlah MDI child form yang terbuka dalam suatu MDI form?

Anda dapat menggunakan properti Count dari koleksi Forms yang akan mengembalikan jumlah form yang telah di load dalam project. Untuk mengetahui jumlah jendela MDI child, anda dapat menggunakan :

```
Public Function GetMDIChildCount() As Integer
```

```
Dim frm As Form
Dim cnt As Integer
```

```
For Each frm In Forms
If (frm.MDIChild And frm.Visible) Then cnt = cnt + 1
Next frm
```

```
GetMDIChildCount = cnt
```

```
End Function
```

Gunakan rutin diatas dengan:

```
NumberOfOpenForms = GetMDIChildCount()
```

Bagaimana cara menghindarkan pemakai menjalankan aplikasi lebih dari 1 kali pada komputer yang sama?

Pada aplikasi tertentu kita membutuhkan aplikasi hanya dijalankan satu kali setiap saat pada komputer yang sama. Untuk melakukan hal tersebut kita dapat menggunakan metoda app.previnstance apakah instance dari aplikasi telah pernah dijalankan sebelumnya, contoh :

```
Sub Main()
If App.PrevInstance Then
MsgBox "aplikasi telah berjalan"
End
Else
MainForm.Show
End If
End Sub
```

atau

```
Private Sub Form_Load()
If App.PrevInstance Then
MsgBox "aplikasi telah berjalan"
Unload Me
End If
End Sub
```

Bagaimana mendapatkan folder dimana program aktif ?

Untuk mendapatkan folder dimana aplikasi anda aktif, dapat digunakan properti App.Path, contoh :

```
cfolder = App.Path
Open cfolder & "\peserta.dat" For random As #1 Len=Len(DatPeserta)
```

Bagaimana saya membuat aplikasi saya berhenti sesaat ?

Untuk membuat aplikasi anda dapat berhenti sejumlah waktu tertentu, anda perlu membuat deklarasi fungsi API sebagai berikut dan memanfaatkannya dalam koding anda, contoh :

```
Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
```

```
'Untuk berhenti selama 5 detik, anda dapat melakukan call dengan sleep(5000)
```

Bagaimana saya mengirim email lewat aplikasi ?

Anda perlu memasukkan komponen MAPI kedalam project anda, kemudian tanamkan MAPI.Session dan MAPI.Messages kedalam form, dan lakukan koding seperti contoh berikut :

```
Private Sub Command1_Click()  
' Pemakaian username dan password sangat tergantung pada setting klien mail anda  
With MAPISession1  
    .UserName = "noname"  
    .Password = "password"  
    .SignOn  
End With  
  
' Kirim sebuah surat  
With MAPIMessages1  
    .SessionID = MAPISession1.SessionID  
    .Compose  
    .RecipAddress = "hendra@indoprog.com"  
    .AddressResolveUI = False  
    .ResolveName  
    .MsgSubject = "Subject anda"  
    .MsgNoteText = "Pesan Anda"  
    .Send False  
End With  
End Sub
```

Dapatkan saya mendapatkan username login windows?

Untuk mendapatkan username, anda perlu mendeklarasikan fungsi API GetUserName sebagai berikut :

```
Private Declare Function GetUserName Lib "advapi32.dll" _  
Alias "GetUserNameA" (ByVal lpBuffer As String, nSize _  
As Long) As Long
```

Kemudian anda dapat mebuat fungsi RetrieveUserName sebagai berikut :

```
Public Function RetrieveUserName() As String  
  
' Deklarasi konstanta untuk maksimal panjang username  
Const MaxLen = 50  
  
' Deklarasi variabel untuk menampung username  
Dim strName As String  
  
' Digunakan untuk mendapatkan panjang username  
  
Dim lngRetVal As Long  
  
' Isi strName dengan spasi  
strName = Space$(MaxLen)  
  
'Ambil username  
  
lngRetVal = GetUserName(strName, MaxLen)  
  
' Buang spasi kanan  
strName = Trim$(strName)
```

```
' Karena GetUserName mengembalikan suatu karakter null pada bagian akhir
' sehingga perlu dihilangkan.
strName = Left$(strName, Len(strName) - 1)
```

```
RetrieveUserName = strName
```

```
End Function
```

Dan akhirnya anda dapat memanggil fungsi RetrieveUsername dari program anda.

Bagaimana jangkauan variabel yang dipesan dengan perintah DIM?

Suatu variabel yang di DIM dalam sub maupun function adalah lokal terhadap rutin tersebut, dan akan dimusnahkan setelah rutin selesai dieksekusi.

Variabel yang di DIM sebagai Private pada bagian general declarations dari suatu Form adalah tersedia bagi seluruh rutin pada form tersebut, dan tetap tersedia selama form dalam keadaan ter Load.

Variabel yang di DIM sebagai Public pada bagian general declarations dari suatu Form tersedia bagi seluruh rutin pada form dengan nama variabel tersebut, misalnya Nama, dan tersedia bagi form atau modul diluar form tersebut dengan nama namaform.namavariabel, misalnya MyForm.Nama selama form tersebut dalam keadaan ter Load.

Variabel yang di DIM sebagai Private pada bagian general declarations dari suatu Module adalah tersedia bagi seluruh rutin dalam module tersebut.

Variabel yang di DIM sebagai Public pada bagian general declarations dari suatu Module adalah tersedia bagi seluruh module dan form dalam project tersebut.

Mengapa saya mengalami Overflow Error dalam perkalian:

Dim x as Long

```
x = 33 * 1000 'menyebabkan overflow error
```

VB, dalam melakukan perhitungan hasilnya disimpan ke x, menggunakan suatu variabel sementara yang memiliki type yang sama dengan type data dalam ekspresi yang dihitung. Pada kasus diatas kedua nilai adalah dalam batasan integer +/- 32767, sedangkan hasil perkalian adalah > 32767, sehingga menyebabkan overflow, solusinya adalah membuat salah satu nilainya menjadi long dengan sufiks &, contoh ...

```
x = 33 * 1000&
```

Dapatkah anda memberi contoh membuat database dengan koding ?

Anda dapat membentuk database secara koding dengan menggunakan **DAO**, contoh berikut saya menggunakan References DAO 3.51 Object Library:

```
Dim MyWs As Workspace
Dim MyDb As Database
Dim MyTb As TableDef
Dim Myfl As Field
Dim MyId As Index
Set MyWs = DBEngine.Workspaces(0)
Set MyDb = MyWs.CreateDatabase("C:\Data.mdb", dbLangGeneral)
Set MyTb = MyDb.CreateTableDef("pegawai")
Set Myfl = MyTb.CreateField("nip", dbText, 6)
MyTb.Fields.Append Myfl
Set Myfl = MyTb.CreateField("nama", dbText, 30)
Myfl.AllowZeroLength = False
MyTb.Fields.Append Myfl
Set Myfl = MyTb.CreateField("alamat", dbText, 50)
MyTb.Fields.Append Myfl
Set Myfl = MyTb.CreateField("pria", dbBoolean, 30)
MyTb.Fields.Append Myfl
Set Myfl = MyTb.CreateField("tglLahir", dbDate, 30)
MyTb.Fields.Append Myfl
```

```

Set Myfl = MyTb.CreateField("berat", dbSingle, 30)
MyTb.Fields.Append Myfl
Set Myfl = MyTb.CreateField("gaji", dbCurrency)
MyTb.Fields.Append Myfl
Set Myfl = MyTb.CreateField("foto", dbLongBinary)
MyTb.Fields.Append Myfl
Set Myfl = MyTb.CreateField("id", dbLong)
Myfl.Attributes = dbAutoIncrField           'Membuat Field autoincrement
MyTb.Fields.Append Myfl

'Membuat Index
Set MyId = MyTb.CreateIndex("P_Key")
MyId.Primary = True
Set Myfl = MyId.CreateField("nip")
MyId.Fields.Append Myfl
MyTb.Indexes.Append MyId
MyDb.TableDefs.Append MyTb
MyDb.Close
    
```

Anda dapat membentuk database secara koding dengan menggunakan **ADO**, contoh berikut saya menggunakan References Microsoft ADO Ext. 2.1 for DDL and Security :

```

Dim cat As New ADOX.Catalog
Dim tbl As New ADOX.Table
Dim idx As New ADOX.Index

cat.Create "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=c:\Data.mdb"

tbl.Name = "MyTable"
Set tbl.ParentCatalog = cat
tbl.Columns.Append "nip", adVarChar, 6
tbl.Columns.Append "nama", adVarChar, 30
tbl.Columns("nama").Properties("Jet OLEDB:Allow Zero Length") = False
tbl.Columns.Append "alamat", adVarChar, 50
tbl.Columns.Append "pria", adBoolean
tbl.Columns.Append "tglahir", adDate
tbl.Columns.Append "gaji", adCurrency
tbl.Columns.Append "berat", adSingle
tbl.Columns.Append "foto", adLongVarBinary
tbl.Columns.Append "id", adInteger
tbl.Columns("id").Properties("AutoIncrement") = True

idx.Name = "P_Key"
idx.Columns.Append "nip"

'Append the index to the table
tbl.Indexes.Append idx

cat.Tables.Append tbl

Set cat = Nothing
    
```

Anda dapat membuat tabel dengan SQL Statement yang dijalankan dengan metode execute pada Object Database dengan **DAO**. Saya menggunakan References DAO 3.51 Object Library untuk contoh berikut :

```

Dim MyWs As Workspace
Dim MyDb As Database
Dim sSQL As String
Set MyWs = DBEngine.Workspaces(0)
Set MyDb = MyWs.CreateDatabase("C:\Data.mdb", dbLangGeneral)
sSQL = "CREATE TABLE pegawai (nip char(6), nama char(30) not null, alamat char(50), pria logical," & _
"tglLahir datetime, berat single, gaji currency, foto longbinary," & _
"id autoincrement, constraint P_KEY PRIMARY KEY (nip));"
MyDb.Execute sSQL
MyDb.Close
    
```

Bagaimana saya membuat database yang diproteksi dengan password ?

Dengan DAO

Anda dapat menggunakan ";pwd=password" pada waktu Create database

```
Set MyDb = MyWs.CreateDatabase("C:\Data.mdb", dbLangGeneral + ";pwd=indoprog")
```

Dengan ADO

Anda dapat menggunakan Jet OLEDB:Database Password=password; pada connection string waktu create catalog

```
cat.Create "Provider=Microsoft.Jet.OLEDB.4.0;Jet OLEDB:Database Password=password;Data Source=c:\Data1.mdb"
```

Bagaimana membuka database yang diproteksi dengan menggunakan password ?

Dengan DAO

Anda dapat menggunakan ";pwd=password" pada waktu OpenDatabase

```
Set MyDb = MyWs.OpenDatabase("C:\Data1.mdb", False, False, "; pwd=password")
```

Dengan ADO

Anda dapat menggunakan Jet OLEDB:Database Password=password; pada connection string waktu open connection

```
MyConn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Jet OLEDB:Database Password=password;Data Source=c:\Data1.mdb"
```

Saya telah memiliki database dan tabel pegawai, bagaimana saya menambahkan suatu index yang baru kedalamnya ?

Anda dapat membuka kembali database tersebut, dan menambahkan index pada table tersebut :

Dengan DAO :

```
Dim MyWs As Workspace
Dim MyDb As Database
Dim MyTb As TableDef
Dim Myfl As Field
Dim MyId As Index
Dim lAda As Boolean
Set MyWs = DBEngine.Workspaces(0)
Set MyDb = MyWs.OpenDatabase("C:\Data.mdb")
Set MyTb = MyDb.TableDefs("pegawai")
lAda = False
For Each idx In MyTb.Indexes
    If idx.Name = "S_Key" Then
        lAda = True
        MsgBox "Index telah ada !"
    End If
Next idx
If Not lAda Then
    Set MyId = MyTb.CreateIndex("S_Key")
    Set Myfl = MyId.CreateField("nama")
    MyId.Fields.Append Myfl
    MyTb.Indexes.Append MyId
End If
MyDb.Close
```

Dengan ADO :

```
Dim MyConn As New ADODB.Connection
Dim cat As New ADOX.Catalog
Dim tbl As New ADOX.Table
Dim idx As New ADOX.Index
```

Dim lada As Boolean

```
MyConn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=c:\Data.mdb"
Set cat.ActiveConnection = MyConn
Set tbl = cat.Tables("MyTable")
```

lada = False

```
For Each idx In tbl.Indexes
    If idx.Name = "S_Key" Then
        MsgBox "Index telah ada !"
        lada = True
    End If
Next idx
```

```
If Not lada Then
    idx.Name = "S_Key"
    idx.Columns.Append "nama"
```

```
'Append the index to the table
tbl.Indexes.Append idx
End If
```

```
Set cat = Nothing
Set MyConn = Nothing
```

Dengan SQL :

```
Dim MyWs As Workspace
Dim MyDb As Database
Dim sSQL As String
Set MyWs = DBEngine.Workspaces(0)
Set MyDb = MyWs.OpenDatabase("C:\Data.mdb")
sSQL = "CREATE INDEX S_Key ON pegawai (nama);"
MyDb.Execute sSQL
MyDb.Close
```

Saya ingin memeriksa apakah didalam database MDB saya telah memiliki tabel yang bernama "MyTable" atau tidak?

Anda dapat memeriksa keberadaan tabel dalam suatu database dengan melakukan For Each ... Next untuk menelusuri semua objek TableDef yang terdapat pada koleksi TableDefs dari database yang bersangkutan, sambil melakukan pemeriksaan terhadap properti Name dari masing-masing TableDef yang didapatkan.

Contoh :

Dengan DAO :

```
Dim MyWs As Workspace
Dim MyDb As Database
Dim MyTb As TableDef
Dim lAda As Boolean
Set MyWs = DBEngine.Workspaces(0)
Set MyDb = MyWs.OpenDatabase("C:\Data.mdb")
lAda = False
For Each MyTb In MyDb.TableDefs
    If MyTb.Name = "MyTable" Then
        MsgBox "Tabel telah ada !"
        lAda = True
    End If
Next MyTb
```

Dengan ADO :

```
Dim MyConn As New ADODB.Connection
Dim cat As New ADOX.Catalog
```

```
Dim tbl As New ADOX.Table
Dim lada As Boolean

MyConn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=c:\Data.mdb"
Set cat.ActiveConnection = MyConn
lAda = False
For Each tbl In cat.Tables
If tbl.Name = "MyTable" Then
    MsgBox "Tabel telah ada"
    lada = True
End If
Next tbl
```

Saya ingin melakukan compact database, bagaimana hal tersebut dilakukan?

Untuk melakukan compact database

Dengan Dao :

```
DBEngine.CompactDatabase "C:\database.mdb", "C:\newdb.mdb"
```

"Microsoft Jet and Replication Objects 2.5 Library" (JRO).

```
Dim oJetEngine As JRO.JetEngine
```

```
Dim sSourceConn As String
```

```
Dim sDestConn As String
```

```
Set oJetEngine = New JRO.JetEngine
```

```
' Engine Type = 4 compacts an Access database in 3.5 format
```

```
' Engine Type = 5 compacts an Access database in 4.0 format (default)
```

```
sSourceConn = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=.\\database.mdb;" & _
    "Jet OLEDB:Engine Type=5;"
```

```
sDestConn = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=.\\newdb.mdb;" & _
    "Jet OLEDB:Engine Type=5;"
```

```
' Compact the database (makes a new copy)
```

```
oJetEngine.CompactDatabase sSourceConn, sDestConn
```

Membuat Program Mengeja Bilangan

Oleh : Hendra & Susan Dewichan

Bagaimana anda membaca 123456789.55 ?

Tentu saja : Seratus Duapuluh Tiga Juta Empat Ratus Lima Puluh Enam Ribu Tujuh Ratus Delapan Puluh Sembilan Rupiah Lima Puluh Lima Sen

Tetapi bagaimana kalau komputer yang melakukan hal tersebut ?

Tentu saja kita membutuhkan suatu program yang dapat menterjemahkan angka-angka tersebut menjadi kata-kata yang sesuai, dan dapatkah anda bayangkan reptnya melakukan hal tersebut.

Saya sering membaca di mailing list pemrograman, banyak orang menanyakan akan hal tersebut, dan memang tidak sesederhana seperti kita mengeja ketika menulis Kwitansi, Slip Tabungan, CEK dan Giro.

Baiklah, sekarang saya akan membantu anda untuk memahami dasar dari program mengeja terbilang, mari kita lihat kembali angka 123456789.55 yang anda baca sebagai Seratus Duapuluh Tiga Juta Empat Ratus Lima Puluh Enam Ribu Tujuh Ratus Delapan Puluh Sembilan Rupiah Lima Puluh Lima Sen, dan akan saya penggal menjadi potongan yang yang lebih kecil sebagai berikut :

123 456 789 .55

dan perhatikan potongan ejaannya

Seratus Duapuluh Tiga **Juta**

Empat Ratus Limapuluh Enam **Ribu**

Tujuh Ratus Delapanpuluh Sembilan **Rupiah**

Limapuluh Lima **Sen**

Jadi pada prinsipnya kita memotongnya menjadi bagian (123) Juta, (456) Ribu, (789) Rupiah, dan (55) Sen.

Sekarang misalnya kita mengambil 123 yang kita baca sebagai Seratus Duapuluh Tiga, dimana kalau kita penggal menjadi potongan yang lebih kecil sebagai berikut :

1 2 3

Se ratus

Dua puluh

Tiga

Jadi pada prinsipnya kita memotongnya menjadi bagian (1) ratus, (2) puluh, (3) satuan.

Sehingga akhirnya kita dapat menyusun suatu program sebagai berikut :

```
Public Function terbilang(x As Currency)
Dim triliun As Currency
Dim milyar As Currency
Dim juta As Currency
Dim ribu As Currency
Dim satu As Currency
Dim sen As Currency
Dim baca As String
'Jika x adalah 0, maka dibaca sebagai 0
If x= 0 Then
    baca = angka(0, 1)
Else
    'Pisah masing-masing bagian untuk triliun, milyar, juta, ribu,
rupiah, dan sen
    triliun = Int(x * 0.001 ^ 4)
    milyar = Int((x - triliun * 1000 ^ 4) * 0.001 ^ 3)
    juta = Int((x - triliun * 1000 ^ 4 - milyar * 1000 ^ 3) / 1000 ^ 2)
    ribu = Int((x - triliun * 1000 ^ 4 - milyar * 1000 ^ 3 - juta * 1000
^ 2) / 1000)
    satu = Int(x - triliun * 1000 ^ 4 - milyar * 1000 ^ 3 - juta * 1000
^ 2 - ribu * 1000)
    sen = Int((x - Int(x)) * 100)
    'Baca bagian triliun dan ditambah akhiran triliun
    If triliun > 0 Then
        baca = ratus(triliun, 5) + "triliun "
    End If
    'Baca bagian milyar dan ditambah akhiran milyar
    If milyar > 0 Then
        baca = ratus(milyar, 4) + "milyar "
    End If
    'Baca bagian juta dan ditambah akhiran juta
    If juta > 0 Then
        baca = baca + ratus(juta, 3) + "juta "
    End If
    'Baca bagian ribu dan ditambah akhiran ribu
    If ribu > 0 Then
        baca = baca + ratus(ribu, 2) + "ribu "
    End If
    'Baca bagian rupiah dan ditambah akhiran rupiah
    If satu > 0 Then
        baca = baca + ratus(satu, 1) + "rupiah "
    End If
    'Baca bagian sen dan ditambah akhiran sen
    If sen > 0 Then
```

```
        baca = baca + ratus(sen, 0) + "sen"
    End If
End If
terbilang = UCase(Left(baca, 1)) & LCase(Mid(baca, 2))
End Function

Function ratus(x As Currency, posisi As Integer) As String
Dim a100 As Integer, a10 As Integer, a1 As Integer
Dim baca As String
a100 = Int(x * 0.01)
a10 = Int((x - a100 * 100) * 0.1)
a1 = Int(x - a100 * 100 - a10 * 10)
'Baca Bagian Ratus
If a100 = 1 Then
    baca = "Seratus "
Else
    If a100 > 0 Then
        baca = angka(a100, 2) + "ratus "
    End If
End If
'Baca Bagian Puluh dan Satuan
If a10 = 1 Then
    baca = baca + angka(a10 * 10 + a1, 2)
Else
    If a10 > 0 Then
        baca = baca + angka(a10, 2) + "puluh "
    End If
    If a1 > 0 Then
        If posisi = 2 And a100 = 0 And a10 = 0 Then
            baca = baca + angka(a1, 1)
        Else
            baca = baca + angka(a1, 2)
        End If
    End If
End If
ratus = baca
End Function

Function angka(x As Integer, posisi As Integer)
Select Case x
    Case 0: angka = "Nol"
    Case 1:
        If posisi = 2 Then
            angka = "Satu "
        Else
            angka = "Se"
        End If
    Case 2: angka = "Dua "
    Case 3: angka = "Tiga "
    Case 4: angka = "Empat "
    Case 5: angka = "Lima "
    Case 6: angka = "Enam "
    Case 7: angka = "Tujuh "
    Case 8: angka = "Delapan "
    Case 9: angka = "Sembilan "
    Case 10: angka = "Sepuluh "
    Case 11: angka = "Sebelas "
```

```
Case 12: angka = "Duabelas "  
Case 13: angka = "Tigabelas "  
Case 14: angka = "Empatbelas "  
Case 15: angka = "Limabelas "  
Case 16: angka = "Enambelas "  
Case 17: angka = "Tujuhbelas "  
Case 18: angka = "Delapanbelas "  
Case 19: angka = "Sembilanbelas "  
End Select  
End Function
```

Ok, akhirnya segala sesuatu menjadi sederhana kalau kita telah memahami konsep dasar pemecahan masalahnya.

Membuat Program Mengeja Bilangan

Function SpellDigit(x)

```
Dim cRet As String
cRet = ""
Select Case x
Case 0: cRet = " ZERO"
Case 1: cRet = " ONE"
Case 2: cRet = " TWO"
Case 3: cRet = " THREE"
Case 4: cRet = " FOUR"
Case 5: cRet = " FIVE"
Case 6: cRet = " SIX"
Case 7: cRet = " SEVEN"
Case 8: cRet = " EIGHT"
Case 9: cRet = " NINE"
Case 10: cRet = " TEN"
Case 11: cRet = " ELEVEN"
Case 12: cRet = " TWELVE"
Case 13: cRet = " THIRTEEN"
Case 14: cRet = " FOURTEEN"
Case 15: cRet = " FIFTEEN"
Case 16: cRet = " SIXTEEN"
Case 17: cRet = " SEVENTEEN"
Case 18: cRet = " EIGHTEEN"
Case 19: cRet = " NINETEEN"
Case 20: cRet = " TWENTY"
Case 30: cRet = " THIRTY"
Case 40: cRet = " FORTY"
Case 50: cRet = " FIFTY"
Case 60: cRet = " SIXTY"
Case 70: cRet = " SEVENTY"
Case 80: cRet = " EIGHTY"
Case 90: cRet = " NINETY"
Case 100: cRet = " ONE HUNDRED"
Case 200: cRet = " TWO HUNDRED"
Case 300: cRet = " THREE HUNDRED"
Case 400: cRet = " FOUR HUNDRED"
Case 500: cRet = " FIVE HUNDRED"
Case 600: cRet = " SIX HUNDRED"
Case 700: cRet = " SEVEN HUNDRED"
Case 800: cRet = " EIGHT HUNDRED"
Case 900: cRet = " NINE HUNDRED"
End Select
SpellDigit = cRet
End Function
```

Function SpellHundred(x)

```
Dim cRet As String
Dim n100 As Integer
Dim n10 As Integer
Dim n1 As Integer
cRet = ""
n100 = Int(x / 100) * 100
n10 = Int((x - n100) / 10) * 10
```

```

n1 = (x - n100 - n10)
If n100 > 0 Then
    cRet = SpellDigit(n100)
End If
If n10 > 0 Then
    If n10 = 10 Then
        cRet = cRet & SpellDigit(n10 + n1)
    Else
        cRet = cRet & SpellDigit(n10)
    End If
End If
If n1 > 0 And n10 <> 10 Then
    cRet = cRet & SpellDigit(n1)
End If
SpellHundred = cRet
End Function

```

Public Function SpellAmount(x)

```

Dim cRet As String
Dim n1000000 As Long
Dim n1000 As Long
Dim n1 As Integer
Dim n0 As Integer
On Error Resume Next
cRet = ""
n1000000 = Int(x / 1000000) * 1000000
n1000 = Int((x - n1000000) / 1000) * 1000
n1 = Int(x - n1000000 - n1000)
n0 = (x - n1000000 - n1000 - n1) * 100
If n1000000 > 0 Then
    cRet = SpellHundred(n1000000 / 1000000) & " MILLION"
End If
If n1000 > 0 Then
    cRet = cRet & SpellHundred(n1000 / 1000) & " THOUSAND"
End If
If n1 > 0 Then
    cRet = cRet & SpellHundred(n1)
End If
If n0 > 0 Then
    cRet = cRet & " AND CENTS" & SpellHundred(n0)
End If
SpellAmount = cRet & " ONLY"
End Function

```