
Tutorial

Microsoft Visual Basic 6.0

(Rangkuman Indonesian Programmer - Indoprogram)

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 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.

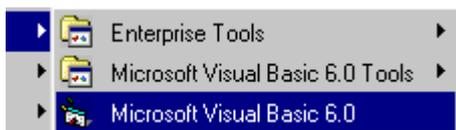
Pada pemrograman Visual, pengembangan aplikasi dimulai dengan pembentukan *user interface*, kemudian *mengatur properti dari objek-objek* yang digunakan dalam user interface, dan baru dilakukan *penulisan kode program* untuk menangani kejadian-kejadian (event). Tahap pengembangan aplikasi demikian dikenal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



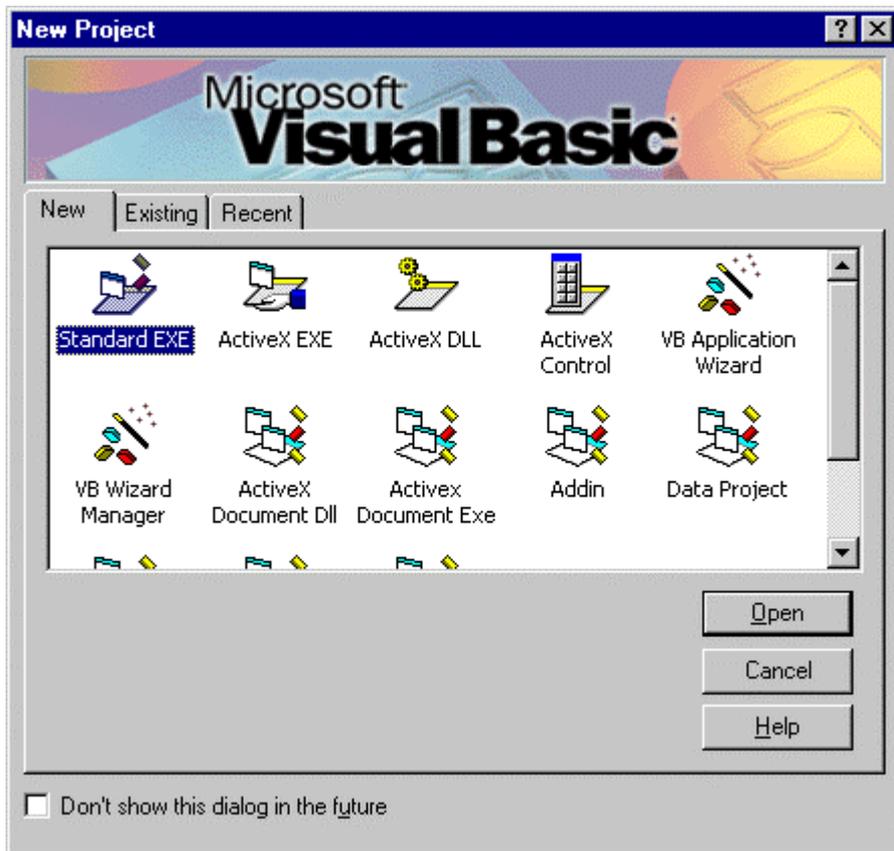
Gambar 1-1. Mengaktifkan IDE Visual Basic 6.0

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**.

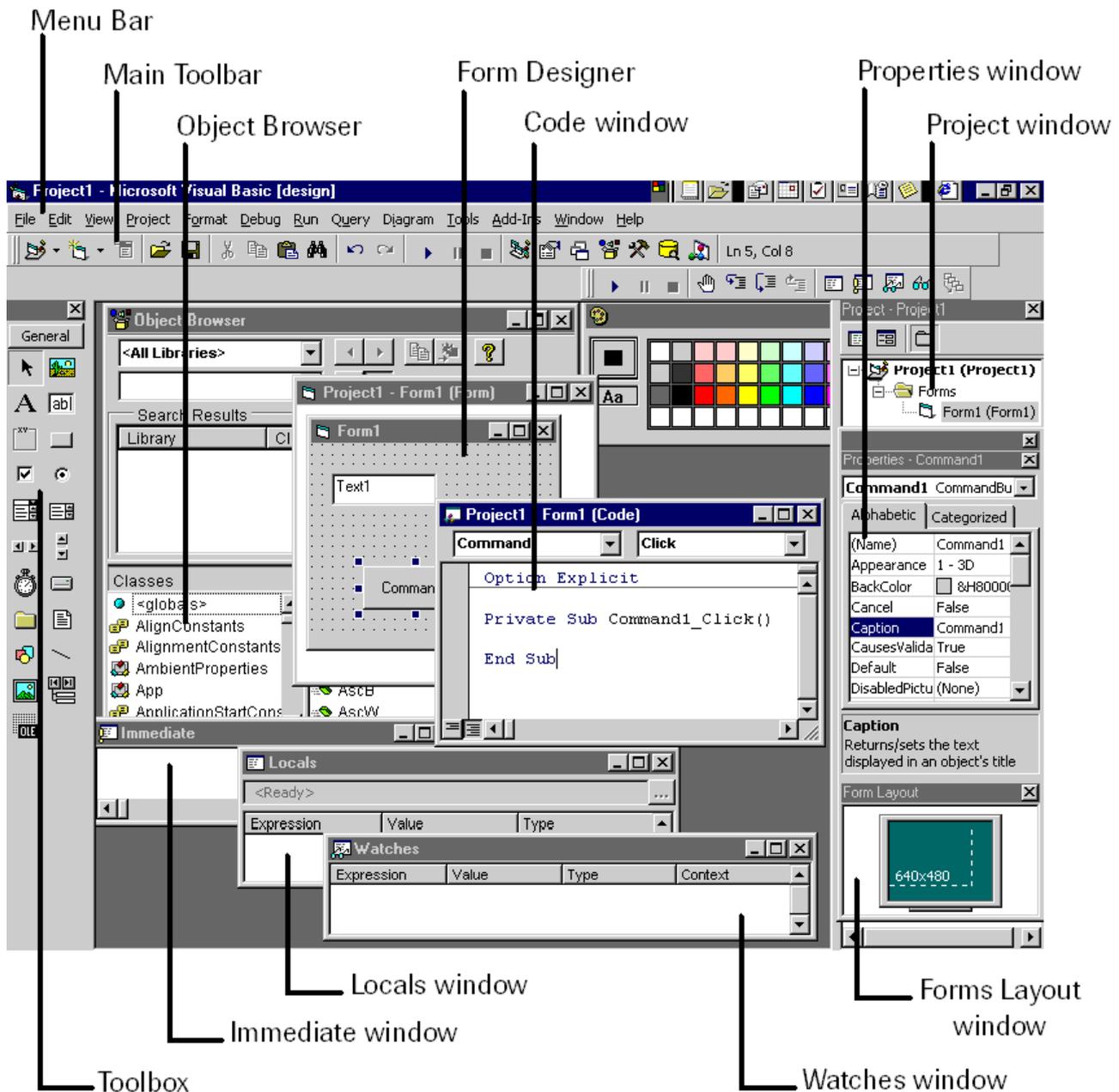


Gambar 1-2. *Dialog box New Project ditampilkan sesaat anda menjalankan IDE Visual Basic 6.*

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.

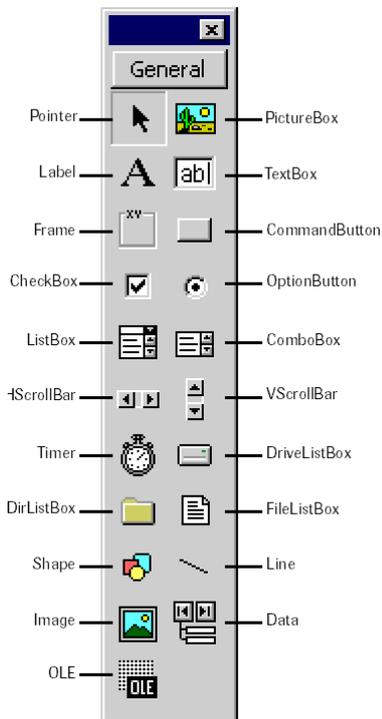
- *Jendela Code*, merupakan tempat bagi anda untuk menulis koding. Anda dapat menampilkan jendela ini dengan menggunakan kombinasi Shift-F7.
- *Jendela Properties*, merupakan daftar properti-properti object yang sedang terpilih. Sebagai contohnya anda dapat mengubah warna tulisan (foreground) dan warna 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 Pallete
- 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 :

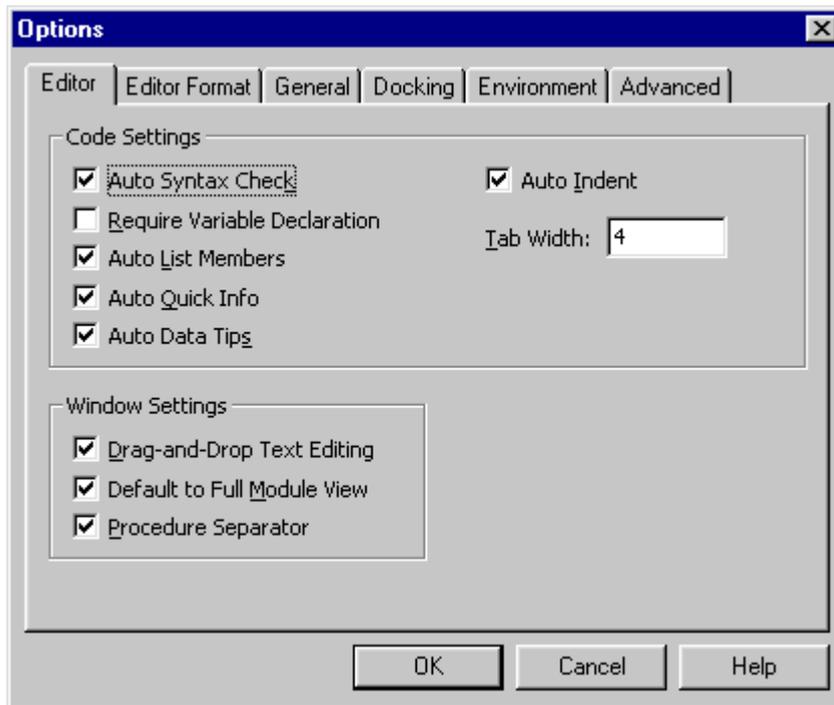
- *Pointer* bukan merupakan suatu kontrol; gunakan icon ini ketika anda ingin memilih kontrol yang sudah berada pada form.
- *PictureBox* adalah kontrol yang digunakan untuk menampilkan image dengan format: BMP, DIB (bitmap), ICO (icon), CUR (cursor), WMF (metafile), EMF (enhanced metafile), GIF, dan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

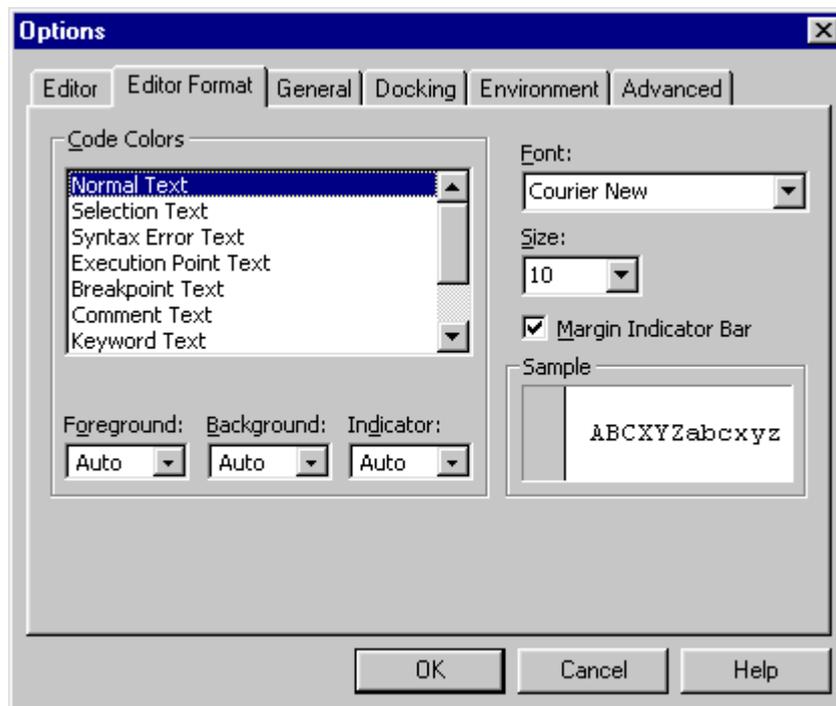
Mengatur Lingkungan Kerja Visual Basic

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 :

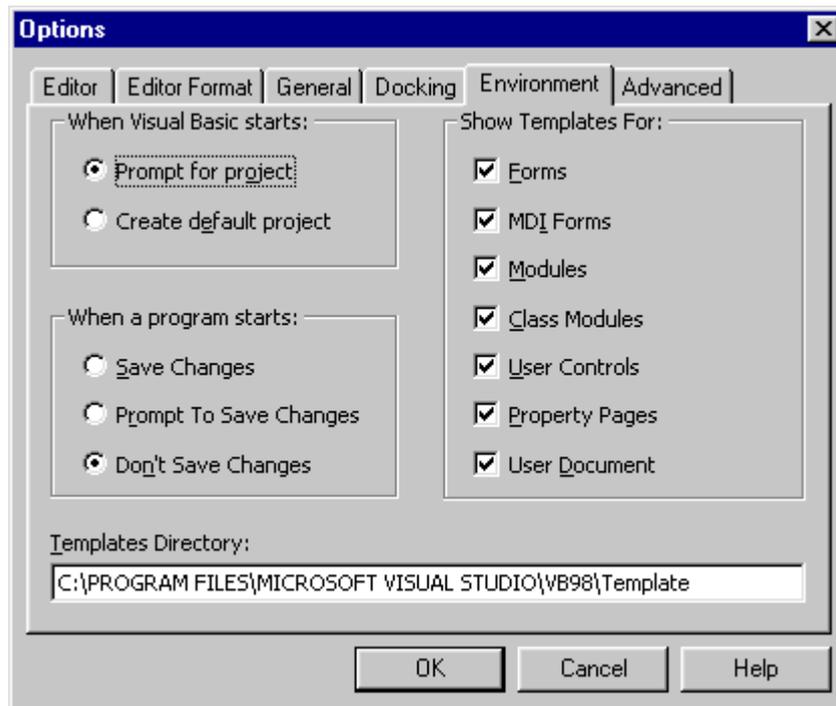
- **Mengatur Editor**



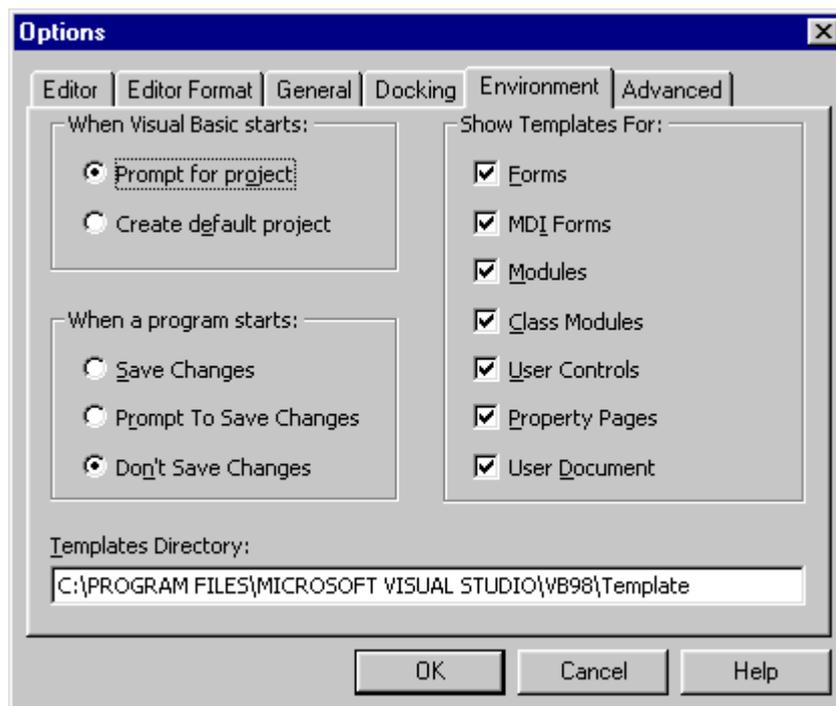
- **Mengatur Format Editor**



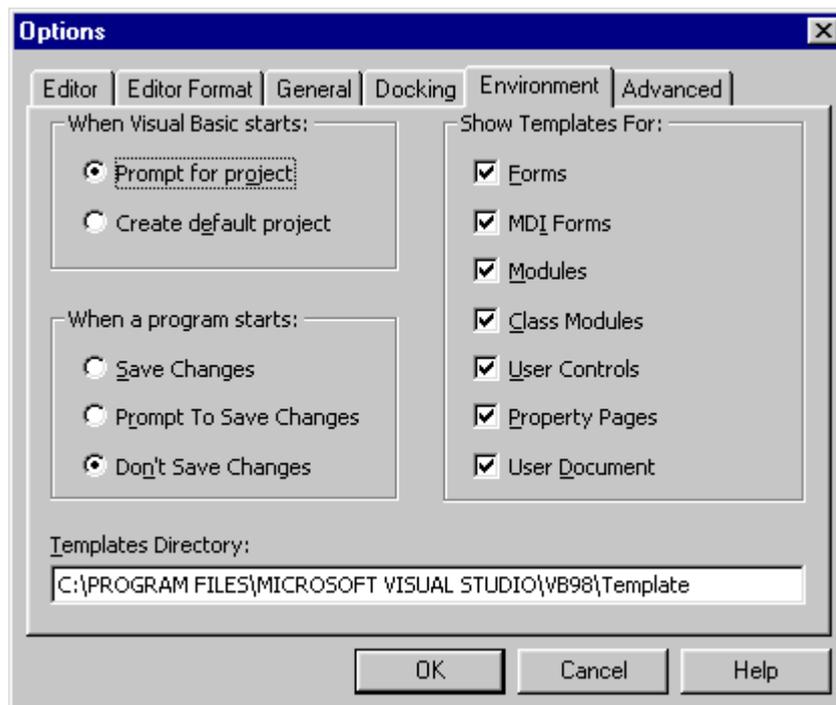
- **Mengatur hal-hal yang General**



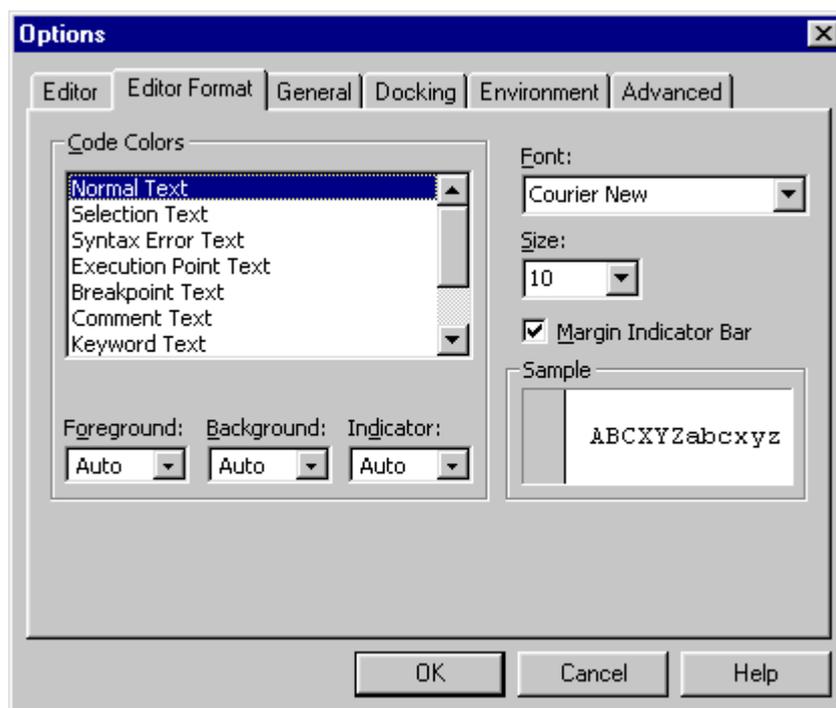
- **Mengatur Docking Jendela**



- **Mengatur Environment**



- **Mengatur SDI Development Environment atau MDI**



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.

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

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*

Operator Like

Salah satu operator yang menarik untuk dibahas adalah operator like, 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*"

Deklarasi Variabel

Visual Basic memungkinkan kita untuk menggunakan variabel tanpa deklarasi. Tetapi hal ini adalah kurang baik untuk program 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]

Memahami Objek pada Visual Basic

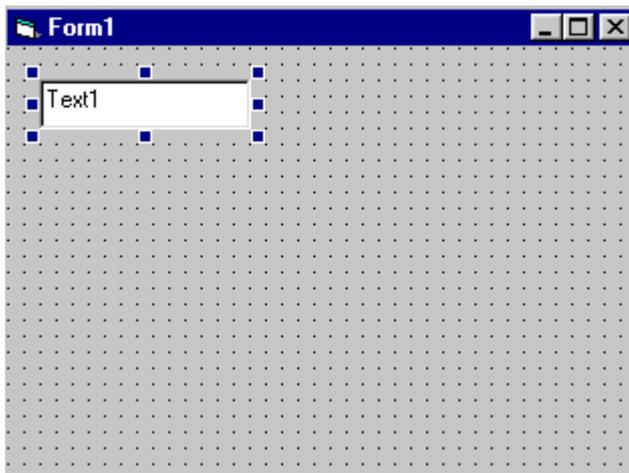
Apa yang dimaksud dengan Objek

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.

Dari mana datangnya Objek

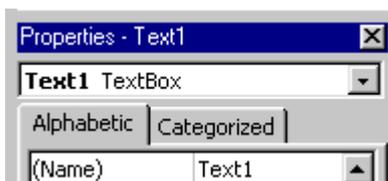
Setiap objek pada Visual Basic didefinisikan oleh suatu class. Untuk memahami hubungan antara objek dengan class-nya, perhatikan contoh berikut :

- 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

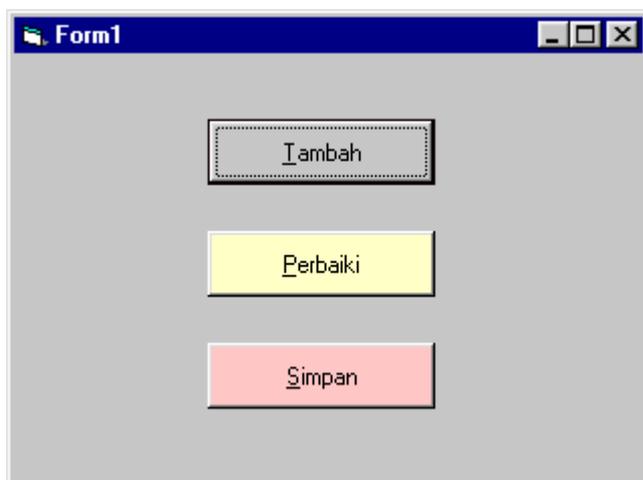
Misalnya anda menanamkan sebuah textbox diatas form1, coba perhatikan jendela properti kontrol tersebut akan menampilkan Text1 (nama dari kontrol textbox tersebut) dan TextBox (nama dari class kontrol tersebut).



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

Bekerja dengan objek

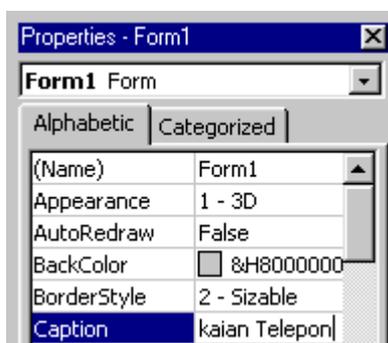
Objek-objek Visual Basic memiliki properti-properti, metode-metode, dan event-event. Pada Visual Basic, suatu data dari objek (setting dan atribut) disebut sebagai properti, sedangkan berbagai subrutin yang dapat beroperasi pada objek tersebut disebut sebagai metodenya. Dan suatu event adalah aksi yang 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 menuliskannya sebagai berikut ::

Objek.Metode

Contoh :

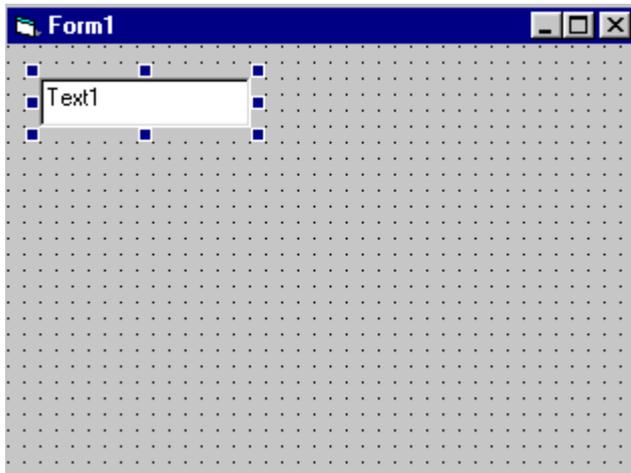
```
Private Sub cmdBuka_Click()  
frmPemakai.show  
End Sub
```

Beberapa properti yang umum

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.



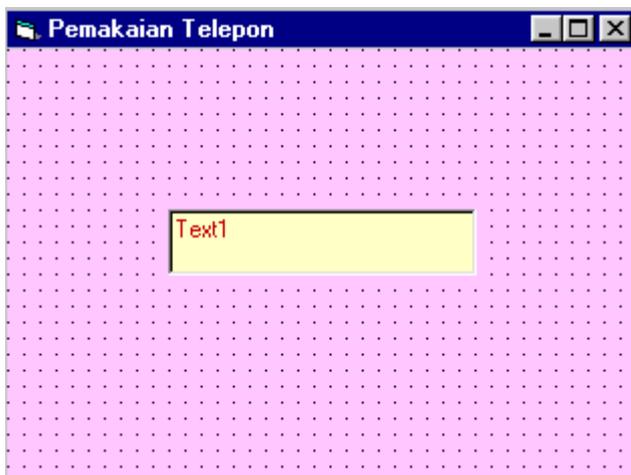
Gambar 2-5, Form1 merupakan kontainer bagi Text1

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.

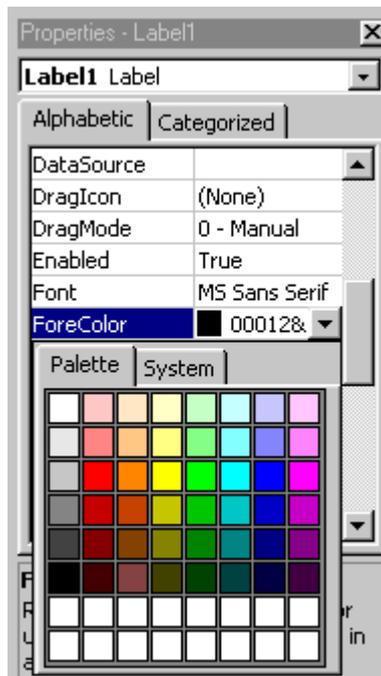
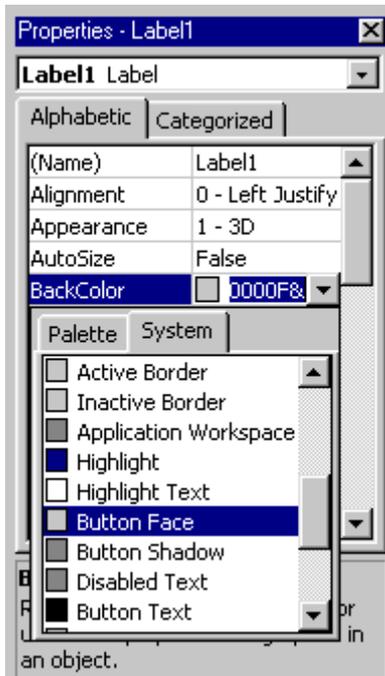
Properti *ForeColor* dan *BackColor*

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 Backcolornya hanya berfungsi jika anda mengganti properti style menjadi 1 - Graphical.



Gambar 2-6, form dengan pemakaian backcolor 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*

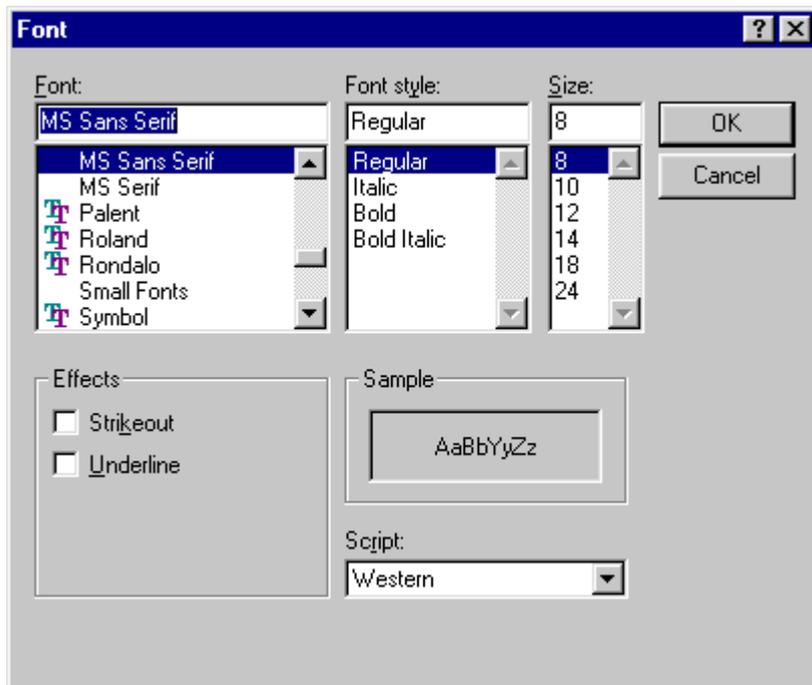
Pada waktu desain, anda dapat menentukan properti Font dengan dialog box, tetapi pada saat runtime, hal tersebut harus dilakukan dengan properti *Name*, *Size*, *Bold*, *Italic*, *Underline*, and *Strikethrough*, karena sebenarnya Font adalah suatu objek campuran.

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.

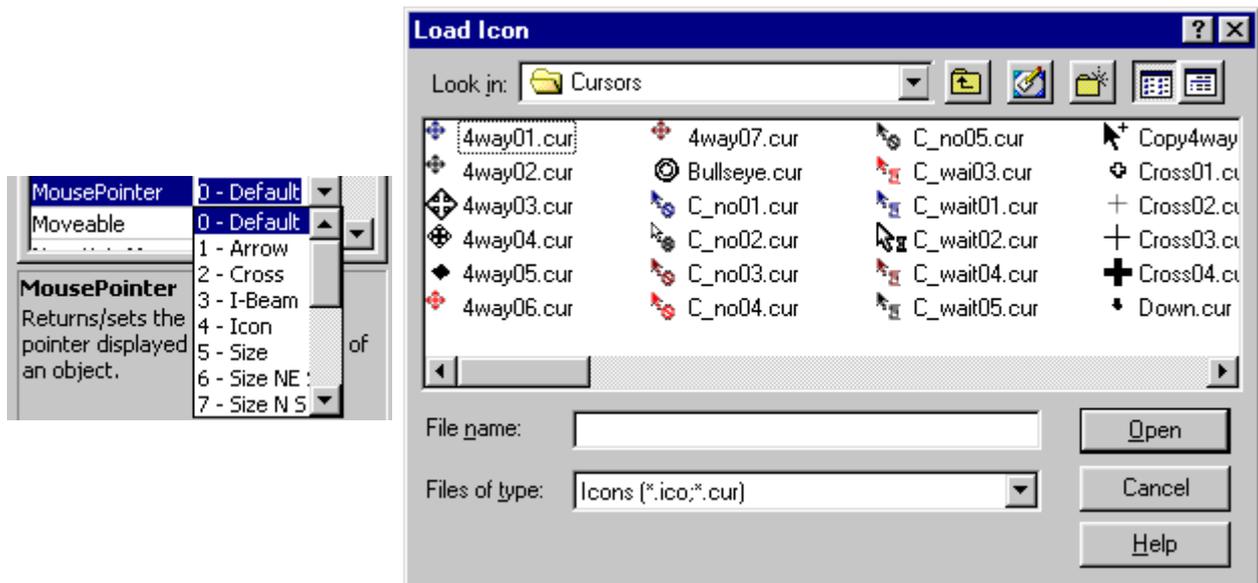
Properti *MousePointer* dan *MouseIcon*

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  
    Label1.Caption = CStr(i)  
    Label1.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
```

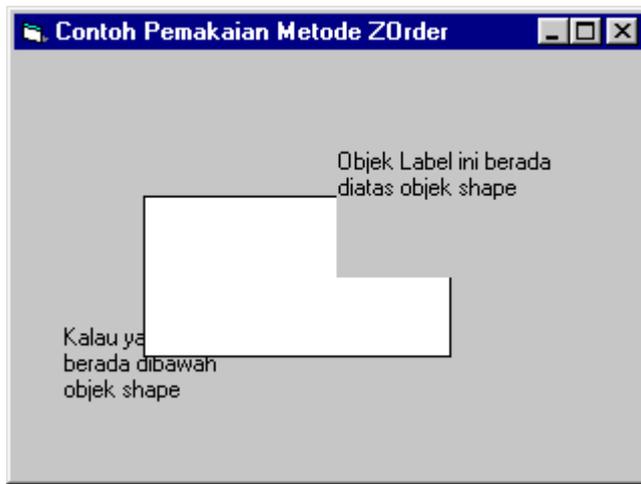
Metode ZOrder

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 :

```
Label1.ZOrder           'Memindahkan ke atas  
Label1.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.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, 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.

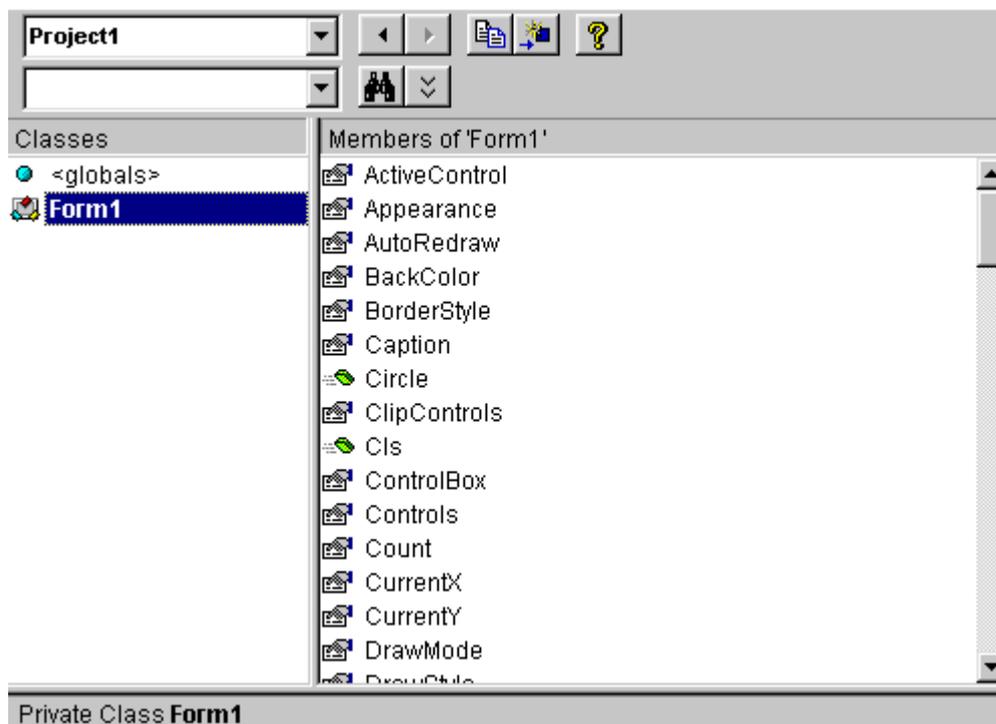
Event *KeyPress*, *KeyDown*, dan *KeyUp*

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, hanay membangkitkan event *KeyDown* dan *KeyUp*.

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

Objek Browser

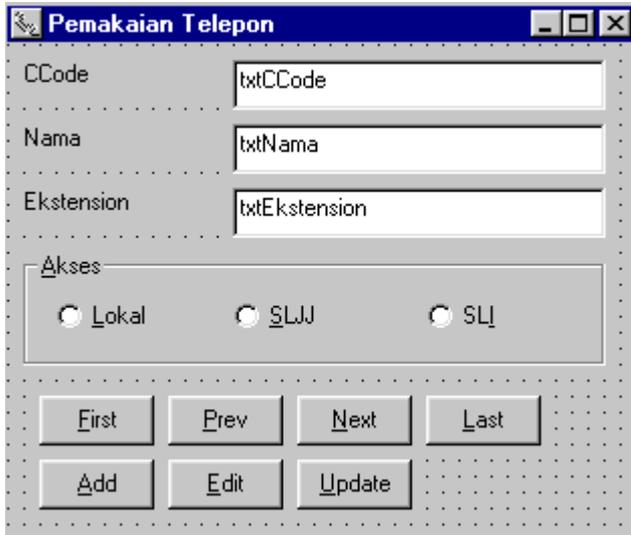
Anda dapat melihat properti-properti, metode-metode, dan event-event pada suatu objek dengan menggunakan fasilitas Objek Browser yang dapat diaktifkan dengan menggunakan menu View, Object Browser.



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

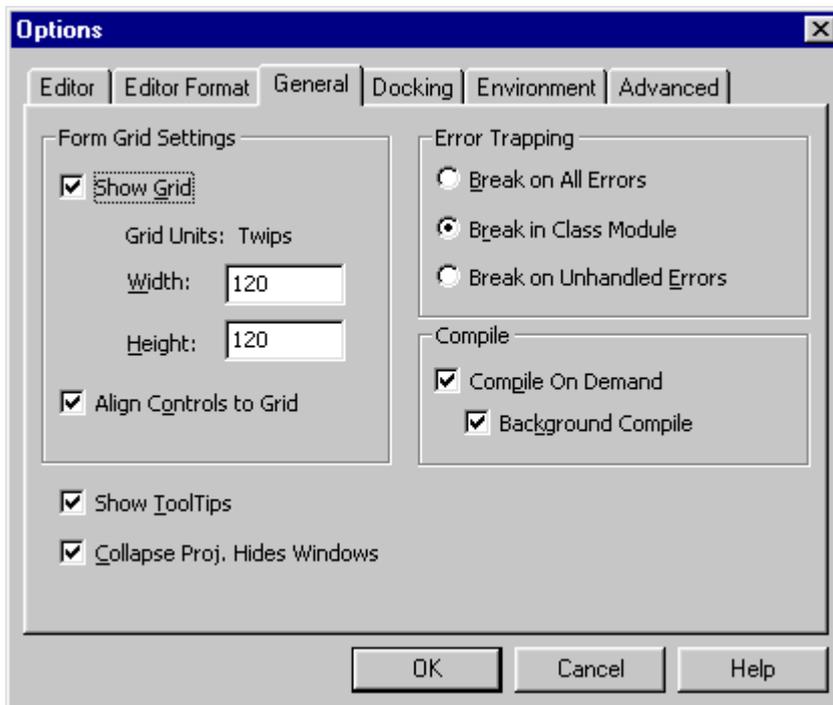
Mengenal Form

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.



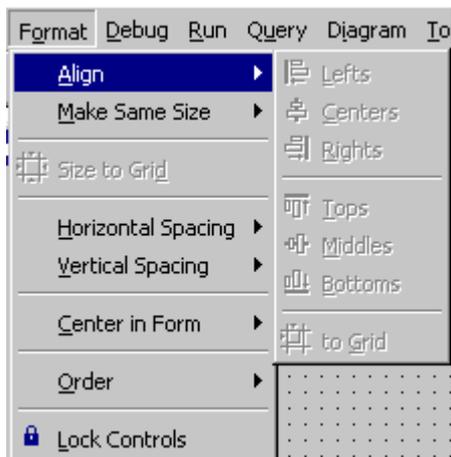
Gambar 2-12, user interface form dengan berbagai kontrol di atasnya

Untuk memudahkan 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

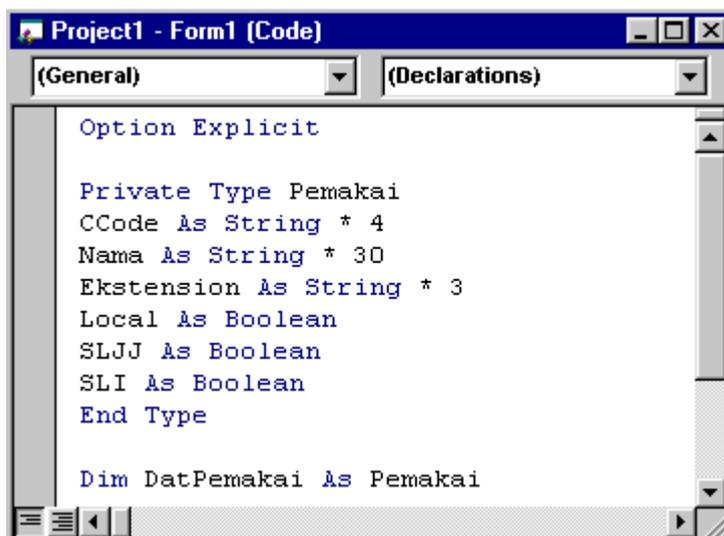


Gambar 2-14, menu format

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.

```

Project1 - Form1 (Code)
(General) Tampil
Sub Tampil()
  If Posisi > 0 Then
    Get #1, Posisi, DatPemakai
    txtCCode = DatPemakai.CCode
    txtNama = DatPemakai>Nama
    txtEkstension = DatPemakai.Ekstensi
    optLocal = DatPemakai.Local
    optSLJJ = DatPemakai.SLJJ
    optSLI = DatPemakai.SLI
  Else
    txtCCode = ""
    txtNama = ""
  End If
End Sub

```

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.

```

Project1 - Form1 (Code)
Form Load
Private Sub Form_Load()
  Open "C:\Pemakai.Dat" For Random As #1
  JlhRec = LOF(1) / Len(DatPemakai)
  If JlhRec > 0 Then
    Posisi = 1
  Else
    Posisi = 0
  End If
  Call Tampil
End Sub

```

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 menyetikkan 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.

Bahasa Visual Basic Application

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 deklarasi 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 control, forms, classes, dan module tidak 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 di suatu form, standard, atau class module, dari pada dalam suatu procedure, membuat variabel itu berlaku untuk semua procedure 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 menggunakannya.

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
```

<http://pcr.aksespraktis.net>

Tutorial Visual Basic

```
        [blokpernyataan-1]]  
[Case ekspresi  
        [blokpernyataan-2]]  
    ...  
[Case Else  
        [blokpernyataan-n]]
```

End Select

Kontrol Standard (bagian 1)

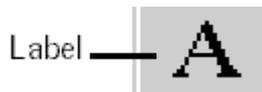


Kontrol standard terdapat pada semua versi Visual Basic, baik pada Learning Edition, Profesional Edition, maupun Enterprised Edition.

Kontrol-kontrol standard ini akan sering anda gunakan untuk pembentukan 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	Digunakan untuk menentukan bentuk border, apakah (0- None, 1 - Fixed Single) 
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

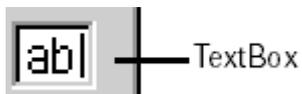
Adapun event-event yang efektif pada kontrol label adalah :

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)
DblClick	Event ini terjadi ketika pemakai melakukan double klik pada kontrol, event ini terjadi dengan urutan (MouseDown, MouseUp, Click, MouseDown, DblClick, 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 :

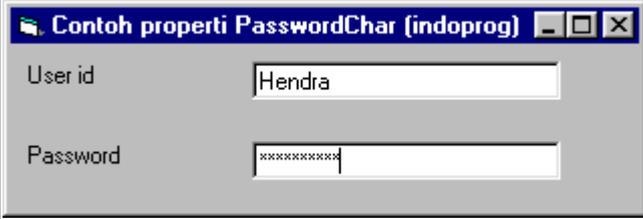
Metoda	Keterangan
Move	Metoda ini digunakan untuk memindahkan letak (koordinat Left, Top) dari kontrol label maupun ukurannya (ukuran Height, Width)
Refresh	Metoda ini digunakan untuk mencetak ulang kontrol label.
ZOrder	Metoda ini digunakan untuk menentukan order kontrol, apakah diatas atau dibawah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

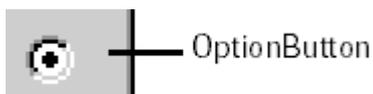
Adapun event-event yang efektif pada kontrol Textbox adalah :

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) If KeyAscii = 13 Then SendKeys "{Tab}" End If End Sub
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 menimandahkan fokus ke kontrol yang bersangkutan
ZOrder	sda

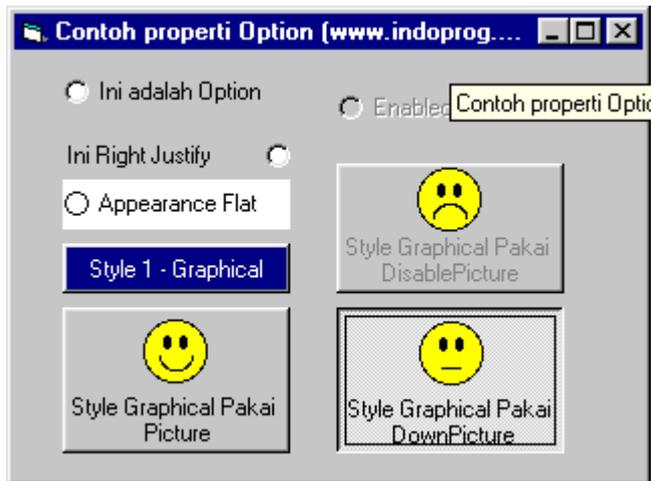
Option



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

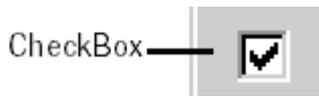
Adapun event-event yang efektif pada kontrol option adalah :

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

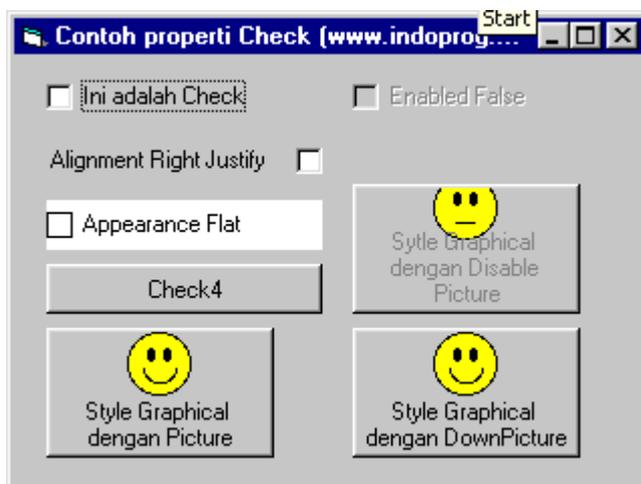
Adapun metoda yang efektif pada kontrol Option adalah :

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

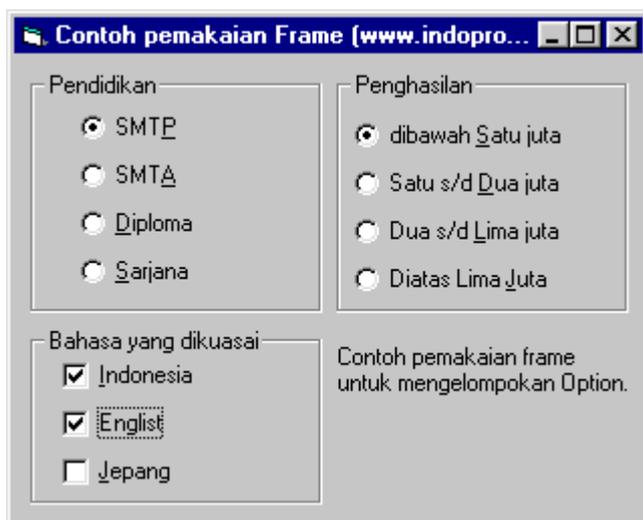
Frame



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.
Mouselcon	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

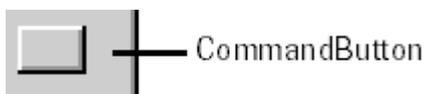
Adapun event-event yang efektif pada kontrol frame adalah :

Event	Keterangan
Click	sda
DbIclick	sda
MouseDown	sda
MouseMove	sda
MouseUp	sda

Adapun metoda yang efektif pada kontrol Frame adalah :

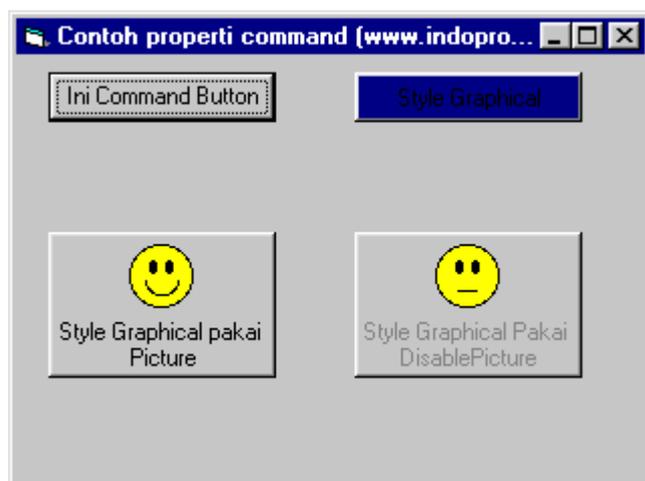
Metoda	Keterangan
Move	sda
Refresh	sda
ZOrder	sda

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 apakah 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

Visual Basic Application

Deklarasi Konstanta

Anda sering menjumpai suatu code yang mengandung nilai yang berulang-ulang dimunculkan, 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 dia 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. Menset 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
```

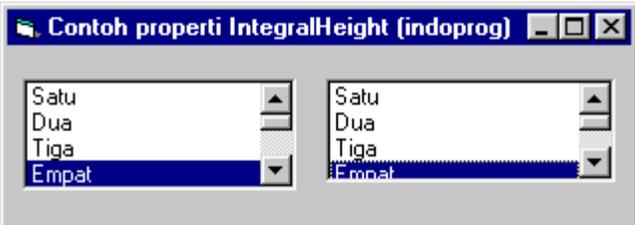
Statement Exit Do bekerja pada semua versi syntax Do loop.

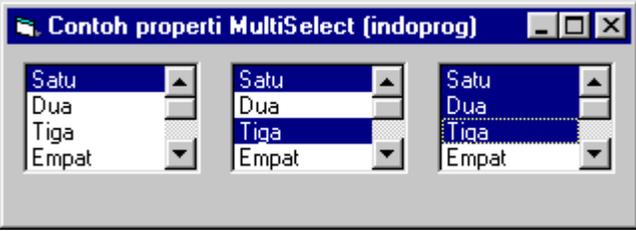
Exit For dan Exit Do adalah kadang-kadang perlu untuk keluar dari suatu loop dengan segera, tanpa melakukan iterasi yang berikutnya.

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 memasuki 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. Misalnda 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)  
        End If  
    Next i  
End Sub
```

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

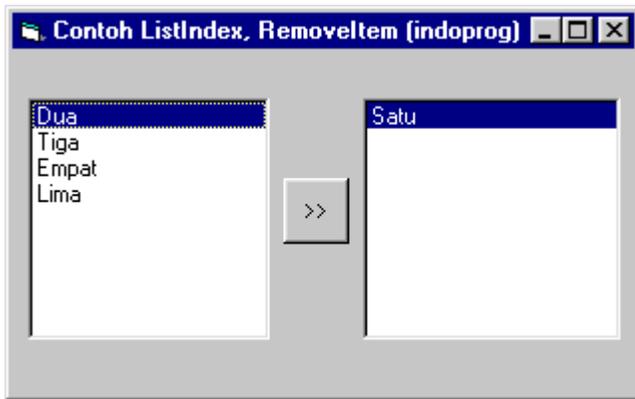
Mendapatkan nomor posisi item yang sedang difokus

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.

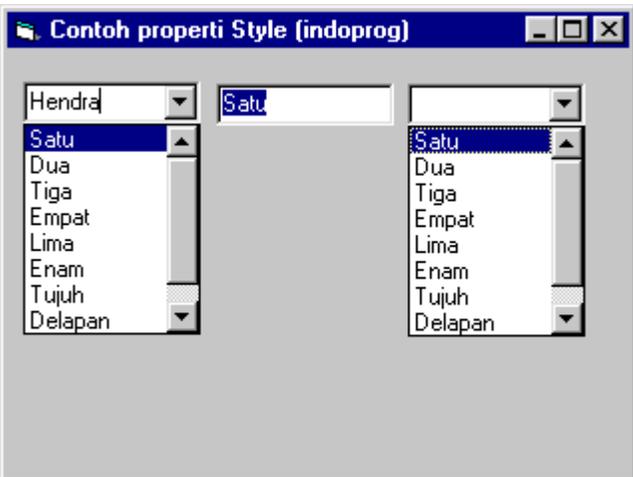
Adapun event-event yang efektif pada Listbox adalah sebagai berikut :

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 Style = Checkbox
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.
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 mengerjakan 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	<p>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.</p> 
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

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 :

- 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.
- 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 namaprocedur (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 programmer, 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.

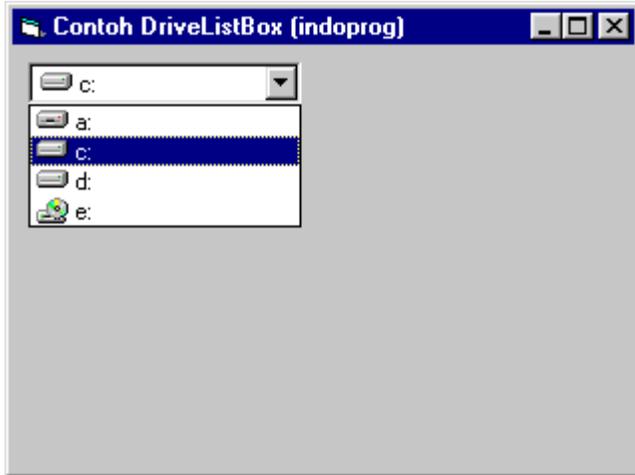
Contoh Fungsi ciptaan untuk mengembalikan nama bulan dari suatu tanggal dalam bahasa Indonesia.

```
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
```

Kontrol Standard (bagian 4)

DriveListBox

DriveListBox dapat digunakan untuk menampilkan drive yang terdapat pada sistem komputer



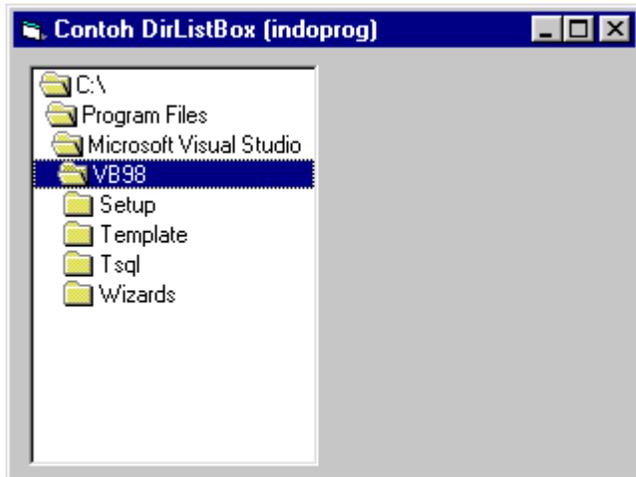
Properti	Fungsi
BackColor	Digunakan untuk menentukan warna latarbelakang kontrol
CauseValidation	Digunakan untuk menentukan apakah event Validate di aktifkan ketika kontrol akan kehilangan fokus (False, True)
DragMode	Menentukan modus drag-n-drop (0 - manual, 1 - automatic)
DragIcon	Digunakan untuk menentukan 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.



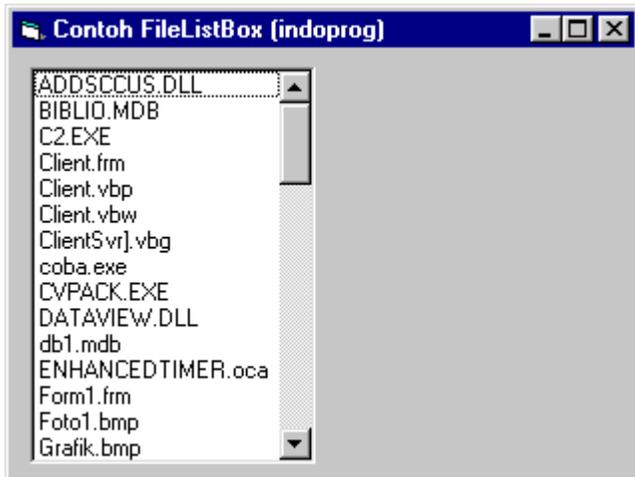
Properti	Fungsi
Appearance	Digunakan untuk menentukan penampilan dari kontrol (0 - Flat, 1 - 3D)
BackColor	Digunakan untuk menentukan warna latarbelakang kontrol
CauseValidation	Digunakan untuk menentukan apakah event Validate di aktifkan ketika kontrol akan kehilangan fokus (False, True)
DragMode	Menentukan 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	Diunakan 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)

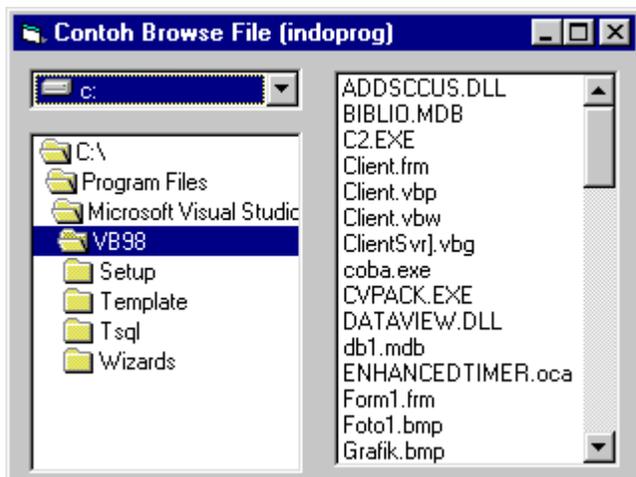
Untuk membatasi file berdasarkan atributnya, anda dapat menggunakan properti Archive, Hidden, Normal, ReadOnly dan System, dengan menentukan masing-masing menjadi True atau False.

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
```

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 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 :

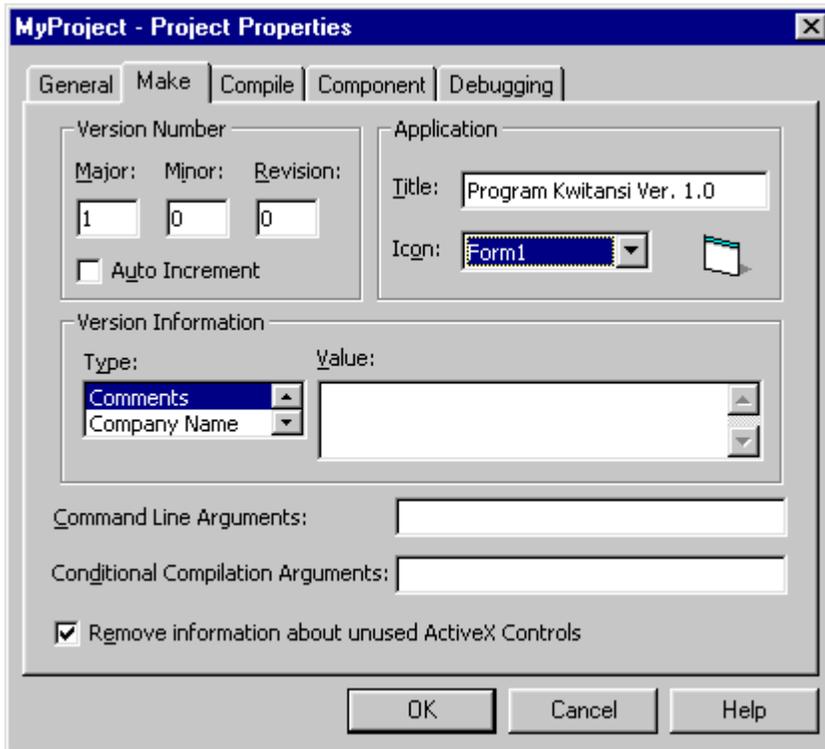
```
Sub Main()  
App.TaskVisible = False  
End Sub
```

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 :

```
' Menggambil 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.

Kontrol Standard (bagian 3)

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.

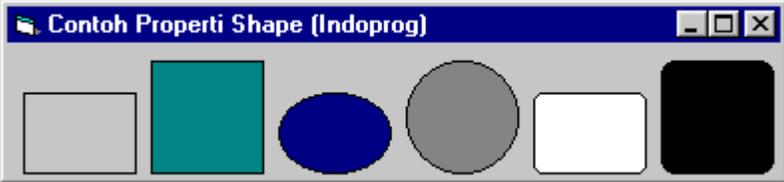
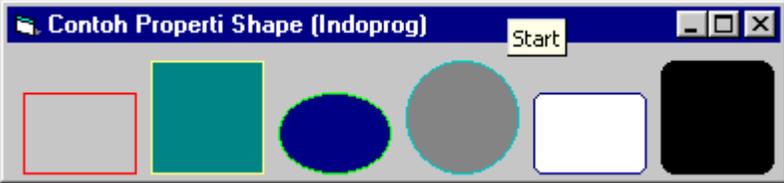
Contoh :

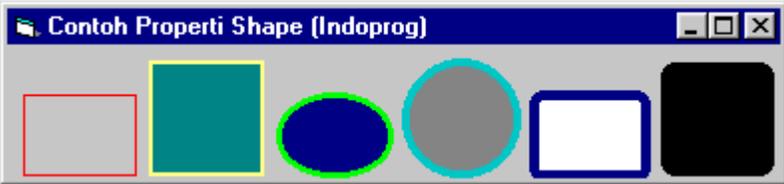
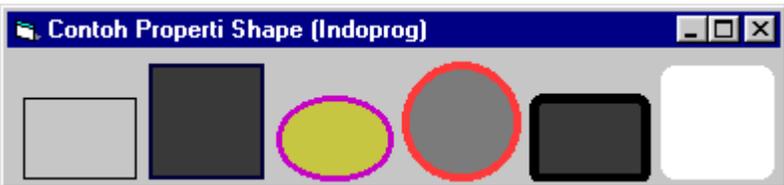
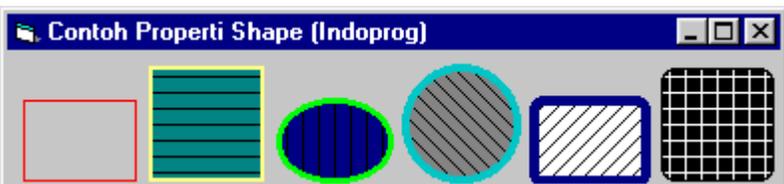
```
Private Sub VScroll_Change()  
  ImgGambar.Top = -VScroll.Value      'Berlawanan arah  
End Sub  
Private Sub HScroll_Change()  
  ImgGambar.Left = -HScroll.Value     'Berlawanan arah  
End Sub
```

Event Scroll, akan dibangkitkan ketika pemakai melakukan pengeseran terhadap Bar dengan menggunakan drag pada tombol kiri mouse, jadi Event Scroll akan terjadi ketika pemakai melakukan pengeseran 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	<p>Menentukan warna latarbelakang dari suatu Shape. Properti ini efektif jika setting Backstyle dari Shape adalah (1 - Opaque)</p> 
BackStyle	<p>Menentukan jenis latarbelakang dari suatu Shape, apakah (0 - Transparent, 1 - Opaque)</p>
BorderColor	<p>Menentukan warna bingkai dari suatu Shape. Properti ini efektif jika setting BorderStyle bukan (0 - Transparent)</p> 

BorderStyle	Menentukan jenis garis border yang digunakan. (0 - Transparent, 1 - Solid, 2 - Dash, 3 - Dot, 4 - Dash Dot, 5 - Dash Dot Dot, dst) 
BorderWidth	Menentukan ukuran ketebalan garis border. 
DrawMode	Menentukan jenis Drawmode yang digunakan. defaultnya adalah (13 - Copy pen) 
FillColor	Menentukan warna yang digunakan untuk menggambarkan pola isinya yang ditentukan pada properti FillStyle. 
FillStyle	Menentukan jenis arsiran yang digunakan untuk isian kontrol Shape. Lihat gambar pada properti FillColor.
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.
Shape	Digunakan untuk menentukan bentuk Shape (0 - Rectangle, 1 - Square, 2 - Oval, 3 - Circle, 4 - Rounded Rectangle, 5 - Rounded Square)
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.
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

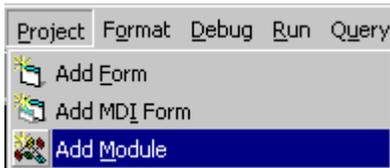
Event Timer adalah event yang dibangkitkan oleh kontrol timer berdasarkan interval waktu yang telah ditentukan.

Contoh :

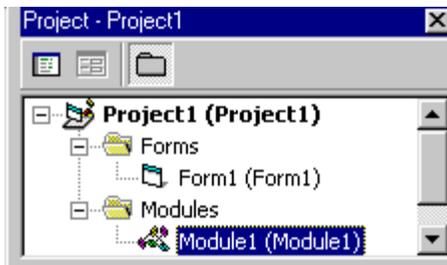
```
Private Sub Timer_Timer()  
Call AturJarum  
End Sub
```

Module Pada Visual Basic

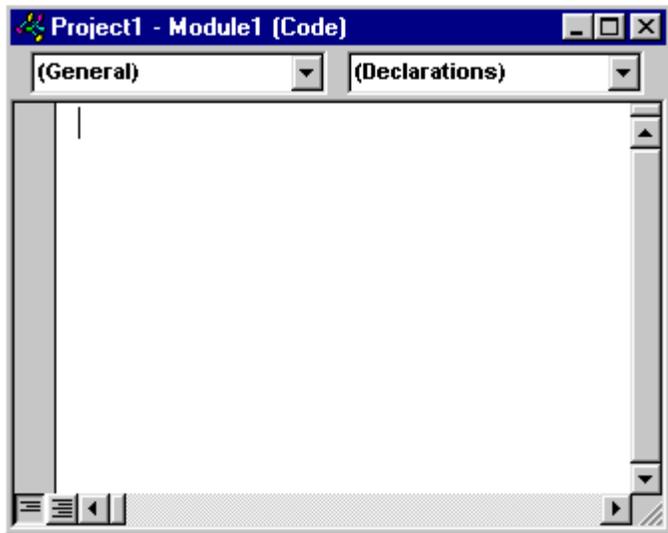
Visual Basic menyediakan module yang dapat digunakan untuk memuat fungsi,subrutin, konstanta, variabel dan type ciptaan yang akan disediakan bagi keseluruhan project. Untuk membuat module dalam project anda dapat menggunakan perintah **Project, Add Module**.



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.

MDI Form, Menu dan Objek Printer

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.

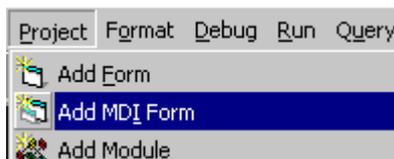
Sedangkan aplikasi seperti Microsoft Word menggunakan MDI, yaitu terdiri dari suatu MDIForm, dan didalamnya merupakan form-form anak (MDIchild).

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

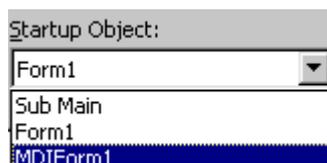
1. Membuat suatu MDI form
Dari menu Insert, Pilih Add MDI Form. (Suatu aplikasi hanya dapat terdiri dari satu MDI form)



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

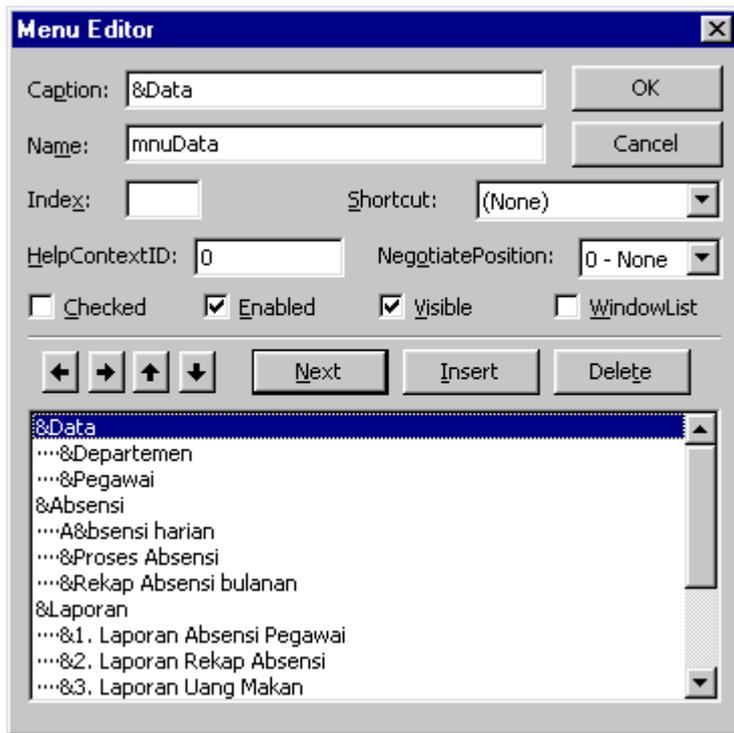
Private Sub mnuTileVertically_Click()
    Arrange vbTileVertical
End Sub

Private Sub mnuCascade_Click()
    Arrange vbCascade
End Sub

Private Sub mnuArrangeIcons_Click()
    Arrange vbArrangeIcons
End Sub
```

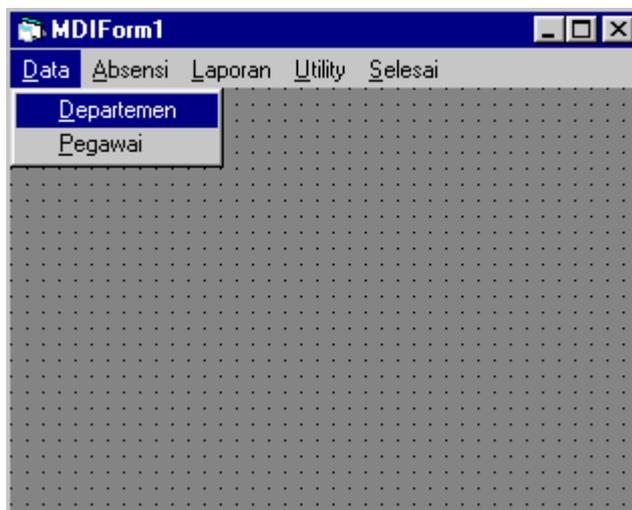
Membuat Menu Pada Visual Basic

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



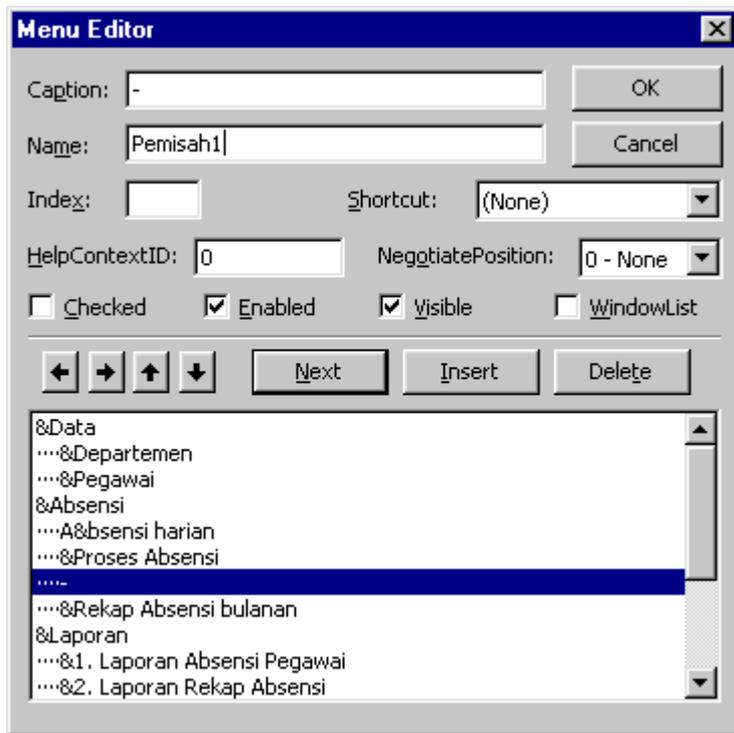
Pada dasarnya setiap item menu memiliki Caption dan sebuah Nama. Anda dapat membentuk Kunci Akses dengan menggunakan tanda & (ampersand) pada Caption dari menu tersebut. Untuk membuat menu anda cukup mengetikkan Caption dan Name, selanjutnya klik pada Next, dan ketikkan menu yang berikutnya, sampai selesai. 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 :

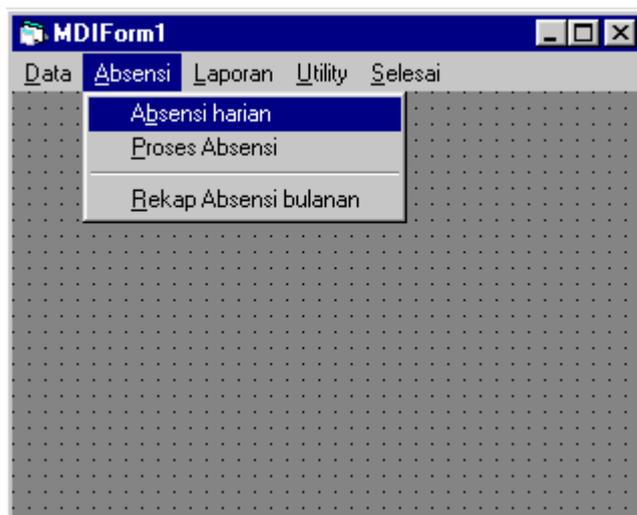


Membuat Garis Pemisah antar Menu

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



Sehingga akan menghasilkan menu sebagai berikut :



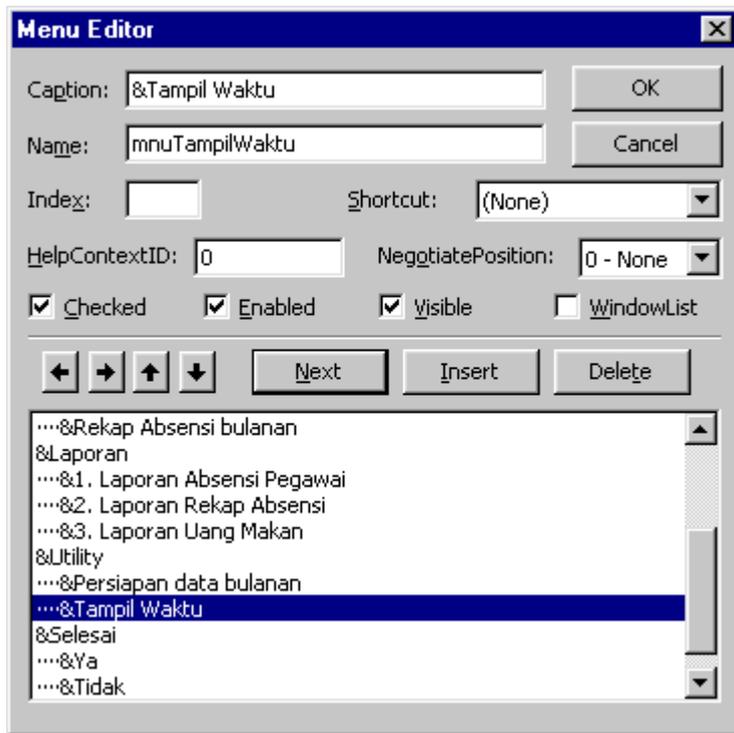
Melakukan Koding terhadap Menu

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

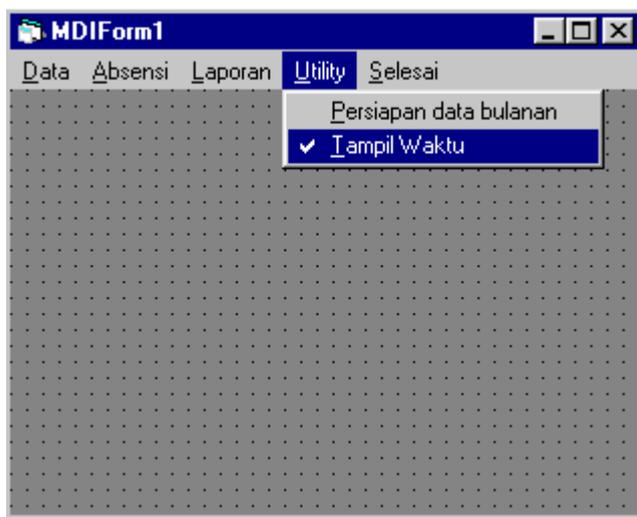
```
Private Sub mnuDataDepartemen_Click()
    frmDepartemen.Show
End Sub
```

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 :

```

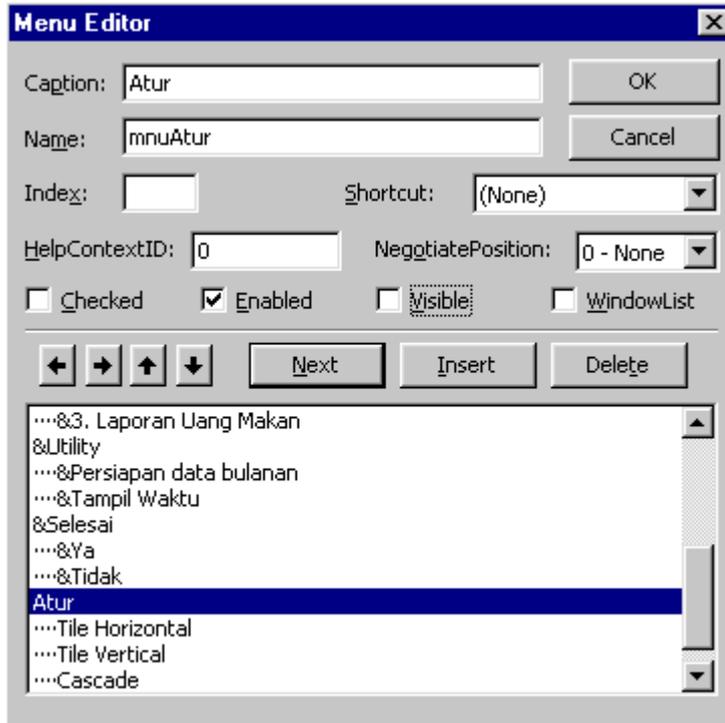
Private Sub mnuTampilWaktu_Click()
mnuTampilWaktu.Checked = Not mnuTampilWaktu.Checked
If mnuTampilWaktu.Checked Then
    frmWaktu.Show
Else
    Unload frmWaktu
End If
End Sub

```

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 Menunya, 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.



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 pencetakkan 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 pencetakkan yang sebenarnya, sedangkan KillDoc untuk membatalkan pencetakkan. 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 pencetakkan dapat menggunakan properti Copies.

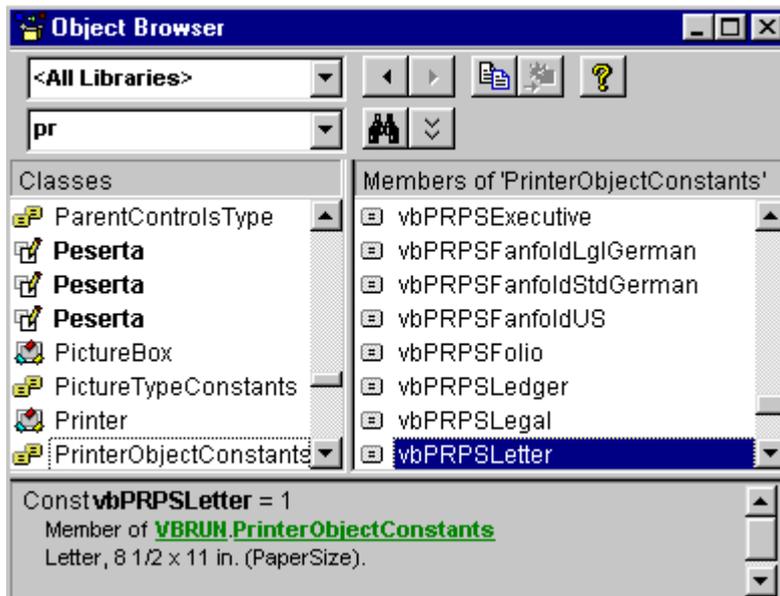
Contoh :

```
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 Indoprogram-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 Indoprogram-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
```