

Cepat Mahir Visual Basic 6.0

Krisna D. Octovhiana
mail4krisna@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Sehuruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Bab 1

Pendahuluan

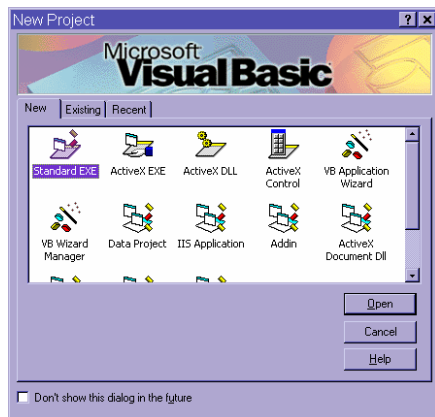
1.1 Mengenal Visual Basic (VB)

Visual Basic adalah salah satu bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Bahasa pemrograman Visual Basic, yang dikembangkan oleh Microsoft sejak tahun 1991, merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (*Beginner's All-purpose Symbolic Instruction Code*) yang dikembangkan pada era 1950-an. Visual Basic merupakan salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi Windows. Visual Basic merupakan salah satu bahasa pemrograman komputer yang mendukung object (*Object Oriented Programming* = OOP).

1.2 Mengenal *Integrated Development Environment (IDE) VB 6*

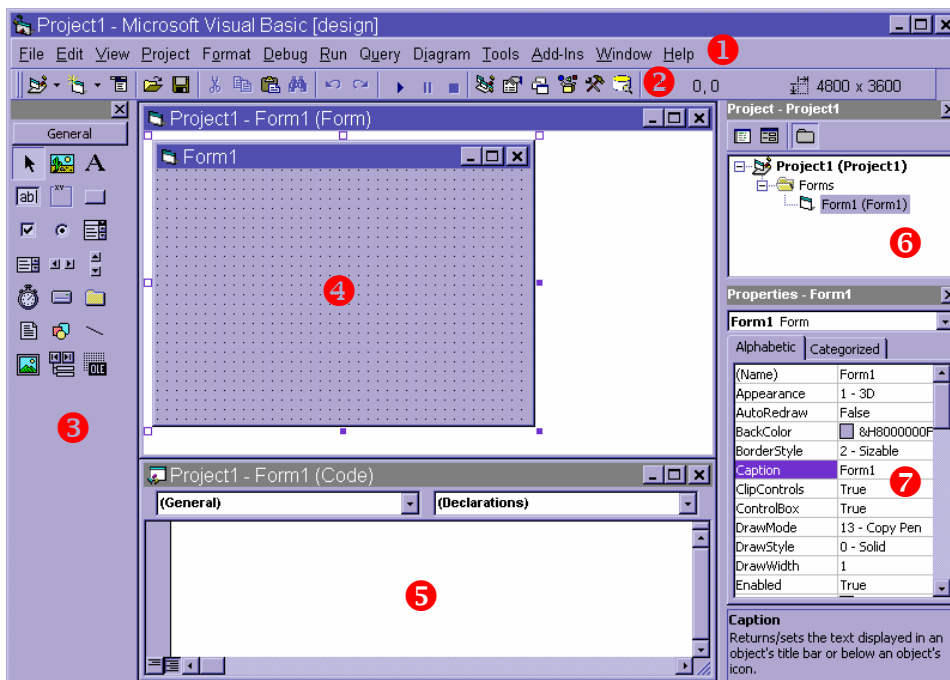
Aktifkan VB 6 melalui tombol Start > Programs > Microsoft Visual Studio 6.0 > Microsoft Visual Basic 6.0.

Tunggulah beberapa saat hingga muncul tampilan berikut :



Pilih Standard EXE dan klik tombol Open.

Anda akan melihat tampilan area kerja atau IDE VB 6. Kenali bagian-bagian utama di dalam IDE VB 6 berikut ini :




Keterangan :

❶ Menubar

❷ Toolbar


❸ Toolbox

Bila Toolbox tidak muncul klik tombol Toolbox () pada bagian Toolbar atau klik menu View > Toolbox.


❹ Jendela Form

Bila Jendela Form tidak muncul klik tombol View Object () pada bagian Project Explorer atau klik menu View > Object.

❺ Jendela Code

Bila Jendela Code tidak muncul klik tombol View Code () di pada bagian Project Explorer atau klik menu View > Code.

❻ Project Explorer

Bila Project Explorer tidak muncul klik tombol Project Explorer () pada bagian Toolbar atau klik menu View > Project Explorer.

❼ Jendela Properties

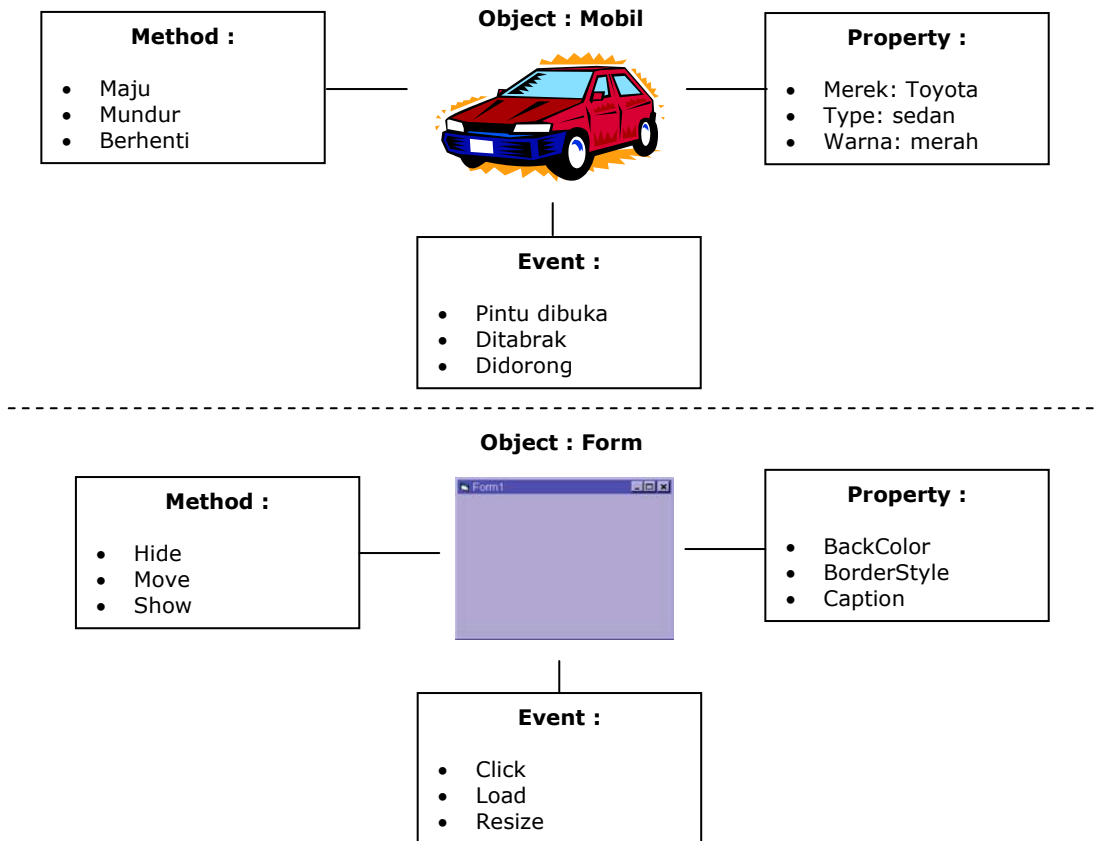
Bila Jendela Properties tidak muncul klik tombol Properties Window () pada bagian Toolbar atau klik menu View > Properties Window.

1.3 Memahami Istilah Object, Property, Method dan Event

Dalam pemrograman berbasis obyek (OOP), anda perlu memahami istilah object, property, method dan event sebagai berikut :

- Object** : komponen di dalam sebuah program
Property : karakteristik yang dimiliki object
Method : aksi yang dapat dilakukan oleh object
Event : kejadian yang dapat dialami oleh object

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



Implementasinya dalam sebuah aplikasi misalnya anda membuat form, maka form tersebut memiliki property, method, dan event. Sebagaimana pemrograman visual lain seperti Delphi dan Java, VB juga bersifat event driven programming. Artinya anda dapat menyisipkan kode program pada event yang dimiliki suatu obyek.

Cepat Mahir Visual Basic 6.0

Krisna D. Octovhiana
mail4krisna@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Sehuruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

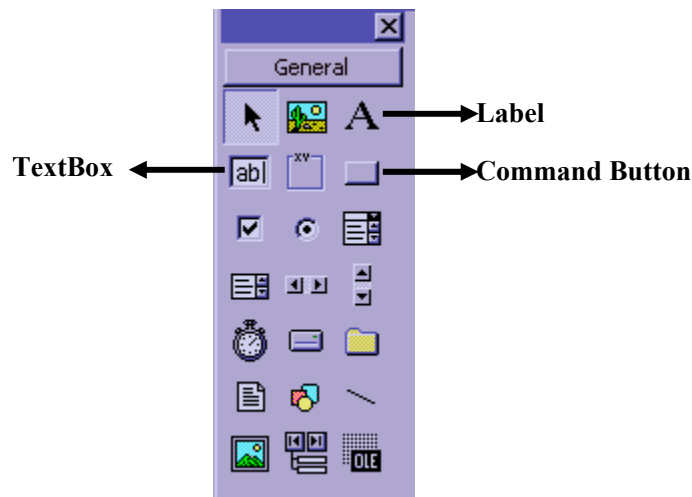
Bab 2

Menggunakan Event dan Property

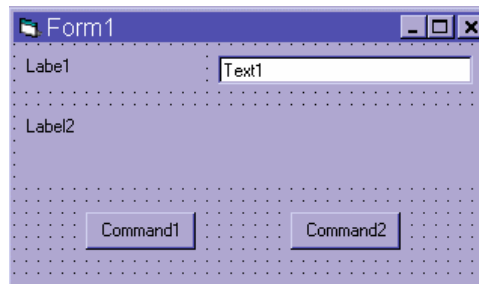
1.1 Membuat *User Interface* (UI)

Aktifkan VB 6 melalui tombol Start.

Kenali jenis-jenis komponen (*control*) yang ada di dalam bagian ToolBox :



Pada Jendela Form buatlah UI seperti ini :

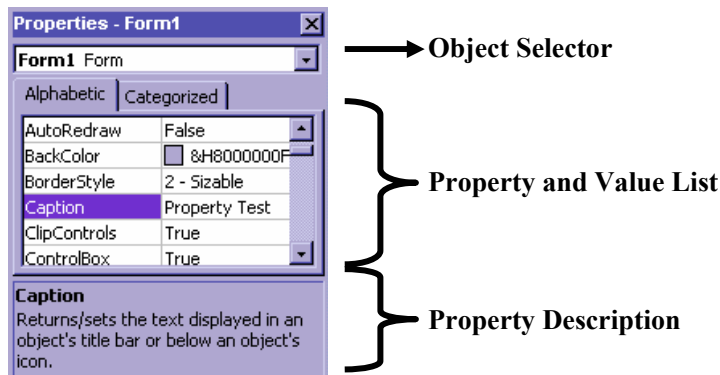


Cara membuatnya :

Klik komponen Label di dalam ToolBox kemudian “gambar” object Label1 di dalam Form dengan cara drag-drop. Lakukan langkah yang sama untuk object Label2. Object Text1 di-“gambar” menggunakan komponen TextBox, sedangkan object Command1 dan Command2 menggunakan komponen Command Button.

1.2 Mengatur Property Object

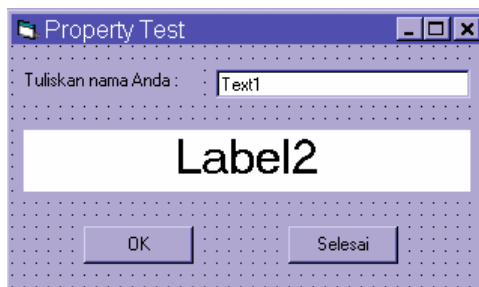
Kenali bagian-bagian di dalam Jendela Properties :



Melalui Jendela Properties atur property setiap object sebagai berikut :

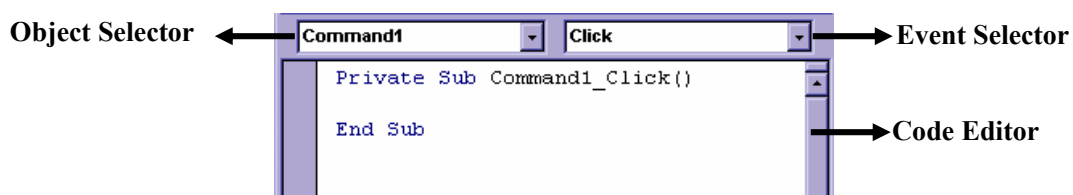
Object	Properties	Value
Form1	Caption StartPosition	Property Test 2 – CenterScreen
Label1	Caption	Tuliskan nama Anda :
Text1	-	-
Label2	Alignment BackColor Font	2 – Center Palette: <putih> Size: 24
Command1	Caption	OK
Command2	Caption	Selesai

Sehingga UI-nya menjadi seperti ini :



1.3 Menulis Kode Program : Property Test

Buka Jendela Code dan kenali bagian-bagian di dalamnya :



Program yang berbasis Windows bersifat *event-driven*, artinya program bekerja berdasarkan event yang terjadi pada object di dalam program tersebut. Misalnya, jika seorang user meng-klik sebuah tombol maka program akan memberikan “reaksi” terhadap event klik tersebut. Program akan memberikan “reaksi” sesuai dengan kode-kode program yang dibuat untuk suatu event pada object tertentu.

Pilih object Command1 pada bagian Object Selector. Secara otomatis pada bagian Code Editor akan muncul blok kode program berikut :


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

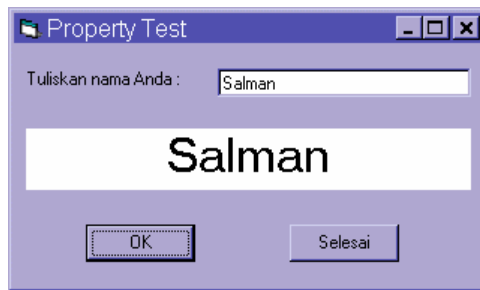
Event Click merupakan event *default* (standar) dari object Command1, Anda bisa melihat event lainnya pada bagian Event Selector.

Pada bagian Code Editor ketikkan kode programnya sebagai berikut :

```
Private Sub Command1_Click()  
    Label2.Caption = Text1.Text  
End Sub  
  
Private Sub Command2_Click()  
    End  
End Sub
```

Simpan Project1 (nama file : **Latihan.vbp**) dan Form1 (nama file : **Lat1.frm**). VB 6 akan menyimpan program yang Anda buat ke dalam beberapa file yang saling terkait, antara lain : file project (*.vbp), file form (*.frm) dan file module (*.bas).

Coba jalankan Project1 dengan meng-klik tombol Start () pada bagian Toolbar atau tekan tombol F5 pada keyboard. Ketikkan “Salman” di dalam TextBox kemudian klik tombol OK. Maka akan tampil hasil sebagai berikut :



Bila ingin keluar klik tombol Selesai.

Penjelasan kode program :

Berikut diberikan penjelasan bagaimana cara kerja program sederhana di atas

```
Private Sub Command1_Click()  
    Label2.Caption = Text1.Text  
End Sub
```

Ketika seorang *user* meng-klik tombol OK (object Command1) maka isi tulisan pada object Label2 akan sama dengan isi tulisan pada object Text1. Property Caption pada obyek Label2 digunakan untuk menampilkan text tertentu di obyek tersebut.

Selanjutnya untuk keluar dari program dituliskan kode berikut :

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

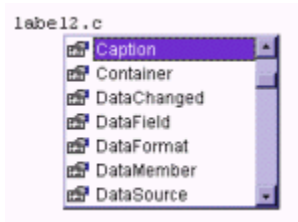

Ketika seorang *user* meng-klik tombol Selesai (object Command2) maka program akan berhenti. Perintah End digunakan untuk menutup program.

Menampilkan tulisan merupakan salah satu property yang dimiliki oleh komponen Label. Property isi tulisan pada komponen Label adalah Caption, sedangkan pada komponen Textbox adalah Text.

Cara penulisan (*syntax*) kode program untuk mengatur property sebuah object adalah sebagai berikut :

```
nama_object.property = value
```

Perhatikan, setelah Anda mengetikkan tanda titik (dot) dibelakang nama_object maka VB 6 akan menampilkan daftar property dan method yang dimiliki oleh nama_object :

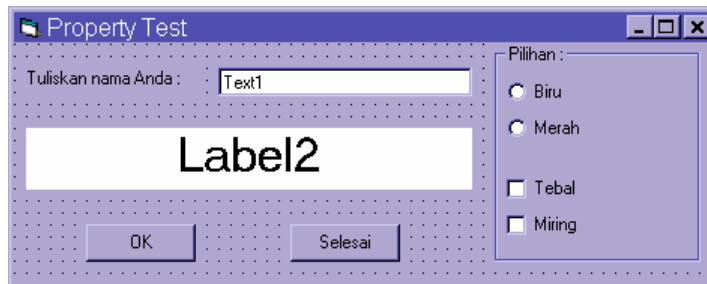


Fasilitas ini disebut Intellisense, dan sangat membantu Anda agar tidak salah mengetikkan nama_object maupun property-nya.

1.4 Modifikasi Kode Program

Ubah tampilan UI-nya menjadi seperti ini :

(Tambahkan komponen Frame, OptionButton dan CheckBox)



Atur property object-object tambahannya sebagai berikut :

Object	Properties	Value
Frame1	Caption	Pilihan :
Option1	Caption	Biru
Option2	Caption	Merah
Check1	Caption	Tebal
Check2	Caption	Miring

Obyek Frame1 harus dibuat terlebih dahulu si form, dan selanjutnya obyek Option dan Check dibuat di dalam Frame1 tersebut. Tujuan pembuatan berbagai pilihan tersebut adalah memperkenalkan pada Anda fasilitas Option dan CheckBox yang digunakan untuk membuat pilihan dalam aplikasi.

Tambahkan kode programnya sebagai berikut :

```
Private Sub Option1_Click()  
    Label2.ForeColor = vbBlue  
End Sub  
  
Private Sub Option2_Click()  
    Label2.ForeColor = vbRed  
End Sub  
  
Private Sub Check1_Click()  
    Label2.FontBold = Check1.Value  
End Sub  
  
Private Sub Check2_Click()  
    Label2.FontItalic = Check2.Value  
End Sub
```

Simpan kembali Form1, kemudian coba jalankan Project1 dan lihat hasilnya :



Cepat Mahir Visual Basic 6.0

Krisna D. Octovhiana
mail4krisna@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Sehuruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

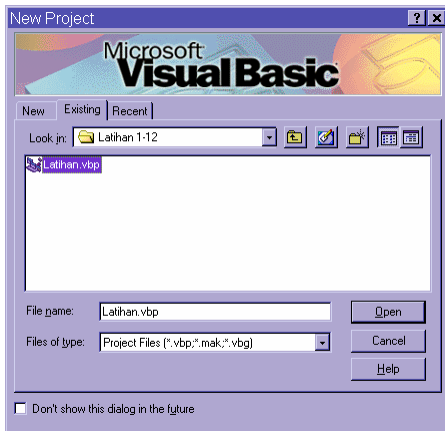
Bab 3

Menggunakan Event dan Method

1.1 Membuat *User Interface* (UI)

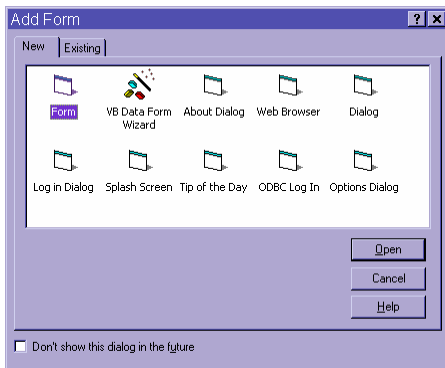
Aktifkan VB 6 melalui tombol Start.

Buka kembali project **Latihan.vbp** :



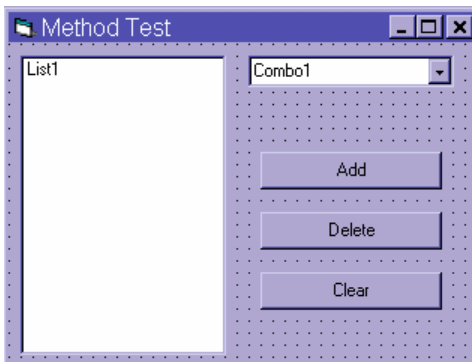
- Klik tab Existing.
- Pilih folder masing-masing pada bagian Look in.
- Pilih file Latihan.vbp.
- Klik tombol Open.

Tambahkan Form baru ke dalam Project :



- Klik menu Project > Add Form.
- Klik tab New dan pilih Form.
- Klik tombol Open.

Pada Jendela Form buatlah UI seperti ini :
(Gunakan komponen ListBox, ComboBox dan CommandButton)



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form2	Caption StartPosition	Method Test 2 – CenterScreen
List1	-	-
Combo1	-	-
Command1	Caption	Add
Command2	Caption	Delete
Command3	Caption	Clear

1.2 Menulis Kode Program : Method Test

Pada bagian Code Editor ketikkan kode programnya sebagai berikut :

```
Private Sub Form_Load()  
    Combo1.AddItem "Umar"  
    Combo1.AddItem "Salman"  
    Combo1.AddItem "Halimah"  
    Combo1.AddItem "Shafira"  
End Sub
```

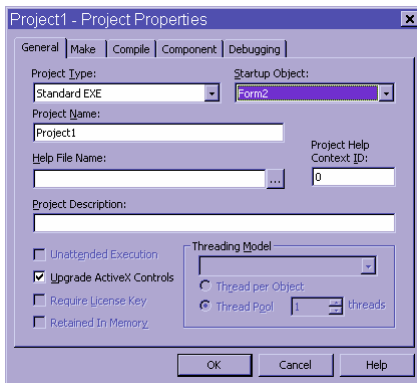
```
Private Sub Command1_Click()  
    List1.AddItem Combo1.Text  
End Sub
```

```
Private Sub Command2_Click()  
    List1.RemoveItem List1.ListIndex  
End Sub
```

```
Private Sub Command3_Click()  
    List1.Clear  
End Sub
```

Simpan Form2 (nama file : **Lat2.frm**).

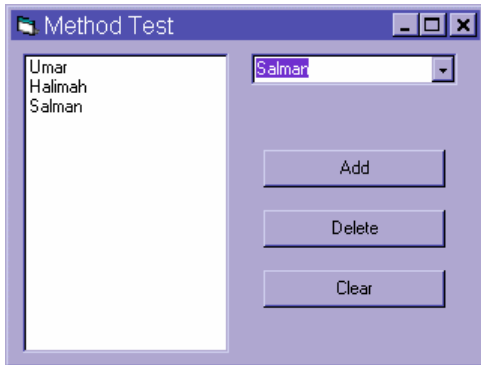
Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form2 :



Keterangan : Startup Object menentukan object mana yang muncul setiap kali sebuah project dijalankan.

Coba jalankan Project1 :

- ✓ Untuk menambah isi ListBox : pilih “Umar” di dalam ComboBox kemudian klik tombol Add. Ulangi untuk nama-nama yang lain secara acak.
- ✓ Untuk menghapus salah satu isi ListBox : pilih “Shafira” di dalam ListBox kemudian klik tombol Delete.
- ✓ Untuk menghapus semua isi ListBox : klik tombol Clear.



Penjelasan kode program :

```
Private Sub Form_Load()  
    Combo1.AddItem "Umar"  
    Combo1.AddItem "Salman"  
    Combo1.AddItem "Halimah"  
    Combo1.AddItem "Shafira"  
End Sub
```

Pada saat Form2 di-load ke dalam memori komputer, object Combo1 diisi dengan beberapa nama.

```
Private Sub Command1_Click()  
    List1.AddItem Combo1.Text  
End Sub
```

Ketika seorang *user* meng-klik tombol Add (object Command1) maka isi object List1 ditambah sesuai dengan isi object Combo1 yang sudah dipilih.

```
Private Sub Command2_Click()  
    List1.RemoveItem List1.ListIndex  
End Sub
```

Ketika seorang *user* meng-klik tombol Delete (object Command2) maka isi object List1 yang dipilih akan dihapus. Catatan : bila belum ada isi yang dipilih akan terjadi *error*.

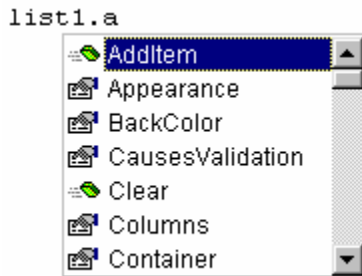
```
Private Sub Command3_Click()  
    List1.Clear  
End Sub
```

Ketika seorang user meng-klik tombol Clear (object Command3) maka semua isi object List1 akan dihapus.

Cara penulisan (*syntax*) kode program untuk menggunakan method dari sebuah object adalah sebagai berikut :

nama_object.method [argument]

Perhatikan, setelah Anda mengetikkan tanda titik (dot) dibelakang nama_object maka VB 6 akan menampilkan daftar property dan method yang dimiliki oleh nama_object :



Fasilitas ini akan membantu Anda agar tidak salah mengetikkan nama_object maupun method-nya.

Komponen ComboBox dan ListBox memiliki beberapa method untuk mengatur isi (*item*) di dalamnya, yaitu :

- **AddItem** – method ini akan menambah isi sesuai dengan argument yang diberikan berupa tulisan. Setiap kali ada isi yang ditambahkan akan diberikan nomer index secara otomatis dimulai dari 0 (nol).

Contoh :

Combo1.AddItem “Umar” → “Umar” adalah argument sebagai isi yang akan ditambahkan ke dalam object Combo1.

List1.AddItem Combo1.Text → property Text dari object Combo1 adalah argument sebagai isi yang akan ditambahkan ke dalam object List1.

- **RemoveItem** – method ini akan menghapus isi sesuai dengan argument yang diberikan berupa nomer index dari isi yang dipilih. Setiap kali ada isi yang dihapus maka nomer index akan disusun ulang secara otomatis. Bila isinya tidak ada (kosong) nomer index-nya adalah -1.

Contoh :

List1.RemoveItem List1.ListIndex → property ListIndex dari object List1 adalah argument sebagai nomer index isi yang akan dihapus.

- **Clear** – method ini akan menghapus semua isi (tidak membutuhkan argument).

Contoh :

List1.Clear → semua isi object List1 akan dihapus.

Cepat Mahir Visual Basic 6.0

Krisna D. Octovhiana
mail4krisna@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Sehuruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Bab 4 Penggunaan Data dan Variabel

1.1 Mengenal Data dan Variabel

Ketika seorang *user* (pengguna) menggunakan sebuah program komputer, seringkali komputer memintanya untuk memberikan informasi. Informasi ini kemudian disimpan atau diolah oleh komputer. Informasi inilah yang disebut dengan **DATA**.

Visual Basic 6 mengenal beberapa type data, antara lain :

- **String** adalah type data untuk teks (huruf, angka dan tanda baca).
- **Integer** adalah type data untuk angka bulat.
- **Single** adalah type data untuk angka pecahan.
- **Currency** adalah type data untuk angka mata uang.
- **Date** adalah type data untuk tanggal dan jam.
- **Boolean** adalah type data yang bernilai TRUE atau FALSE.

Data yang disimpan di dalam memory komputer membutuhkan sebuah wadah. Wadah inilah yang disebut dengan **VARIABEL**. Setiap variabel untuk menyimpan data dengan type tertentu membutuhkan alokasi jumlah memory (*byte*) yang berbeda.

Variabel dibuat melalui penulisan deklarasi variabel di dalam kode program :

Dim <nama_variabel> As <type_data>

Contoh : Dim nama_user As String

Aturan di dalam penamaan variabel :

- Harus diawali dengan huruf.
- Tidak boleh menggunakan spasi. Spasi bisa diganti dengan karakter underscore (_).
- Tidak boleh menggunakan karakter-karakter khusus (seperti : +, -, *, /, <, >, dll).
- Tidak boleh menggunakan kata-kata kunci yang sudah dikenal oleh Visual Basic 6 (seperti : dim, as, string, integer, dll).

Sebuah variabel hanya dapat menyimpan satu nilai data sesuai dengan type datanya. Cara mengisi nilai data ke dalam sebuah variabel :

<nama_variabel> = <nilai_data>

Contoh : nama_user = "krisna"

Untuk type data tertentu nilai_data harus diapit tanda pembatas. Type data string dibatasi tanda petik-ganda : "nilai_data". Type data date dibatasi tanda pagar : #nilai_data#. Type data lainnya tidak perlu tanda pembatas.

Sebuah variabel mempunyai ruang-lingkup (*scope*) dan waktu-hidup (*lifetime*) :

- **Variabel global** adalah variabel yang dapat dikenali oleh seluruh bagian program. Nilai data yang tersimpan didalamnya akan hidup terus selama program berjalan.
- **Variabel lokal** adalah variabel yang hanya dikenali oleh satu bagian program saja. Nilai data yang tersimpan didalamnya hanya hidup selama bagian program tersebut dijalankan.

Variabel yang nilai datanya bersifat tetap dan tidak bisa diubah disebut **KONSTANTA**. Penulisan deklarasi konstanta di dalam kode program :

Const <nama_kontanta> As <type_data> = <nilai_data>

Contoh : Const tgl_gajian As Date = #25/09/2003#

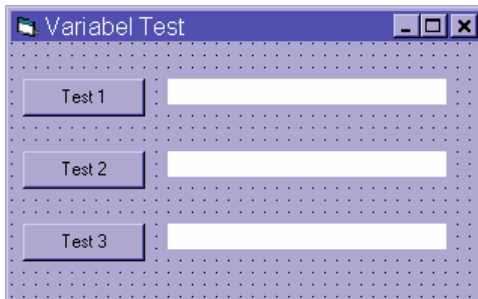
1.2 Contoh Program : Variabel Test

Aktifkan VB 6 melalui tombol Start.

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini :
(Gunakan komponen CommandButton dan Label)



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form3	Caption StartPosition	Variabel Test 2 – CenterScreen
Command1-3	Caption	Test 1 Test 2 Test 3
Label1-3	BackColor Caption	Palette : <putih> <kosong>

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

```
Dim test2 As Integer  
  
Private Sub Command1_Click()  
    Dim test1 As String  
  
    test1 = "nusantara"  
  
    Label1.Caption = test1  
    Label2.Caption = test2  
    Label3.Caption = test3  
End Sub
```

```
Private Sub Command2_Click()
    test2 = 10
```

```
    Label1.Caption = test1
    Label2.Caption = test2
    Label3.Caption = test3
End Sub
```

```
Private Sub Command3_Click()
    Const test3 As Single = 90.55
```

```
    Label1.Caption = test1
    Label2.Caption = test2
    Label3.Caption = test3
End Sub
```

Simpan Form3 (nama file : **Lat3.frm**).

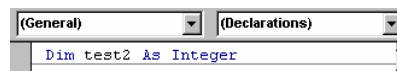
Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form3.

Coba jalankan Project1. Perhatikan sifat variabel global, variabel lokal dan konstanta !

Klik Tombol	Nilai Var. Test1	Nilai var. Test2	Nilai Var. Test3
Test1	nusantara	Kali Pertama : 0 Berikutnya : 10	<kosong>
Test2	<kosong>	10	<kosong>
Test3	<kosong>	10	90,55

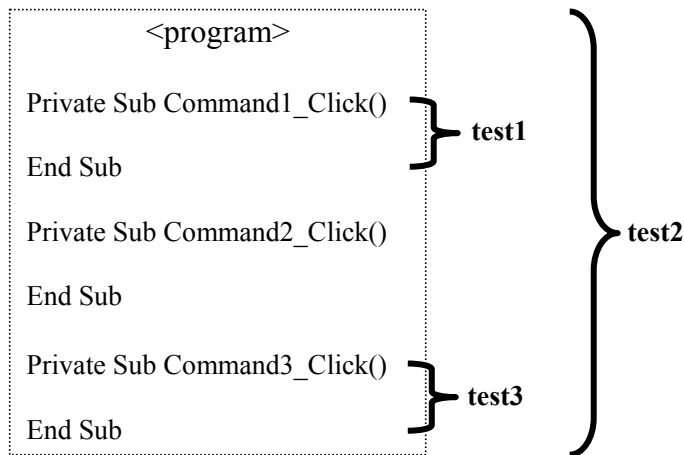
1.3 Penjelasan Kode Program

Kode Program	Deskripsi
Dim test2 As Integer	Variabel test2 dideklarasikan untuk data bertipe integer sebagai variabel global sehingga bisa digunakan oleh semua blok kode program. Catatan : setiap variabel global harus selalu dideklarasikan pada bagian atas kode program - pastikan Object Selector berisi (General) dan Event Selector berisi (Declarations) .
Private Sub Command1_Click() Dim test1 As String	Bila <i>user</i> meng-klik tombol Test1 : Variabel test1 dideklarasikan untuk data bertipe string sebagai variabel lokal khusus untuk blok kode Command1_Click. Catatan : setiap variabel lokal dideklarasikan pada bagian dalam blok kode program tertentu. Blok kode program disebut procedure (akan dibahas dalam bab khusus).



<pre>test1 = "nusantara" Label1.Caption = test1 Label2.Caption = test2 Label3.Caption = test3 End Sub Private Sub Command2_Click() test2 = 10 Label1.Caption = test1 Label2.Caption = test2 Label3.Caption = test3 End Sub Private Sub Command3_Click() Const test3 As Single = 90.55 Label1.Caption = test1 Label2.Caption = test2 Label3.Caption = test3 End Sub</pre>	<p>Variabel test1 diisi dengan data nusantara (string). Ingat tanda “...” sebagai pembatas.</p> <p>Tampilkan isi variabel test1, test2 dan test3 pada Label1, Label2 dan Label3.</p> <ul style="list-style-type: none">Label1 akan menampilkan teks nusantara, karena variabel test1 berisi nusantara.Label2 akan menampilkan angka 0, karena variabel test2 belum ada isinya. Variabel test2 boleh digunakan karena bersifat global.Label3 tidak menampilkan apa-apa (kosong), karena variabel test3 hanya bersifat lokal untuk blok kode Command3_Click. <p>Bila <i>user</i> meng-klik tombol Test2 : Variabel test2 diisi dengan data 10 (integer). Variabel test2 boleh diisi/diganti isinya karena bersifat global.</p> <p>Tampilkan isi variabel test1, test2 dan test3 pada Label1, Label2 dan Label3.</p> <ul style="list-style-type: none">Label1 tidak menampilkan apa-apa (kosong), karena variabel test1 hanya bersifat lokal untuk blok kode Command1_Click.Label2 akan menampilkan angka 10, karena variabel test2 isinya angka 10.Label3 tidak menampilkan apa-apa (kosong), karena variabel test3 hanya bersifat lokal untuk blok kode Command3_Click. <p>Bila <i>user</i> meng-klik tombol Test3 : Variabel test3 dideklarasikan sekaligus diisi dengan data 90,55 (single) – sebagai konstanta.</p> <p>Tampilkan isi variabel test1, test2 dan test3 pada Label1, Label2 dan Label3.</p> <ul style="list-style-type: none">Label1 tidak menampilkan apa-apa (kosong), karena variabel test1 hanya bersifat lokal untuk blok kode Command1_Click.Label2 akan menampilkan angka 10, karena variabel test2 isinya masih angka 10. Variabel test2 boleh digunakan karena bersifat global.Label3 akan menampilkan angka 90,55, karena variabel test3 isinya angka 90,55.
---	---

Scope dan *LifeTime* masing-masing variabel :



Untuk membuktikan bahwa nilai data sebuah konstanta bersifat tetap, modifikasi kode program sbb :

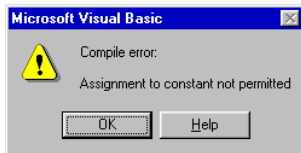
```
Private Sub Command3_Click()
  Const test3 As Single = 90.55

  test3 = 50.22

  Label1.Caption = test1
  Label2.Caption = test2
  Label3.Caption = test3
End Sub
```

Perhatikan : penulisan angka pecahan (single) pada kode program tetap menggunakan tanda titik (.) meskipun regional setting komputernya sudah Indonesia. Bila tidak akan muncul pesan error.

Coba jalankan Project1, kemudian klik tombol Test 3. Akan muncul pesan error :



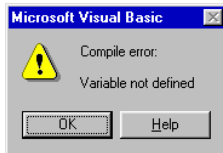
Klik OK, kemudian stop programnya dengan meng-klik tombol End pada Toolbar.

Agar program tetap berjalan tanpa harus menghapus kode program yang salah, maka tambahkan tanda petik-tunggal (') di depan kode program yang salah tersebut (contoh : 'test3 = 50.22). Teks setelah tanda petik-tunggal oleh Visual Basic 6 diartikan sebagai komentar dan tidak akan di-eksekusi ketika program berjalan.

Visual Basic 6 membolehkan sebuah variabel tidak dideklarasikan terlebih dahulu (dengan perintah Dim atau Const), tetapi hal ini akan membuat error pada kode program agak sulit dilacak. Jadi, sebaiknya setiap variabel selalu dideklarasikan (baik yang lokal maupun global). Agar dapat diketahui variabel mana yang belum dideklarasikan, pada bagian atas kode program tambahkan perintah berikut :

Option Explicit

Coba jalankan Project1, kemudian klik tombol Test 1. Akan muncul pesan error :



Klik OK, kemudian stop programnya dengan meng-klik tombol End pada Toolbar.

Cepat Mahir Visual Basic 6.0

Krisna D. Octovhiana
mail4krisna@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Bab 5

Penggunaan Operator

1.1 Apa itu Operator ?

OPERATOR di dalam bahasa pemrograman bisa diartikan sebagai simbol yang digunakan untuk melakukan suatu operasi terhadap nilai data. Simbol operator bisa berupa karakter ataupun kata khusus.

Visual Basic 6 mengenal tiga jenis operator, yaitu :

1. **Operator Aritmatika** digunakan untuk operasi matematis terhadap nilai data. Simbol-simbol yang digunakan :

Simbol	Operasi Matematis	Contoh
^	pemangkatan	5^2 hasilnya 25
*	perkalian	$5 * 2$ hasilnya 10
/	pembagian (hasil pecahan)	$5 / 2$ hasilnya 2,5
\	pembagian (hasil bulat)	$5 \setminus 2$ hasilnya 2
Mod	sisa pembagian	$5 \text{ Mod } 2$ hasilnya 1
+	penjumlahan	$5 + 2$ hasilnya 7
-	pengurangan	$5 - 2$ hasilnya 3
&	penggabungan string	$5 \& 2$ hasilnya 52

2. **Operator Perbandingan** digunakan untuk operasi yang membandingkan nilai data. Simbol-simbol yang digunakan :

Simbol	Operasi Perbandingan	Contoh
<	lebih kecil	$5 < 2$ hasilnya FALSE
>	lebih besar	$5 > 2$ hasilnya TRUE
<=	lebih kecil atau sama dengan	$5 <= 2$ hasilnya FALSE
>=	lebih besar atau sama dengan	$5 >= 2$ hasilnya TRUE
=	sama dengan	$5 = 2$ hasilnya FALSE
<>	tidak sama dengan	$5 <> 2$ hasilnya TRUE

3. **Operator Logika** digunakan untuk operasi yang membandingkan suatu perbandingan. Simbol-simbol yang digunakan :

Simbol	Operasi Logika	Contoh
Or	atau	$(5 < 2) \text{ Or } (5 > 2)$ hasilnya TRUE
And	dan	$(5 < 2) \text{ And } (5 > 2)$ hasilnya FALSE
Not	Tidak	$\text{Not } (5 < 2)$ hasilnya TRUE

1.2 Contoh Program : Operator Test

Aktifkan VB 6 melalui tombol Start.

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini :



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form4	Caption StartPosition	Operator Test 2 – CenterScreen
Label1-2	Caption	Var 1 : Var 2 :
Text1-2	Text	<kosong>
Frame1-3	Caption	Op. Aritmatika Op. Perbandingan Op. Logika
Option1-14	Caption	+, -, *, /, && >, <, =, <>, >=, <= Not, Or, And
Label3	Alignment BackColor ForeColor Caption	2 – Center Palette : <putih> Palette : <merah> <kosong>

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

```
Dim var1 As Single, var2 As Single
Dim hasil As Single
```

```
Private Sub Option1_Click()
    var1 = Text1.Text
    var2 = Text2.Text
    hasil = var1 + var2
    Label3.Caption = hasil
End Sub
```

```
Private Sub Option2_Click()
    var1 = Text1.Text
    var2 = Text2.Text
```

```
    hasil = var1 - var2  
    Label3.Caption = hasil  
End Sub
```

```
Private Sub Option3_Click()  
    var1 = Text1.Text  
    var2 = Text2.Text  
    hasil = var1 * var2  
    Label3.Caption = hasil  
End Sub
```

```
Private Sub Option4_Click()  
    var1 = Text1.Text  
    var2 = Text2.Text  
    hasil = var1 / var2  
    Label3.Caption = hasil  
End Sub
```

```
Private Sub Option5_Click()  
    var1 = Text1.Text  
    var2 = Text2.Text  
    hasil = var1 & var2  
    Label3.Caption = hasil  
End Sub
```

```
Private Sub Option6_Click()  
    var1 = Text1.Text  
    var2 = Text2.Text  
    hasil = (var1 > var2)  
    'Label3.Caption = hasil  
    Label3.Caption = Format(hasil, "True/False")  
End Sub
```

```
Private Sub Option7_Click()  
    var1 = Text1.Text  
    var2 = Text2.Text  
    hasil = (var1 < var2)  
    Label3.Caption = Format(hasil, "True/False")  
End Sub
```

```
Private Sub Option8_Click()  
    var1 = Text1.Text  
    var2 = Text2.Text  
    hasil = (var1 = var2)  
    Label3.Caption = Format(hasil, "True/False")  
End Sub
```

```
Private Sub Option9_Click()  
    var1 = Text1.Text  
    var2 = Text2.Text  
    hasil = (var1 <> var2)  
    Label3.Caption = Format(hasil, "True/False")  
End Sub
```

```
Private Sub Option10_Click()  
    var1 = Text1.Text  
    var2 = Text2.Text  
    hasil = (var1 >= var2)  
    Label3.Caption = Format(hasil, "True/False")  
End Sub
```

```
Private Sub Option11_Click()  
    var1 = Text1.Text  
    var2 = Text2.Text  
    hasil = (var1 <= var2)  
    Label3.Caption = Format(hasil, "True/False")  
End Sub
```

```
Private Sub Option12_Click()  
    var1 = IIf(Text1.Text = "True", -1, 0)  
    hasil = Not (var1)  
    Label3.Caption = Format(hasil, "True/False")  
End Sub
```

```
Private Sub Option13_Click()  
    var1 = IIf(Text1.Text = "True", -1, 0)  
    var2 = IIf(Text2.Text = "True", -1, 0)  
    hasil = (var1 Or var2)  
    Label3.Caption = Format(hasil, "True/False")  
End Sub
```

```
Private Sub Option14_Click()  
    var1 = IIf(Text1.Text = "True", -1, 0)  
    var2 = IIf(Text2.Text = "True", -1, 0)  
    hasil = (var1 And var2)  
    Label3.Caption = Format(hasil, "True/False")  
End Sub
```

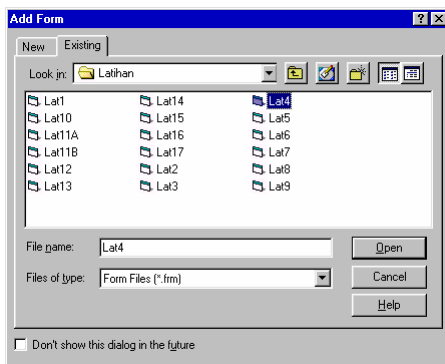
Simpan Form4 (nama file : **Lat4.frm**).

Catatan :

Anda bisa men-download file Lat4.frm serta file-file latihan sebelumnya.

Untuk menambahkan file Form ke dalam sebuah Project, lakukan langkah-langkah berikut ini :

1. Klik menu Project > Add Form. Kemudian klik tab Existing.



2. Tentukan lokasi file pada bagian Look in, lalu pilih filenya.
3. Klik tombol Open.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form4.

Coba jalankan Project1. Perhatikan hasil setiap operasi !

1.3 Penjelasan Kode Program

Kode Program	Deskripsi
<pre>Dim var1 As Single, var2 As Single Dim hasil As Single Private Sub Option1_Click() var1 = Text1.Text var2 = Text2.Text hasil = var1 + var2 Label3.Caption = hasil End Sub</pre>	<p>Deklarasi variabel global : var1, var2 dan hasil untuk menyimpan data bertipe single.</p> <p>Ketika Option1 di-klik oleh <i>user</i> : Simpan angka yang diketik pada Text1 → var1. Simpan angka yang diketik pada Text2 → var2. Jumlahkan var1 dengan var2 → hasil. Tampilkan hasil pada Label3.</p> <p>Catatan :</p> <ul style="list-style-type: none"> ▪ Prosesnya sama untuk Option2 s/d Option 5. ▪ Operator & sering digunakan untuk menggabungkan data string : a & b → ab. ▪ Akan terjadi <i>error</i> bila Text1 atau Text2 kosong atau bukan diketik angka. Penanganan <i>error</i> akan dibahas pada bab lain.
<p>...</p>	
<pre>Private Sub Option6_Click() var1 = Text1.Text var2 = Text2.Text hasil = (var1 > var2) Label3.Caption = Format(hasil, "True/False") End Sub</pre>	<p>Ketika Option6 di-klik oleh <i>user</i> : Simpan angka yang diketik pada Text1 → var1. Simpan angka yang diketik pada Text2 → var2. Bandingkan apakah var1 lebih besar dari var2. Tampilkan hasil perbandingan pada Label3.</p> <p>Catatan :</p> <ul style="list-style-type: none"> ▪ Prosesnya sama untuk Option7 s/d Option11. ▪ Hasil dari suatu perbandingan adalah True (benar) atau False (salah). Komputer menampilkan True dengan angka -1 dan False dengan angka 0. ▪ Untuk mengubah angka -1 menjadi kata "True" dan angka 0 menjadi kata "False" digunakan perintah Format.
<p>...</p>	
<pre>Private Sub Option12_Click() var1 = IIf(Text1.Text = "True", -1, 0) hasil = Not (var1) Label3.Caption = Format(hasil, "True/False") End Sub</pre>	<p>Ketika Option12 di-klik oleh <i>user</i> : Bila Text1 diketik kata True ubah menjadi angka -1 selain itu ubah menjadi angka 0 → var1 Bandingkan var1 menggunakan operator Not. Tampilkan hasilnya pada Label3.</p> <p>Catatan :</p> <ul style="list-style-type: none"> ▪ Prosesnya sama untuk Option13 dan Option14. ▪ Untuk mengubah kata "True" menjadi angka -1 dan selain itu menjadi angka 0 digunakan perintah IIF. Perintah IIF akan dibahas pada bab lain. ▪ Komputer hanya bisa memproses perbandingan dengan menggunakan angka -1 dan 0.

	<ul style="list-style-type: none">▪ Data string bersifat <i>case sensitive</i>. Jadi, kata “True” harus ditulis benar (huruf T kapital), karena “True” berbeda dengan “true” atau “tRUE” dll.
--	---

Coba lengkapi hasil operasi logika berikut ini :

Var1	Var2	Or	And
True	True		
True	False		
False	True		
False	False		

Bisa Anda bedakan sifat operator Or dengan And ?

Cepat Mahir Visual Basic 6.0

Krisna D. Octovhiana
mail4krisna@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Bab 6

Penggunaan Struktur Kontrol IF

1.1 Mengenal Struktur Kontrol

Struktur kontrol di dalam bahasa pemrograman adalah perintah dengan bentuk (struktur) tertentu yang digunakan untuk mengatur (mengontrol) jalannya program.

Visual Basic 6 mengenal dua jenis struktur kontrol, yaitu :

1. **Struktur kontrol keputusan** - digunakan untuk memutuskan kode program mana yang akan dikerjakan berdasarkan suatu kondisi. Akan dibahas pada bab ini.
2. **Struktur kontrol pengulangan** - digunakan untuk melakukan pengulangan kode program. Akan dibahas pada bab selanjutnya.

Ada dua bentuk struktur kontrol keputusan, yaitu :

1. Struktur **IF...THEN**. Akan dibahas pada bab ini.
2. Struktur **SELECT...CASE**. Akan dibahas pada bab selanjutnya.

Bentuk penulisan (*syntax*) struktur IF...THEN :

1. **IF <kondisi> THEN <kode program>**
Bila <kondisi> bernilai True maka <kode program> akan dikerjakan.
2. **IF <kondisi> THEN**
 <blok kode program 1>
ELSE
 <blok kode program 2>
END IF

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

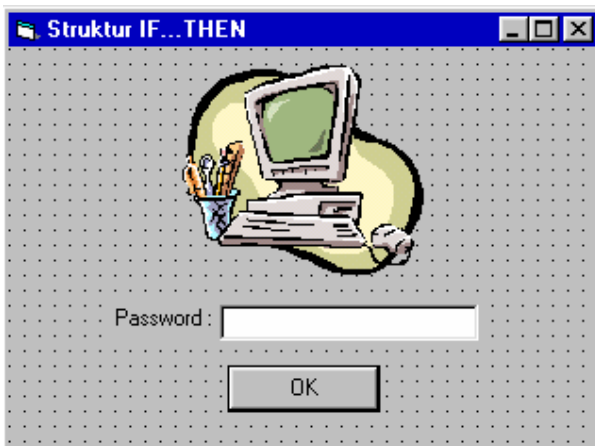
1.1 Contoh Program : Struktur IF...THEN

Aktifkan VB 6 melalui tombol Start.

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini :
(Gunakan komponen Image, Label, TextBox dan CommandButton)



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form5	Caption StartPosition	Struktur IF...THEN 2 – CenterScreen
Image1	Stretch Picture Visible	True Komputer.wmf False
Label1	Caption	Password :
Text1	PasswordChar Text	* <kosong>
Command1	Caption Default	OK True

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

```
Private Sub Command1_Click()
    If Text1.Text = "nusantara" Then Image1.Visible = True
End Sub
```

Simpan Form5 (nama file : **Lat5.frm**).

Catatan :

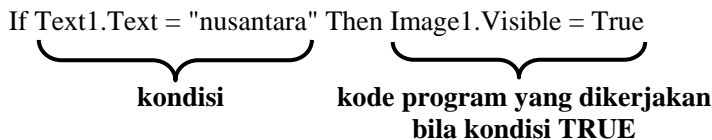
Anda bisa men-download file Lat5.frm dan Komputer.wmf.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form5.

Coba jalankan Project1 :

- Ketikkan sembarang teks pada TextBox lalu klik tombol OK atau tekan Enter → tidak terjadi apa-apa.
- Ketikkan “nusantara” pada TextBox lalu klik tombol OK atau tekan Enter → gambar komputer akan muncul.

Penjelasan kode program :



Modifikasi kode programnya menjadi sebagai berikut :

```
Private Sub Command1_Click()
    If Text1.Text = "nusantara" Then
        Image1.Visible = True
        Text1.Enabled = False
        Command1.Enabled = False
    Else
        MsgBox "Passwordnya Salah !"
    End If
End Sub
```



```
Text1.Text = ""  
Text1.SetFocus  
End If  
End Sub
```

Coba jalankan Project1 :

- Ketikkan sembarang teks pada TextBox lalu klik tombol OK atau tekan Enter → muncul kotak pesan “Passwordnya Salah!”. Klik tombol OK pada kotak pesan tersebut.
- Ketikkan “nusantara” pada TextBox lalu klik tombol OK atau tekan Enter → gambar komputer akan muncul. TextBox dan tombol OK menjadi *disable* (tidak bisa digunakan).

Penjelasan kode program :

```
      kondisi  
      ┌──────────────────┐  
If Text1.Text = "nusantara" Then  
    Image1.Visible = True  
    Text1.Enabled = False  
    Command1.Enabled = False  
Else  
    MsgBox "Passwordnya Salah !"  
    Text1.Text = ""  
    Text1.SetFocus  
End If
```

└──────────────────┘ **blok kode program yang dikerjakan
bila kondisi TRUE**

└──────────────────┘ **blok kode program yang dikerjakan
bila kondisi FALSE**

Catatan tambahan :

- Teks “nusantara” harus diketik huruf kecil semua. Ingat : data string bersifat *case sensitive* !
- Agar password-nya tidak bersifat *case sensitive*, modifikasi pernyataannya menjadi :
If LCCase(Text1.Text) = "nusantara" Then
Fungsi LCCase adalah untuk mengkonversi semua string yang diinput ke Text1.Text menjadi huruf kecil, walaupun user menginputnya dengan huruf kapital.

TANTANGAN :

Coba Anda modifikasi struktur kontrol dan kode programnya sehingga seorang *user* hanya mempunyai 3 kali kesempatan untuk mengetikkan password-nya. Setelah 3 kali kesempatan dan password-nya masih salah maka program akan otomatis berhenti.

Cepat Mahir Visual Basic 6.0

Krisna D. Octovhiana
mail4krisna@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Bab 7

Penggunaan Struktur Kontrol SELECT

1.1 Mengenal Struktur Kontrol

Struktur kontrol di dalam bahasa pemrograman adalah perintah dengan bentuk (struktur) tertentu yang digunakan untuk mengatur (mengontrol) jalannya program.

Visual Basic 6 mengenal dua jenis struktur kontrol, yaitu :

1. **Struktur kontrol keputusan** - digunakan untuk memutuskan kode program mana yang akan dikerjakan berdasarkan suatu kondisi. Akan dibahas pada bab ini.
2. **Struktur kontrol pengulangan** - digunakan untuk melakukan pengulangan kode program. Akan dibahas pada bab selanjutnya.

Ada dua bentuk struktur kontrol keputusan, yaitu :

1. Struktur **IF...THEN**. Sudah dibahas pada bab yang lalu.
2. Struktur **SELECT...CASE**. Akan dibahas pada bab ini.

1.2 Struktur Kontrol Select...Case

Bentuk penulisan (*syntax*) struktur SELECT...CASE :

```
SELECT CASE <pilihan>  
CASE <pilihan 1>  
    <blok kode program 1>  
CASE <pilihan 2>  
    <blok kode program 2>  
CASE <pilihan n>  
    <blok kode program n>  
[CASE ELSE  
    <blok kode program x>]  
END SELECT
```

Bila <pilihan> sesuai dengan <pilihan 1> maka <blok kode program 1> akan dikerjakan, dst. Tetapi bila <pilihan> tidak ada yang sesuai dengan <pilihan 1> s/d <pilihan n> maka <blok kode program x> yang akan dikerjakan.

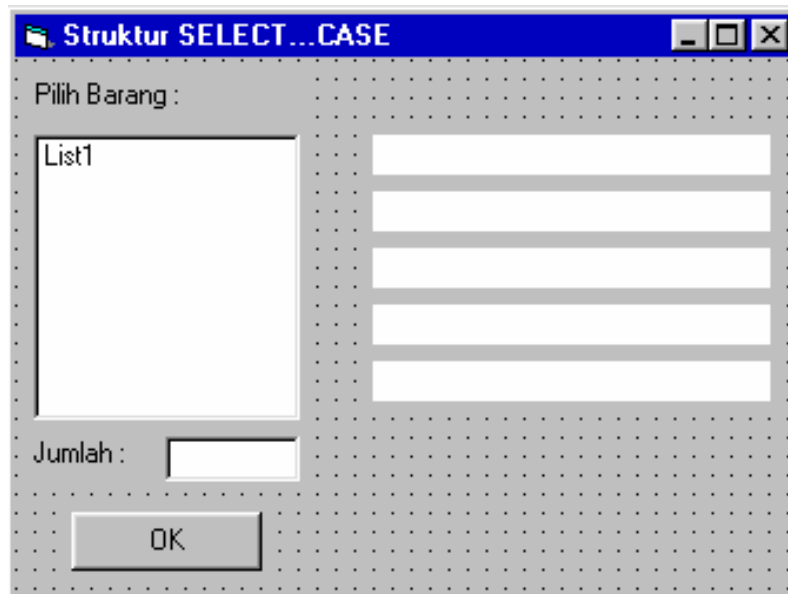
1.3 Contoh Program : Struktur SELECT...CASE

Aktifkan VB 6 melalui tombol Start.

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini :
(Gunakan komponen Label, ListBox, TextBox dan CommandButton)



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form6	Caption StartPosition	Struktur SELECT...CASE 2 – CenterScreen
Label1	Caption	Pilih Barang :
List1	-	-
Label2	Caption	Jumlah :
Text1	Text	<kosong>
Command1	Caption	OK
Label3-7	Name BackColor Caption	lblBarang, lblHarga, lblJumlah, lblDiskon, lblTotal Palette : <putih> <kosong>

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

```
Private Sub Form_Load()
    List1.AddItem "Disket"
    List1.AddItem "Buku"
    List1.AddItem "Kertas"
    List1.AddItem "Pulpen"
End Sub

Private Sub Command1_Click()
    Dim harga As Currency, total As Currency
    Dim jumlah As Integer
    Dim diskon As Single
    Dim satuan As String

    If List1.Text = "" Then
        MsgBox "Anda belum memilih barang !!"
        List1.ListIndex = 0
    End If
End Sub
```

```
Exit Sub
End If

If Text1.Text = "" Then
    MsgBox "Anda belum mengisi jumlah barang !!"
    Text1.SetFocus
    Exit Sub
End If

Select Case List1.Text
Case "Disket"
    harga = 35000
    satuan = "Box"
Case "Buku"
    harga = 20000
    satuan = "Lusin"
Case "Kertas"
    harga = 25000
    satuan = "Rim"
Case "Pulpen"
    harga = 10000
    satuan = "Pak"
End Select
lblBarang.Caption = "Barang : " & List1.Text
lblHarga.Caption = "Harga : " & Format(harga, "Currency") & "/" & satuan
lblJumlah.Caption = "Jumlah : " & Text1.Text & " " & satuan

jumlah = Text1.Text

Select Case jumlah
Case Is < 10
    diskon = 0
Case 10 To 20
    diskon = 0.15
Case Else
    diskon = 0.2
End Select

total = jumlah * (harga * (1 - diskon))

lblDiskon.Caption = "Diskon : " & Format(diskon, "0 %")
lblTotal.Caption = "Total Bayar : " & Format(total, "Currency")
End Sub
```

Simpan Form6 (nama file : **Lat6.frm**).

Catatan :

Anda bisa men-download file Lat6.frm dan file latihan-latihan sebelumnya.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form6.

Coba jalankan Project1 :

- List1 akan terisi nama-nama barang.
- Klik tombol OK → akan muncul kotak pesan "Anda belum memilih barang !!".

- Klik OK, nama barang pertama akan terpilih secara otomatis. Anda boleh memilih nama barang yang lainnya.
- Klik tombol OK → akan muncul kotak pesan "Anda belum mengisi jumlah barang !!".
- Klik OK, text1 akan menjadi focus. Isi jumlah barangnya, misalnya : 10.
- Klik tombol OK → akan tampil : nama barang, harga barang (per-satuannya), jumlah barang (dengan satuannya), diskon dan total bayar.
- Coba ganti pilihan nama barang (pada List1) dan jumlah barang (pada Text1) lalu klik lagi tombol OK.

Penjelasan kode program :

```
Select Case List1.Text      cek barang yang dipilih :
Case "Disket"
    harga = 35000
    satuan = "Box"        } bila barang = Disket
Case "Buku"
    harga = 20000
    satuan = "Lusin"     } bila barang = Buku
Case "Kertas"
    harga = 25000
    satuan = "Rim"      } bila barang = Kertas
Case "Pulpen"
    harga = 10000
    satuan = "Pak"      } bila barang = Pulpen
End Select
```

```
lblBarang.Caption = "Barang : " & List1.Text
lblHarga.Caption = "Harga : " & Format(harga, "Currency") & "/" & satuan
lblJumlah.Caption = "Jumlah : " & Text1.Text & " " & satuan } tampilkan hasilnya
```

```
jumlah = Text1.Text
```

```
Select Case jumlah      cek jumlah barang :
Case Is < 10
    diskon = 0          bila jumlah < 10 → diskon = 0%
Case 10 To 20
    diskon = 0.15      bila jumlah 10-20 → diskon = 15%
Case Else
    diskon = 0.2       bila jumlah > 20 → diskon = 20%
End Select
```

```
total = jumlah * (harga * (1 - diskon))
```

```
lblDiskon.Caption = "Diskon : " & Format(diskon, "0 %")
lblTotal.Caption = "Total Bayar : " & Format(total, "Currency") } Hitung total bayar dan tampilkan hasilnya
```

Catatan :

- Bila jumlah barang diisi dengan selain angka akan muncul pesan error.
- Untuk mengecek isi Text1 angka atau bukan, tambahkan kode program berikut :
If Not IsNumeric(Text1.Text) Then
 MsgBox "Isi jumlah barang harus angka !!"
 Text1.SetFocus
 Exit Sub
End If

Cepat Mahir Visual Basic 6.0

Krisna D. Octovhiana
mail4krisna@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Bab 8

Penggunaan Struktur Kontrol Pengulangan

1.1 Mengenal Struktur Kontrol

Struktur kontrol di dalam bahasa pemrograman adalah perintah dengan bentuk (struktur) tertentu yang digunakan untuk mengatur (mengontrol) jalannya program.

Visual Basic 6 mengenal dua jenis struktur kontrol, yaitu :

1. **Struktur kontrol keputusan** - digunakan untuk memutuskan kode program mana yang akan dikerjakan berdasarkan suatu kondisi. Sudah dibahas pada bab lalu.
2. **Struktur kontrol pengulangan** - digunakan untuk melakukan pengulangan kode program. Akan dibahas pada bab ini.

Ada dua bentuk struktur kontrol pengulangan (*looping*), yaitu :

1. Struktur **FOR...NEXT**.
2. Struktur **DO...LOOP**.

1.2 Struktur Kontrol For...Next

Bentuk penulisan (*syntax*) struktur For...Next :

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

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

1.3 Struktur Kontrol Do...Loop

Bentuk penulisan (*syntax*) struktur Do...Loop :

1. **DO WHILE** <kondisi>
 <blok kode program>
LOOP

<blok kode program> akan diulang selama <kondisi> bernilai TRUE. Pengulangan berhenti bila <kondisi> sudah bernilai FALSE.

2. **DO UNTIL** <kondisi>
 <blok kode program>
LOOP

<blok kode program> akan diulang sampai <kondisi> bernilai TRUE. Pengulangan berhenti bila <kondisi> sudah bernilai TRUE.

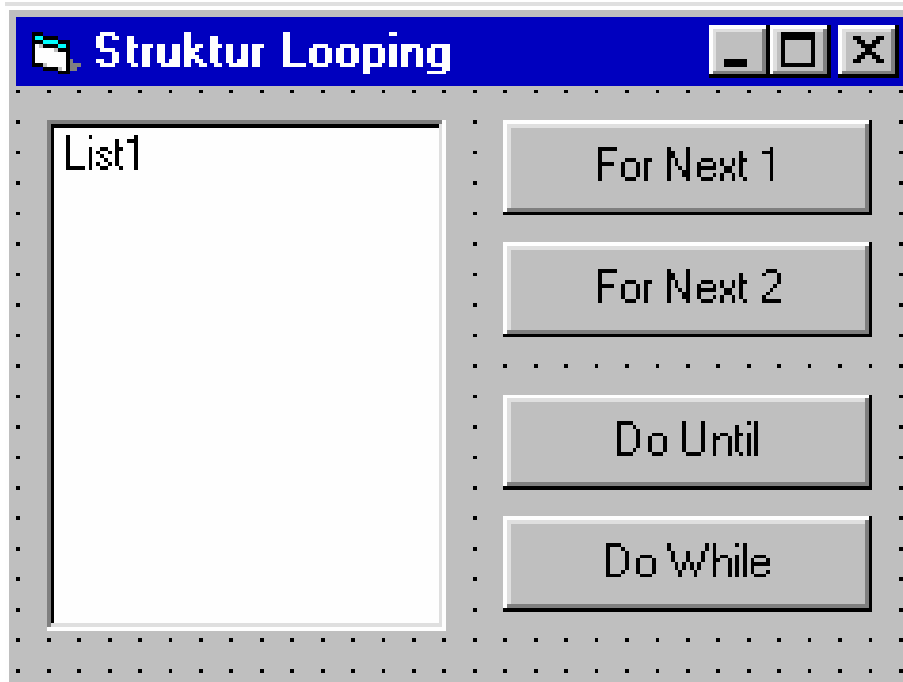
1.4 Contoh Program : Struktur Looping

Aktifkan VB 6 melalui tombol Start.

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini :
(Gunakan komponen ListBox dan CommandButton)



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form7	Caption StartPosition	Struktur Looping 2 - CenterScreen
List1	-	-
Command1-4	Caption	For Next 1 For Next 2 Do Until Do While

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

```
Dim i As Integer
```

```
Private Sub Command1_Click()  
    List1.Clear  
    For i = 1 To 100  
        List1.AddItem "Angka " & i  
    Next i  
End Sub
```

```
Private Sub Command2_Click()  
    List1.Clear
```

```
For i = 100 To 1 Step -2  
    List1.AddItem "Angka " & i  
Next i  
End Sub
```

```
Private Sub Command3_Click()  
    List1.Clear  
    i = Asc("A")  
    Do Until i > Asc("Z")  
        List1.AddItem "Huruf " & Chr(i)  
        i = i + 1  
    Loop  
End Sub
```

```
Private Sub Command4_Click()  
    List1.Clear  
    i = Asc("Z")  
    Do While i >= Asc("A")  
        List1.AddItem "Huruf " & Chr(i)  
        i = i - 1  
    Loop  
End Sub
```

Simpan Form7 (nama file : **Lat7.frm**).

Catatan :

Anda bisa men-download file Lat7.frm dan file latihan-latihan sebelumnya.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form7.

Coba jalankan Project1 :

- Klik Command1 → List1 akan berisi : Angka 1 s/d Angka 100.
- Klik Command2 → List1 akan berisi : Angka genap antara 1 s/d 100 dengan urutan *descending*.
- Klik Command3 → List1 akan berisi : Huruf A s/d Huruf Z.
- Klik Command4 → List1 akan berisi : Huruf A s/d Huruf Z dengan urutan *descending*.

Penjelasan kode program :

```
For i = 1 To 100  
    List1.AddItem "Angka " & i  
Next i
```

pencacah (i) → 1 s/d 100
kode program yang diulang

```
For i = 100 To 1 Step -2  
    List1.AddItem "Angka " & i  
Next i
```

pencacah (i) → 1 s/d 100 dengan langkah -2 setiap pengulangan
kode program yang diulang

```
i = Asc("A")  
Do Until i > Asc("Z")
```

nilai i awal → 65
pengulangan sampai nilai i > 91
kode program yang diulang

```
List1.AddItem "Huruf " & Chr(i)
i = i + 1
Loop
```

nilai i ditambah 1

```
i = Asc("Z")
Do While i >= Asc("A")
    List1.AddItem "Huruf " & Chr(i)
    i = i - 1
Loop
```

nilai i awal → 91
pengulangan selama nilai i >= 65
kode program yang diulang
nilai i dikurangi 1

Catatan :

- Bila ingin keluar dari struktur pengulangan sebelum pengulangannya selesai gunakan perintah **EXIT FOR** atau **EXIT DO**. Contoh :

```
For i = 1 To 100
    List1.AddItem "Angka " & i
    If i = 50 Then Exit For
Next i
```

```
Do Until i > Asc("Z")
    List1.AddItem "Huruf " & Chr(i)
    If Chr(i) = "M" Then Exit Do
    i = i + 1
Loop
```

Cepat Mahir Visual Basic 6.0

Krisna D. Octovhiana
mail4krisna@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Bab 9 Penggunaan Array

1.1 Apa itu Array ?

Array merupakan sekumpulan nilai data yang “dikelompokkan” dalam sebuah variabel. Array digunakan bila ada beberapa nilai data yang tipe datanya sama dan akan mendapat perlakuan yang sama pula. Misalnya, ada 10 nilai data dengan tipe string dan akan diolah dengan cara yang sama, maka akan lebih mudah jika menggunakan sebuah array dibandingkan bila menggunakan 10 variabel yang berbeda. Setiap nilai data di dalam sebuah array disebut **elemen array** dan masing-masing dibedakan dengan nomer indeksinya.

1.2 Menggunakan Array

Sebuah array dideklarasikan dengan cara yang sama dengan variabel, yaitu menggunakan perintah Dim, kemudian diikuti dengan tanda kurung dan jumlah elemen arraynya dikurangi 1 :

Dim nama_array(jumlah_elemen - 1) As tipe_data

Contoh :

Dim NamaSiswa(99) As String → array NamaSiswa akan mempunyai elemen sebanyak 100 dengan nomer indeks mulai dari 0 s/d 99.

Selanjutnya untuk mengisi nilai data ke dalam array :

nama_array(no_indeks) = nilai_data

Contoh :

```
NamaSiswa(0) = "umar"  
NamaSiswa(1) = "fatimah"  
...
```

Untuk “mengosongkan” nilai data, bisa menggunakan struktur kontrol For...Next :

```
For i = 0 To 99  
    NamaSiswa(i) = ""  
Next i
```

Lebih mudah bukan ?

Catatan :

- Agar nomer indeks array dimulai dari 1 (bukan 0), bisa menggunakan perintah **Option Base 1** sebelum perintah Dim-nya, contoh :

```
Option Base 1  
Dim NamaSiswa(100) As String
```

- Atau bisa juga menggunakan kata **To** di dalam penentuan jumlah elemen array, contoh :

```
Dim NamaSiswa(1 To 100) As String
```

- Untuk mengetahui berapa jumlah elemen di dalam sebuah array, bisa menggunakan perintah **Ubound(nama_array)**, contoh :

```
Dim NamaSiswa(1 To 100) As String  
Dim JmlElemen As Integer  
  
JmlElemen = Ubound(NamaSiswa) → hasilnya = 100
```

- Bila jumlah elemen array yang dibutuhkan tidak diketahui atau ingin bisa diubah-ubah, maka bagian jumlah_elemen pada perintah Dim tidak perlu diisi, contoh :

```
Dim NamaSiswa() As String
```

Teknik ini biasa disebut sebagai array dinamis, biasanya untuk mengisi data yang berkembang terus, dan belum bisa ditentukan di awal jumlah elemennya.

- Untuk mengubah jumlah elemen array digunakan perintah Redim :

```
Redim [Preserve] nama_array(jumlah_elemen)
```

Contoh :

```
Redim NamaSiswa(1 To 150) → jumlah elemen array NamaSiswa menjadi 150 dengan  
nomer indeks dari 1 s/d 150.
```

Penggunaan kata Preserve dapat dilihat pada contoh program.

1.3 Contoh Program : Array Test

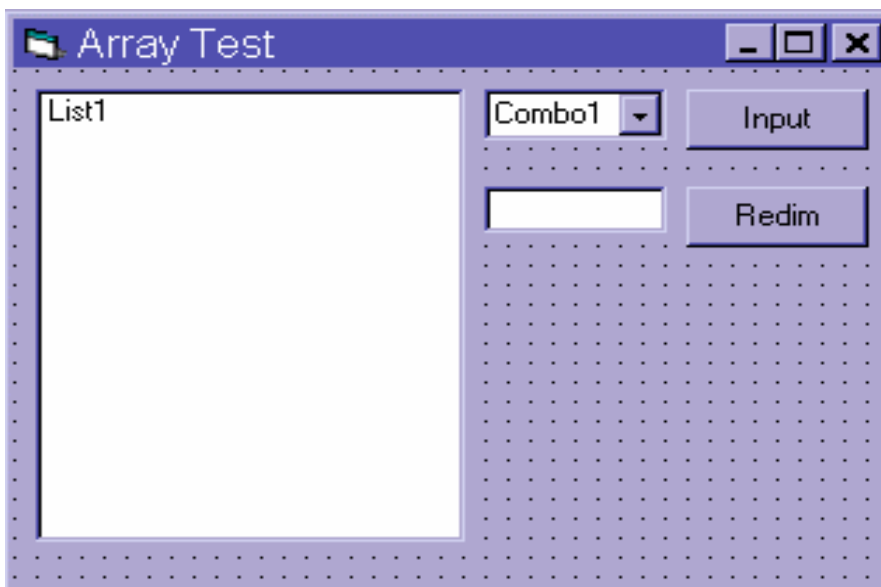
Aktifkan VB 6 melalui tombol Start.

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini :

(Gunakan komponen ListBox, ComboBox, TextBox dan CommandButton)



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form8	Caption StartPosition	Array Test 2 – CenterScreen
List1	-	-
Combo1	-	-
Text1	Text	<kosong>
Command1	Caption	Input
Command2	Caption	Redim

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

```
Dim NamaSiswa(4) As String

Private Sub Form_Load()
    Dim i As Integer

    For i = 0 To 4
        Combo1.AddItem i
    Next i
    Combo1.ListIndex = 0
End Sub

Private Sub Command1_Click()
    Dim no As Integer, i As Integer

    no = CInt(Combo1.Text)
    NamaSiswa(no) = InputBox("Tuliskan nama siswa no : " & no, "Input
Nama Siswa")
    If NamaSiswa(no) <> "" Then
        List1.Clear
        For i = 0 To 4
            List1.AddItem "NamaSiswa(" & i & ")=" & NamaSiswa(i)
        Next i
    End If
End Sub
```

Simpan Form8 (nama file : **Lat8.frm**).

Catatan : Anda bisa men-download file Lat8.frm dan file latihan-latihan sebelumnya.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form8.

Coba jalankan Project1 :

- Pilih nomer indeks elemen array NamaSiswa melalui Combo1 (antara 0 s/d 4)
- Klik Command1 → akan muncul kotak input untuk mengisi nama siswa
- Ketikkan nilai data untuk elemen array dengan nomer indeks yang dipilih, misalnya Salman

- Klik tombol OK
- Isi setiap elemen array NamaSiswa akan muncul di dalam List1
- Ulangi langkahnya untuk mengisi elemen array nomer indeks lainnya

Penjelasan kode program :

```
Dim NamaSiswa(4) As String
Private Sub Form_Load()
    Dim i As Integer

    For i = 0 To 4
        Combo1.AddItem i
    Next i
    Combo1.ListIndex = 0
End Sub
```

Deklarasi array NamaSiswa dengan jumlah elemen 5
Pada saat form di-load ke dalam memori :

Isi Combo1 dengan angka 0 s/d 4
(sesuai dengan no indeks untuk setiap elemen di dalam
array NamaSiswa)

```
Private Sub Command1_Click()  
    Dim no As Integer, i As Integer
```

```
    no = CInt(Combo1.Text)
```

```
    NamaSiswa(no) = InputBox("Tuliskan  
    nama siswa no : " & no, "Input Nama  
    Siswa")
```

```
    If NamaSiswa(no) <> "" Then
```

```
        List1.Clear
```

```
        For i = 0 To 4
```

```
            List1.AddItem "NamaSiswa(" & i  
                & ")=" & NamaSiswa(i)
```

```
        Next i
```

```
    End If
```

```
End Sub
```

Ketika user meng-klik Command1 :

Simpan nomer indeks yang dipilih ke dalam variabel no

Catatan :

Perintah CInt digunakan untuk mengkonversi sebuah nilai data menjadi bertipe integer.

Tampilkan kotak input untuk mengisi nilai data elemen array dengan nomer indeks yang dipilih.

Jika kotak input tidak kosong maka :

Hapus isi List1.

Tampilkan nilai data setiap elemen array NamaSiswa.

Catatan :

Jika tombol Cancel pada kotak input di-klik sama artinya variabel ataupun array yang menyimpan hasil input bernilai kosong ("").

Modifikasi kode programnya menjadi sebagai berikut :

```
Dim NamaSiswa() As String
```

```
Private Sub Form_Load()
```

```
    Dim i As Integer
```

```
    ReDim NamaSiswa(1 To 5)
```

```
    For i = 1 To 5
```

```
        Combo1.AddItem i
```

```
    Next i
```

```
    Combo1.ListIndex = 0
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
    Dim no As Integer, i As Integer
```

```
    no = CInt(Combo1.Text)
```

```
    NamaSiswa(no) = InputBox("Tuliskan nama siswa no : " & no, "Input Nama Siswa")
```

```
    If NamaSiswa(no) <> "" Then
```

```
        List1.Clear
```

```
        For i = 1 To UBound(NamaSiswa)
```

```
            List1.AddItem "NamaSiswa(" & i & ")=" & NamaSiswa(i)
```

```
        Next i
```

```
    End If
```

```
End Sub
```

```
Private Sub Command2_Click()  
    Dim num As Integer, i As Integer  
  
    If Not IsNumeric(Text1.Text) Then Exit Sub  
  
    num = CInt(Text1.Text)  
  
    ReDim NamaSiswa(1 To num)  
  
    Combo1.Clear  
    List1.Clear  
    For i = 1 To UBound(NamaSiswa)  
        Combo1.AddItem i  
        List1.AddItem "NamaSiswa(" & i & ")=" & NamaSiswa(i)  
    Next i  
    Combo1.ListIndex = 0  
End Sub
```

Coba jalankan Project1 :

- Pilih nomer indeks elemen array NamaSiswa melalui Combo1 (antara 1 s/d 5)
- Klik Command1 → akan muncul kotak input untuk mengisi nama siswa
- Ketikkan nilai data untuk elemen array dengan nomer indeks yang dipilih, misalnya Salman
- Klik tombol OK
- Isi setiap elemen array NamaSiswa akan muncul di dalam List1
- Ketikkan jumlah elemen array yang diinginkan pada Text2, lalu klik Command2
- Jumlah elemen array NamaSiswa akan berubah

Catatan :

- Setelah perintah Redim maka jumlah array akan diubah, tetapi mengakibatkan isi setiap nilai data di dalam array tersebut dikosongkan.
- Agar perintah Redim tidak mengosongkan isi setiap nilai data di dalam array-nya, gunakan kata Preserve setelah perintah Redim-nya, contoh :

```
ReDim Preserve NamaSiswa(1 To num)
```

Cepat Mahir Visual Basic 6.0

Krisna D. Octovhiana
mail4krisna@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Bab 10

Penggunaan Kontrol Array

1.1 Apa itu Kontrol Array ?

Kontrol array merupakan sekumpulan kontrol yang “dikelompokkan” dengan nama yang sama di dalam sebuah Form. Kontrol array digunakan bila ada beberapa kontrol yang sama dan akan mendapat perlakuan yang sama pula. Misalnya, ada 5 buah TextBox di dalam sebuah Form dan akan diprogram dengan cara yang sama, maka akan lebih mudah jika membuat sebuah TextBox sebagai kontrol array dibandingkan bila membuat 5 buah TextBox yang berbeda. Setiap object di dalam kontrol array masing-masing dibedakan dengan nomer indeksinya.

1.2 Membuat Kontrol Array

Untuk membuat kontrol array (misalnya TextBox) sebanyak 5 buah di dalam sebuah form, lakukan langkah berikut ini :

- (1) Buatlah sebuah TextBox di dalam sebuah Form.
- (2) Aturilah property TextBox tersebut sebagai berikut :
Name : txtData
Index : 0 (nol)

Perhatikan : kontrol TextBox-nya akan menjadi object **txtData(0)**
(lihat bagian Object Selector pada Jendela Properties).

- (3) Klik object txtData(0) pada Form, kemudian klik tombol Copy pada bagian Toolbar.
- (4) Untuk membuat TextBox kedua, klik tombol Paste pada bagian Toolbar. TextBox kedua akan muncul di pojok kiri Form sebagai object txtData(1), aturlah posisinya di dalam Form.
- (5) Lakukan langkah ke-4 di atas sebanyak 4 kali (sesuai dengan jumlah TextBox yang dibutuhkan).
- (6) Di dalam Form akan ada 5 buah TextBox dengan nama yang sama (yaitu txtData) dan masing-masing menjadi object txtData yang dibedakan nomer indexnya (mulai dari 0 s/d 4).

Selanjutnya object-object yang dibuat dengan kontrol array bisa diprogram dengan lebih mudah. Misalnya untuk “mengosongkan” object txtData, bisa menggunakan struktur kontrol For...Next :

```
For i = 0 To 4  
    txtData(i).Text = ""  
Next i
```

Cara ini lebih mudah bila dibandingkan cara “konvensional” berikut :

```
Text1.Text = ""  
Text2.Text = ""  
...  
Text4.Text = ""
```

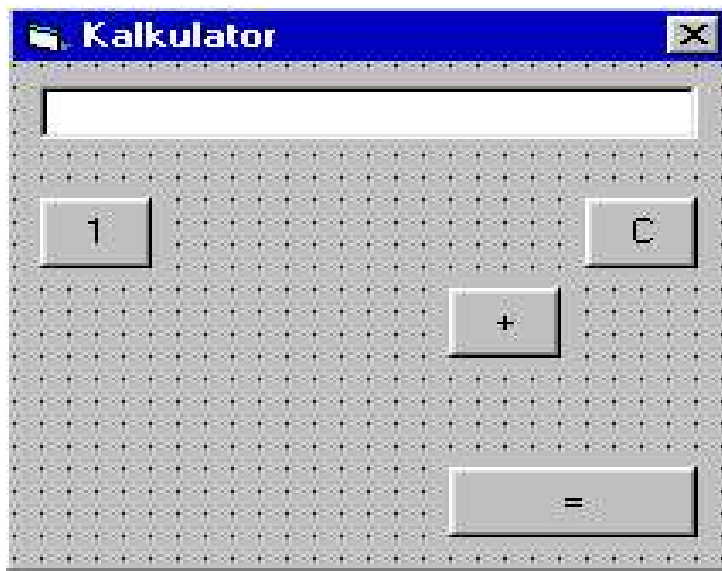
1.3 Contoh Program : Kalkulator

Aktifkan VB 6 melalui tombol Start.

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini :
(Gunakan komponen TextBox dan CommandButton)

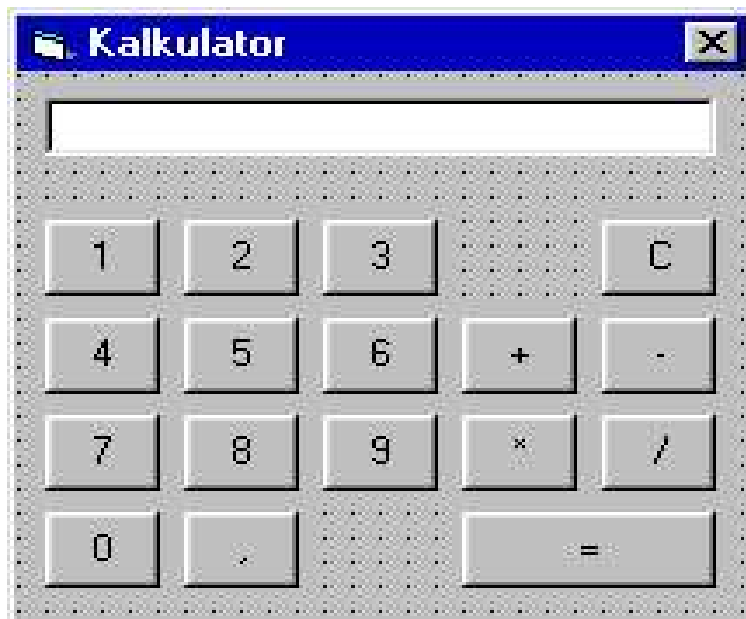


Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form9	BorderStyle Caption StartPosition	1 – Fixed Single Kalkulator 2 – Center Screen
Text1	Alignment Text	1 – Right Justify <kosong>
Command1	Name Caption Index	cmdAngka 1 0
Command2	Name Caption	cmdHapus C
Command3	Name Caption Index	cmdOperator + 0
Command4	Name Caption	cmdHitung =

Buat kontrol array dari cmdAngka (sebanyak 10 buah) dan dari cmdOperator (sebanyak 3 buah). Kemudian atur property caption masing-masing sebagai berikut :

Object	Caption
cmdAngka(1)	2
cmdAngka(2)	3
...	...
cmdAngka(10)	,
cmdOperator(1)	-
cmdOperator(2)	*
cmdOperator(3)	/



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

```
Dim angka(1 To 2) As Single
Dim operator As String

Private Sub cmdAngka_Click(Index As Integer)
    Text1.Text = Text1.Text & cmdAngka(Index).Caption
End Sub

Private Sub cmdOperator_Click(Index As Integer)
    If Text1.Text = "" Then Exit Sub

    angka(1) = CSng(Text1.Text)
    operator = cmdOperator(Index).Caption
    Text1.Text = ""
End Sub

Private Sub cmdHitung_Click()
    Dim hasil As Single

    If Text1.Text = "" Then Exit Sub

    angka(2) = CSng(Text1.Text)

    Select Case operator
    Case "+"
        hasil = angka(1) + angka(2)
    Case "-"
        hasil = angka(1) - angka(2)

    Case "*"
        hasil = angka(1) * angka(2)
    Case "/"
        hasil = angka(1) / angka(2)
    End Select
```

```
Text1.Text = hasil  
End Sub
```

```
Private Sub cmdHapus_Click()  
Text1.Text = ""  
End Sub
```

Simpan Form9 (nama file : **Lat9.frm**).

Catatan : Anda bisa men-download file Lat9.frm dan file latihan-latihan sebelumnya.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form9.

Coba jalankan Project1 :

- Klik salah satu tombol angka, misalnya 5
- Klik salah satu tombol operator, misalnya *
- Klik lagi salah satu tombol angka, misalnya 6
- Klik tombol =
- Akan muncul pada TextBox angka 30 (hasil penghitungan $5 * 6$)
- Klik tombol C untuk menghapus isi TextBox

Penjelasan kode program :

```
Dim angka(1 To 2) As Single  
Dim operator As String  
Private Sub cmdAngka_Click(Index As Integer)  
Text1.Text = Text1.Text & _  
cmdAngka(Index).Caption  
End Sub
```

```
Private Sub cmdOperator_Click(Index As _  
Integer)  
Sub  
If Text1.Text = "" Then Exit  
angka(1) = CSng(Text1.Text)  
operator = cmdOperator(Index).Caption  
Text1.Text = ""  
End Sub
```

Deklarasi variabel-variabel global

Bila user meng-klik salah satu cmdAngka :
Tampilkan angka pada Text1 sesuai dengan caption cmdAngka yang di-klik

Perhatikan : setiap event object yang dibuat dengan kontrol array mempunyai keterangan Index yang menunjukkan nomer index object yang menerima event tersebut.

Bila user meng-klik salah satu cmdOperator :

Cek apakah Text1 kosong, bila True maka proses tidak perlu dilanjutkan
Konversi isi Text1 menjadi Single, lalu simpan ke dalam variabel angka(1) ← variabel array
Simpan caption cmdOperator yang di-klik ke dalam variabel operator (sebagai tanda operasi penghitungan)
Kosongkan Text1


```
Private Sub cmdHitung_Click()  
    Dim hasil As Single  
  
    If Text1.Text = "" Then Exit  
Sub  
    angka(2) = CSng(Text1.Text)  
  
    Select Case operator  
  
    Case "+"  
        hasil = angka(1) + angka(2)  
    Case "-"  
        hasil = angka(1) - angka(2)  
    Case "*"  
        hasil = angka(1) * angka(2)  
    Case "/"  
        hasil = angka(1) / angka(2)  
    End Select  
  
    Text1.Text = hasil  
End Sub  
  
Private Sub cmdHapus_Click()  
    Text1.Text = ""  
End Sub
```

Bila user meng-klik cmdHitung :
Deklarasi variabel lokal

Cek apakah Text1 kosong, bila True maka proses tidak perlu dilanjutkan

Konversi isi Text1 menjadi Single, lalu simpan ke dalam variabel angka(2) ← variabel array

Lakukan penghitungan sesuai dengan isi variabel operator dan hasil penghitungan disimpan dalam variabel hasil :

Operasi penjumlahan

Operasi pengurangan

Operasi perkalian

Operasi pembagian

Tampilkan hasil penghitungan pada Text1

Bila user meng-klik cmdHapus :

Kosongkan Text1

Catatan :

- Program kalkulator ini masih sangat sederhana, sehingga kemungkinan ada *error* ataupun “kejangalan” lainnya.

Cepat Mahir Visual Basic 6.0

Krisna D. Octovhiana
mail4krisna@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Bab 11

Menangani Error

1.1 Jenis-Jenis Error

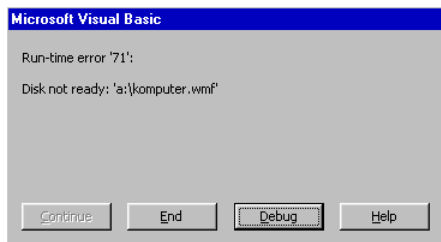
Dalam proses pembuatan program, bisa saja terjadi error yang menyebabkan program tidak berjalan sebagaimana mestinya. Dilihat dari penyebabnya ada 3 jenis error yang bisa terjadi, yaitu :

1. **Syntax error** – adalah error yang disebabkan oleh kesalahan menulis kode program. Misalnya : salah menuliskan nama object, property atau methodnya. Error jenis ini relatif mudah ditangani, IDE VB 6 akan memberi tanda kode program mana yang menimbulkan syntax error.

```
If FileName <> "" Then  
    Image1.Picture = LoadPicture(FileName) ←  
End If  
Exit Sub
```

```
Awal:  
Image1.Pictur = Nothing ←  
FileName = InputBox("Tuliskan nama file gambar :")
```

2. **Runtime error** – adalah error yang disebabkan oleh sistem komputer ketika melakukan sesuatu. Misalnya : menyimpan file ke disket tetapi disketnya tidak ada. Sistem akan “memberitahu” kepada program informasi error yang terjadi. Informasi error yang penting diantaranya adalah nomer error dan deskripsi error. VB 6 “menyimpan” informasi error tersebut pada object Err. Melalui object Err inilah kita bisa menangani runtime error.



```
If FileName <> "" Then  
    Image1.Picture = LoadPicture(FileName) ←  
End If  
Exit Sub
```

3. **Logical error** – adalah error yang disebabkan oleh kesalahan logika pemrograman (dari si programmer). Misalnya : salah meletakkan urutan kode program. Error jenis ini relatif sulit diketahui dan bisa saja baru diketahui setelah program di-*compile* menjadi *executable file* (*.exe). Kejadian seperti ini sering disebut sebagai *bug*.

Pada bab ini akan dibahas khusus bagaimana cara menangani runtime error.

1.2 Menangani Runtime Error

Untuk menangani runtime error, di dalam kode program dibuatkan bagian khusus untuk menangani error yang terjadi. Bagian khusus ini diberi tanda berupa <label>. Agar setiap error yang terjadi dapat ditangani oleh bagian khusus ini, pada awal kode program ditulis perintah berikut ini :

On Error GoTo <label>

Contoh :

On Error GoTo AdaError

```
...  
...  
... } kode-kode program yang  
      mungkin menimbulkan  
      runtime error
```

AdaError:

```
...  
...  
... } kode-kode program untuk  
      menangani runtime error
```

Selanjutnya untuk mengetahui informasi error apa yang terjadi, gunakan object Err. Object Err merupakan object *built-in* yang disediakan VB 6 untuk menangani error. Seperti halnya object yang dibuat dengan kontrol/komponen (misalnya TextBox), maka object Err juga mempunyai property dan method.

Property object Err yang penting diantaranya adalah property Number dan Description. Untuk mengetahui nomer error yang terjadi, gunakan property Number. Sedangkan untuk mengetahui deskripsi error yang terjadi, gunakan property Description.

Dengan menggunakan property Number, kita bisa menangani setiap error dengan cara yang berbeda. Misalnya : untuk setiap error yang terjadi akan dimunculkan kotak pesan dengan bahasa Indonesia. Cara ini dapat Anda lihat pada contoh program.

Setelah error ditangani, kode program dapat dikembalikan ke bagian tertentu dengan perintah Resume. Ada 3 bentuk perintah Resume, yaitu :

- **Resume** – kembali ke kode program yang menimbulkan error. Biasanya digunakan untuk *retry* (mencoba lagi).
- **Resume <label>** – kembali ke bagian tertentu pada kode program yang diberi tanda <label>.
- **Resume Next** – kembali ke kode program berikutnya (kode program setelah kode program yang menimbulkan error). Biasanya digunakan untuk *abort* atau *cancel* (mengabaikan atau membatalkan).

Contoh penggunaan perintah Resume dapat Anda lihat pada contoh program.

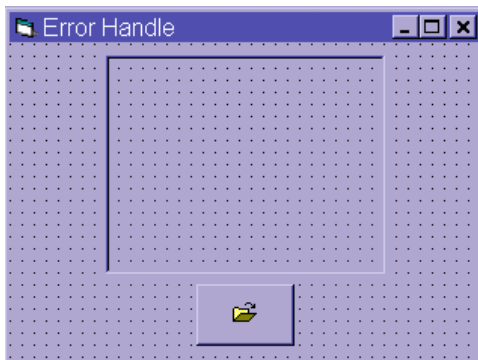
1.3 Contoh Program : Error Handle

Aktifkan VB 6 melalui tombol Start.

Buka kembali project **Latihan.vbp**.

Tambahkan Form baru ke dalam Project.

Pada Jendela Form buatlah UI seperti ini :
(Gunakan komponen Image dan CommandButton)



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form10	Caption StartPosition	Error Handle 2 – Center Screen
Image1	BorderStyle Stretch	1 – Fixed Single True
Command1	Caption Style Picture ToolTipText	<kosong> 1 – Graphical OpenFile.ico Open Picture File

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

```
Private Sub Command1_Click()  
    Dim FileName As String  
  
    Image1.Picture = Nothing  
    FileName = InputBox("Ketikkan path dan nama file gambar :", "Open Picture File", FileName)  
    If FileName <> "" Then  
        Image1.Picture = LoadPicture(FileName)  
    End If  
End Sub
```

Simpan Form10 (nama file : **Lat10.frm**).

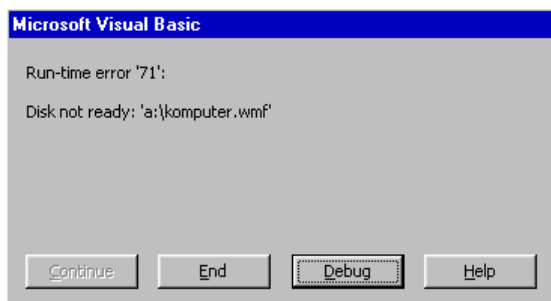
Catatan : Anda bisa men-download file Lat10.frm dan OpenFile.ico.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form10.

Untuk mencoba program ini sebelumnya siapkan sebuah disket dan copy-kan 1 file gambar (misalnya file komputer.wmf pada bab 6).

Coba jalankan Project-nya :

- Klik tombol Open Picture File, akan muncul kotak input.
- Ketikkan lokasi (path) dan nama file gambar, misalnya : a:\komputer.wmf.
- Masukkan disketnya, lalu klik tombol OK.
- Gambar akan muncul → tidak terjadi error.
- Keluarkan disketnya, lalu coba ulangi langkah sebelumnya.
- Akan muncul kotak pesan error :



Klik tombol Debug untuk melihat kode program mana yang menimbulkan runtime error.

- Klik tombol End pada bagian Toolbar untuk menghentikan program.

Agar program bisa menangani error, modifikasi kode programnya menjadi sebagai berikut :

```
Private Sub Command1_Click()  
    Dim FileName As String
```

On Error GoTo AdaError

```
Image1.Picture = Nothing  
FileName = InputBox("Ketikkan path dan nama file gambar :", "Open Picture File", FileName)
```

```
If FileName <> "" Then  
    Image1.Picture = LoadPicture(FileName)  
End If  
Exit Sub
```

AdaError:

```
MsgBox "Error No : " & Err.Number & vbCrLf & _  
    Err.Description, vbCritical + vbOKOnly, Me.Caption  
End Sub
```

Coba jalankan Project-nya

Bila terjadi error seperti sebelumnya akan muncul kotak pesan error :



Klik tombol OK.

Program akan tetap berjalan.

User bisa mengulangi perintah untuk membuka file gambar (disket dimasukkan lebih dulu agar tidak terjadi error lagi).

Agar pesan error-nya menggunakan bahasa Indonesia, modifikasi kode programnya menjadi sebagai berikut :

```
Private Sub Command1_Click()  
    Dim FileName As String  
    Dim ErrMsg As String
```

On Error GoTo AdaError

```
Image1.Picture = Nothing  
FileName = InputBox("Ketikkan path dan nama file gambar :", "Open Picture File", FileName)  
If FileName <> "" Then  
    Image1.Picture = LoadPicture(FileName)  
End If  
Exit Sub
```

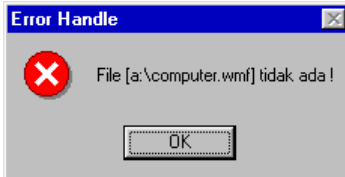
AdaError:

```
Select Case Err.Number  
Case 53  
    ErrMsg = "File [" & FileName & "] tidak ada !"  
Case 71  
    ErrMsg = "Disket belum dimasukkan !"
```

```
Case Else
    ErrMsg = Err.Description
End Select
MsgBox ErrMsg, vbCritical + vbOKOnly, Me.Caption
End Sub
```

Coba jalankan Project-nya

Bila terjadi error akan muncul kotak pesan error :



Klik tombol OK.
Program akan tetap berjalan.
Lakukan *try-and-error* untuk melihat setiap nomer dan deskripsi error yang mungkin terjadi.

Agar setelah error ditangani kode program dapat dikembalikan ke bagian tertentu, modifikasi lagi kode programnya menjadi sebagai berikut :

```
Private Sub Command1_Click()
    Dim FileName As String
    Dim ErrMsg As String
    Dim Ask As VbMsgBoxResult

    On Error GoTo AdaError

    Awal:
    Image1.Picture = Nothing
    FileName = InputBox("Ketikkan path dan nama file gambar :", "Open Picture File", FileName)
    If FileName <> "" Then
        Image1.Picture = LoadPicture(FileName)
    End If
    Exit Sub

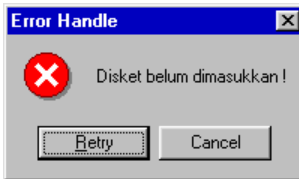
AdaError:
    Select Case Err.Number
    Case 53
        ErrMsg = "File [" & FileName & "] tidak ada !"
    Case 71
        ErrMsg = "Disket belum dimasukkan !"
    Case Else
        ErrMsg = Err.Description
    End Select

    Ask = MsgBox(ErrMsg, vbCritical + vbRetryCancel, Me.Caption)

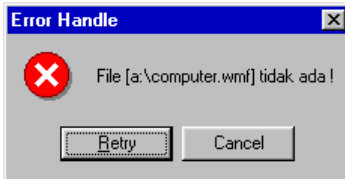
    Select Case Ask
    Case vbRetry
        If Err.Number = 53 Then Resume Awal Else Resume
    Case vbCancel
        Resume Next
    End Select
End Sub
```

Coba jalankan Project-nya

Bila terjadi error akan muncul kotak pesan error :



Klik tombol Retry untuk mencoba lagi (disketnya dimasukkan lebih dulu),
atau
klik tombol Cancel untuk membatalkan.



Klik tombol Retry untuk mencoba lagi (lalu ganti nama filenya),
atau
klik tombol Cancel untuk membatalkan.

Cepat Mahir Visual Basic 6.0

Krisna D. Octovhiana
mail4krisna@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Bab 12

Penggunaan Procedure

1.1 Apa itu Procedure ?

Procedure adalah blok kode program yang berisi perintah-perintah untuk mengerjakan tugas tertentu. Bila di dalam kode program yang kita buat ada perintah-perintah untuk melakukan tugas yang sama di beberapa tempat, maka akan lebih baik perintah-perintah tersebut dibuat dalam sebuah procedure. Kemudian, procedure itu bisa di-‘panggil’ bila diperlukan.

Penggunaan procedure sangat menghemat penulisan kode program, karena kode-kode program yang sama di beberapa tempat cukup dibuat pada satu bagian saja. Selain itu, procedure akan memudahkan perbaikan kode program bila terjadi perubahan atau kesalahan, karena perbaikan cukup dilakukan pada satu bagian saja.

1.2 Jenis-jenis Procedure

Pada VB6 ada 4 jenis procedure, yaitu :

- **Procedure Sub** – procedure yang tidak mengembalikan nilai setelah ‘tugas’-nya selesai.
- **Procedure Function** – procedure yang mengembalikan nilai setelah ‘tugas’-nya selesai.
- **Procedure Event** – procedure untuk suatu event pada sebuah object. Digunakan di dalam class module.
- **Procedure Property** – procedure untuk mengubah (*let*) atau mengambil (*get*) nilai property pada sebuah object. Digunakan di dalam class module.

Pada bab ini akan dibahas penggunaan procedure sub dan function. Procedure event dan property akan dibahas pada bab selanjutnya.

Bentuk penulisan (*syntax*) procedure sub :

```
[Public | Private] Sub <nama_sub> ([<argumen>])  
    ...  
    <isi procedure>  
    ...  
End Sub
```

Sedangkan bentuk penulisan (*syntax*) procedure function :

```
[Public | Private] Function <nama_function> ([<argumen>]) As <tipe_data>  
    ...  
    <isi procedure>  
    ...  
End Function
```

Pernyataan [Public | Private] menentukan ruang lingkup (*scope*) procedure. Sebuah procedure dengan scope public bisa digunakan dalam lingkup project. Sedangkan procedure dengan scope private hanya bisa digunakan dalam lingkup form saja.

<nama_sub> atau <nama_function> dibuat sebagai pengenal procedure saat di-‘panggil’. Aturan penamaan sebuah procedure sama dengan aturan penamaan sebuah variabel. Nama sebuah procedure dibuat unik, tidak boleh ada yang sama.

<argumen> merupakan serangkaian nilai dan tipe data yang dipakai oleh procedure untuk mengerjakan ‘tugas’-nya. Sebuah procedure bisa saja tidak memakai argumen sama sekali.

Pernyataan As <tipe_data> pada procedure function menentukan tipe data nilai yang akan dikembalikan (*return value*) setelah ‘tugas’-nya selesai.

Untuk menggunakan sebuah procedure, maka procedure tersebut harus di-‘panggil’ pada bagian tertentu dari kode program. Procedure sub di-‘panggil’ dengan pernyataan :

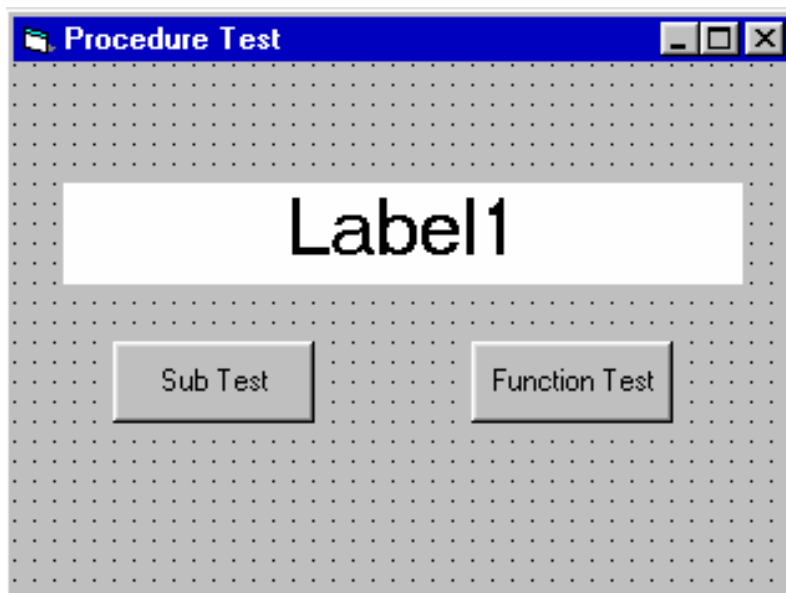
```
Call <nama_sub>([<argumen>])
```

Sedangkan procedure function bisa di-‘panggil’ langsung dengan menyisipkannya di dalam kode program yang memanggilnya.

Procedure boleh ditulis dimana saja dalam kode program, tetapi biasanya ditulis dibagian atas atau bawah agar mudah ditangani.

1.3 Contoh Program : Procedure Test

1. Aktifkan VB 6 melalui tombol Start.
2. Buka kembali project **Latihan.vbp**.
3. Tambahkan Form baru ke dalam Project.
4. Pada Jendela Form buatlah UI seperti ini :
5. (Gunakan komponen Label dan CommandButton)



6. Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form11	Caption StartPosition	Procedure Test 2 – Center Screen
Label1	Alignment BackColor Font ToolTipText	2 – Center Palette : <putih> Size : 24 Dobel-Klik di Sini
Command1	Caption ToolTipText	Sub Test Klik di Sini
Command2	Caption	Function Test

7. Buka Jendela Code, lalu pada bagian Code Editor ketikkan kode programnya sebagai berikut :

```
Private Sub TulisTeks(teks As String, warna As ColorConstants)
    With Label1
        .Caption = teks
        .ForeColor = warna
    End With
End Sub

Private Function JumlahAngka() As String
    Dim angka1 As String, angka2 As String
    Dim hasil As Single

    angka1 = InputBox("Tulis angka 1 :", "Jumlah Angka")
    angka2 = InputBox("Tulis angka 2 :", "Jumlah Angka")

    If angka1 <> "" And angka2 <> "" Then
        hasil = CSng(angka1) + CSng(angka2)
        JumlahAngka = Cstr(hasil)
    End If
End Function

Private Sub Label1_DblClick()
    Call TulisTeks("Hai", vbBlue)
End Sub

Private Sub Command1_Click()
    Call TulisTeks("Hallo", vbRed)
End Sub

Private Sub Command2_Click()
    Label1.Caption = "Jumlah = " & JumlahAngka()
End Sub
```

8. Simpan Form11 (nama file : **Lat11.frm**).

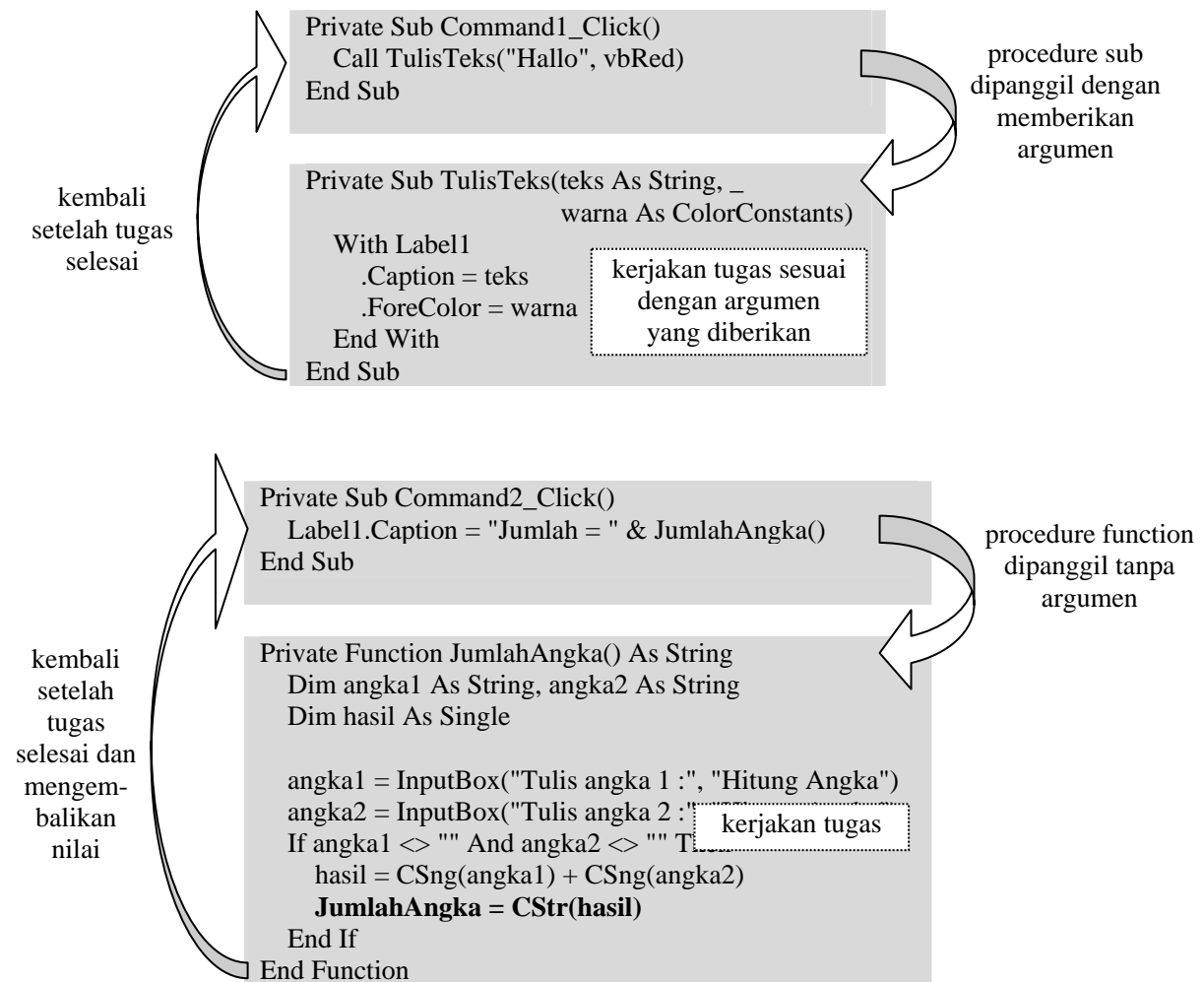
Catatan : Anda bisa men-download file Lat1-11.zip.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form11.

Coba jalankan Project-nya :

- Dobel-klik Label1, akan muncul tulisan “Hai” dengan warna biru.
- Klik Command1, akan muncul tulisan “Hallo” dengan warna merah.
- Klik Command2, akan muncul kotak input.
- Ketikkan angka 5, lalu klik tombol OK.
- Ketikkan angka 10, lalu klik tombol OK.
- Akan muncul tulisan “Jumlah = 15”.

Proses ketika procedure di-'panggil' :



Dapat anda lihat bahwa prosedur `TulisTeks` dipanggil, dan dimasukkan 2 variabel ke dalamnya yaitu `teks` dan `warna`. Selanjutnya kedua argumen tersebut diolah oleh prosedur `TulisTeks`. Akhirnya ditampilkan tulisan dan warnanya yang merupakan hasil dari properti `Caption` dan `ForeColor` dari obyek label.

1.4 Pernyataan Optional

Nilai argumen di dalam sebuah procedure dapat dibuat menjadi optional (boleh ada atau tidak ada). Bila nilai argumen yang optional tidak ada, maka yang digunakan adalah nilai default-nya. Untuk membuat nilai argumen menjadi optional, tambahkan pernyataan **Optional** di depan argumen-nya dan berikan nilai default-nya, contoh :

```
Private Sub TulisTeks(teks As String, Optional warna As ColorConstants =  
vbMagenta)  
    With Label1  
        .Caption = teks  
        .ForeColor = warna  
    End With  
End Sub
```

Bila nilai argumen warna tidak ada, maka warna teks pada Label1 adalah magenta, contoh :

```
Call TulisTeks("Hallo")
```

1.5 Pernyataan ByRef dan ByVal

Nilai argumen yang dikirimkan ke dalam procedure bisa saja berasal dari sebuah variabel. Namun demikian, perlu diperhatikan bahwa nilai argumen yang dikirimkan ke dalam procedure bisa berdasarkan referensi (ByRef) atau berdasarkan nilai (ByVal). Default-nya nilai argumen dikirimkan berdasarkan referensi. Apabila variabel dikirimkan berdasarkan referensi, maka semua perubahan yang dilakukan terhadap argumen akan mempengaruhi variabel tersebut. Hal ini terkadang menimbulkan *logical error*. Sebagai ilustrasi perhatikan contoh berikut ini :

Dim Harga As Currency, Total As Currency

```
Private Sub HitungDiskon(HargaAwal As Currency, Diskon As Single)  
    HargaAwal = HargaAwal * (1 - Diskon)  
    Total = CCur(HargaAwal)  
End Sub  
  
Private Sub Command3_Click()  
    Harga = 10000  
    Total = 0  
    Call HitungDiskon(Harga, 0.1)  
    MsgBox "Harga " & Harga & " setelah diskon 10% menjadi " & Total  
End Sub
```

Hasilnya → **Harga 9000 setelah diskon 10% menjadi 9000**

Variabel Harga dikirimkan berdasarkan referensi (ByRef), sehingga perubahan pada argumen HargaAwal akan mempengaruhi variabel Harga.

Modifikasi procedure sub HitungDiskon menjadi sebagai berikut :

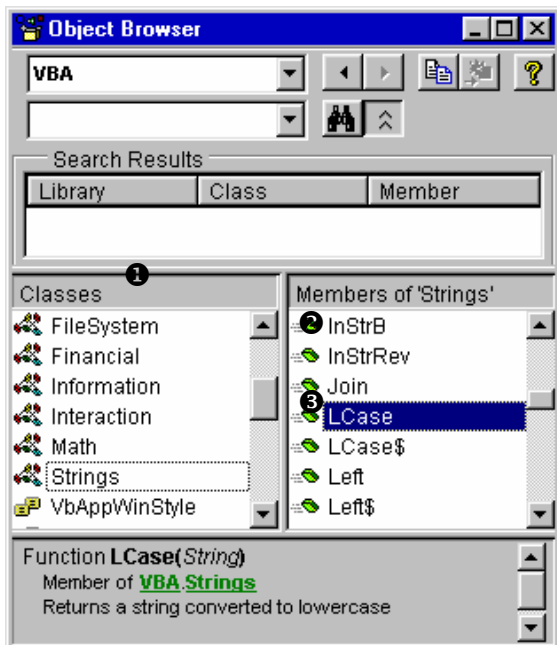
```
Private Sub HitungDiskon(ByVal HargaAwal As Currency, Diskon As Single)  
    HargaAwal = HargaAwal * (1 - Diskon)  
    Total = CCur(HargaAwal)  
End Sub
```

Hasilnya → **Harga 10000 setelah diskon 10% menjadi 9000**

Sekarang variabel Harga dikirimkan berdasarkan nilai (ByVal), sehingga perubahan pada argumen HargaAwal tidak akan mempengaruhi variabel Harga.

1.6 Procedure Built-in

VB6 menyediakan cukup banyak procedure yang siap-pakai (*built-in*) untuk berbagai ‘tugas’. Coba klik menu View – Object Browser atau tekan tombol F2 pada keyboard :



1. Cari module Strings pada bagian Classes.
2. Lihat daftar pada bagian Member of ‘Strings’ yang merupakan kumpulan procedure dengan ‘tugas’ pengolahan data string.
3. Klik nama procedure untuk mengetahui apa ‘tugas’-nya.

Cepat Mahir Visual Basic 6.0

Krisna D. Octovhiana
mail4krisna@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Bab 13

Penanganan Keyboard

13.1. Mengenal Event Keyboard

Event keyboard merupakan salah satu elemen utama dari interaksi antara *user* dengan program yang kita buat. Event keyboard terjadi saat *user* menekan (*pressed*) ataupun melepas (*released*) tombol pada keyboard.

Menangani event keyboard dapat dilakukan pada 2 (dua) level, yaitu :

1. Pada **level kontrol** (*low-level*) – menangani event keyboard yang terjadi pada sebuah kontrol, misalnya ketika *user* mengetik pada sebuah TextBox. Tidak semua kontrol mempunyai event keyboard. Hanya kontrol yang bisa mendapatkan focus (dicirikan dengan property TabIndex dan TabStop) saja yang mempunyai event keyboard.
2. Pada **level form** (*hight-level*) – menangani event keyboard yang terjadi pada lingkup sebuah form. Artinya form akan lebih dulu merespon event keyboard daripada kontrol-kontrol yang ada di dalamnya. Agar form selalu lebih dulu merespon event keyboard maka property KeyPreview pada form tersebut harus diset menjadi True.

13.2. Jenis-jenis Event Keyboard

VB6 menyediakan 3 (tiga) jenis event pada form dan pada beberapa kontrol yang bisa menerima input dari *user* melalui keyboard, yaitu :

1. Event **KeyPress** – terjadi ketika tombol-tombol yang mempunyai kode ASCII pada keyboard ditekan. ASCII (*American Standard Code for Information Interchange*) adalah kode dari sekumpulan karakter pada tombol keyboard yang terdiri dari : abjad, angka dan beberapa karakter khusus (Enter, Escape, Tab, Backspace).
2. Event **KeyDown** – terjadi ketika setiap tombol pada keyboard ditekan.
3. Event **KeyUp** – terjadi ketika setiap tombol pada keyboard dilepas.

Perbedaan antara event KeyPress dengan KeyDown adalah :

- Event KeyPress hanya berlaku untuk tombol-tombol yang mempunyai kode ASCII saja. Tombol-tombol tertentu - seperti : tombol fungsi (F1 s/d F12), tombol panah, tombol keypad – tidak mempunyai kode ASCII.
- Event KeyPress tidak bisa merespon penekanan tombol yang di kombinasi dengan Shift, Ctrl dan Alt.

Bila event KeyPress terjadi maka event tersebut akan mengembalikan nilai dari argumen **KeyAscii** yaitu kode ASCII dari tombol keyboard yang ditekan. Contoh :

```
Private Sub Form1_KeyPress (KeyAscii As Integer)
    MsgBox "Kode ASCII tombol yang ditekan : " & KeyAscii
End Sub
```

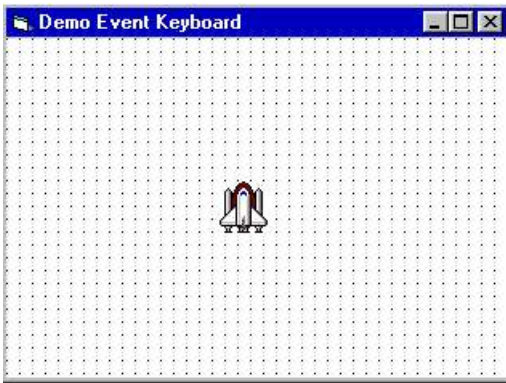
Sedangkan event KeyDown dan KeyUp akan mengembalikan nilai dari argumen **KeyCode** dan **Shift**. Argumen KeyCode berisi kode tombol keyboard yang ditekan dan argumen Shift berisi kode penekanan tombol Shift, Ctrl dan Alt. Konstanta nilai untuk kedua argumen tersebut dapat dilihat melalui jendela Object Browser. Contoh :

```
Private Sub Form1_KeyDown(KeyCode As Integer, Shift As Integer)
    MsgBox "Kode tombol yang ditekan : " & KeyCode
End Sub
```

```
Private Sub Form1_KeyUp(KeyCode As Integer, Shift As Integer)
    MsgBox "Kode tombol yang dilepas : " & KeyCode
End Sub
```

13.3. Contoh Program : “Demo Event Keyboard”

- (1) Aktifkan IDE VB6.
- (2) Bukalah kembali file project **Latihan.vbp**.
- (3) Tambahkan Form baru ke dalam project.
- (4) Pada Jendela Form buatlah UI seperti ini :
(Gunakan kontrol PictureBox)



Pengaturan property setiap object-nya adalah sebagai berikut :

Object	Properties	Value
Form12	BackColor Caption KeyPreview ScaleMode StartPosition	Palette: <putih> Demo Event Keyboard True 3 – Pixel 2 – Center Screen
Picture1	Name AutoSize BackColor BorderStyle Picture ScaleMode	picRoket True Palette: <putih> 0 – None Roket.ico 3 – Pixel

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

```
'Deklarasi variabel global
Dim xpos As Single, ypos As Single

Private Sub Form_Load()
    'Pindahkan posisi roket ke tengah form
    xpos = (Me.ScaleWidth - picRoket.Width) / 2
    ypos = (Me.ScaleHeight - picRoket.Height) / 2
    picRoket.Move xpos, ypos
End Sub

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    'Cek tombol keyboard yang ditekan
    Select Case KeyCode
        Case vbKeyLeft           'Tombol panah kiri
            Call RoketKeKiri
        Case vbKeyRight          'Tombol panah kanan
            Call RoketKeKanan
    End Select
End Sub

Private Sub Form_KeyUp(KeyCode As Integer, Shift As Integer)
    'Cek tombol keyboard yang dilepas
    Select Case KeyCode
        Case vbKeyUp             'Tombol panah atas
            Call RoketKeAtas
        Case vbKeyDown           'Tombol panah bawah
```

```
        Call RaketKeBawah
    End Select
End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)
    'Cek tombol keyboard yang ditekan
    If KeyAscii = vbKeyEscape Then 'Tombol Escape
        If MsgBox("Tutup program ?", vbQuestion + vbYesNo, _
            Me.Caption) = vbYes Then Unload Me
    End If
End Sub

Private Sub RaketKeKiri()
    xpos = xpos - 10          'Geser ke kiri 10 pixel
    If xpos < 0 Then          'Cek bila melewati batas kiri
        xpos = 0
    End If
    picRaket.Move xpos       'Pindahkan posisi roket
End Sub

Private Sub RaketKeKanan()
    xpos = xpos + 10

    'Geser ke kanan 10 pixel
    If xpos > Me.ScaleWidth - picRaket.Width Then 'Cek bila melewati batas
                                                    kanan
        xpos = Me.ScaleWidth - picRaket.Width
    End If
    picRaket.Move xpos

    'Pindahkan posisi roket
End Sub

Private Sub RaketKeAtas()
    ypos = ypos - 10          'Geser ke atas 10 pixel
    If ypos < 0 Then          'Cek bila melewati batas atas
        ypos = 0
    End If
    picRaket.Move xpos, ypos 'Pindahkan posisi roket
End Sub

Private Sub RaketKeBawah()
    ypos = ypos + 10

    'Geser ke bawah 10 pixel
    If ypos > Me.ScaleHeight - picRaket.Height Then 'Cek bila melewati batas
                                                    bawah
        ypos = Me.ScaleHeight - picRaket.Height
    End If
    picRaket.Move xpos, ypos

    'Pindahkan posisi roket
End Sub
```

Simpan Form12 (nama file : **Lat12.frm**).

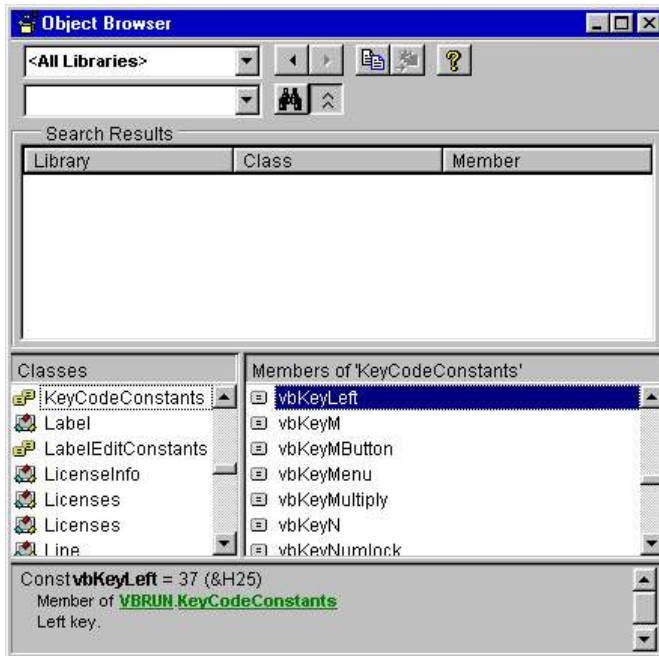
Catatan : Anda bisa men-download file Lat1-12.zip.

Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form12.

Coba jalankan Project1 :

- Saat form muncul posisi roket ada di tengah form.
- Coba gerakkan roket menggunakan tombol-tombol panah pada keyboard.
- **Perhatikan** : roket akan bergerak ke kiri/kanan saat tombol ditekan dan roket akan bergerak ke atas/bawah saat tombol dilepas.
- Untuk menutup form, tekan tombol Esc(ape).

Melihat nilai konstanta (dari argumen KeyAscii atau KeyCode) untuk tombol-tombol pada keyboard :



1. Klik menu View – Object Browser atau tekan tombol F2 pada keyboard.
2. Cari KeyCodeConstants pada bagian Classes.
3. Lihat daftar pada bagian Member of 'KeyCodeConstants', inilah kumpulan konstanta untuk setiap tombol pada keyboard.
4. Klik nama konstanta dan lihat pada bagian Description untuk mengetahui tombol apa yang 'diwakilinya'.
5. Klik tombol Copy untuk menyalin konstanta ke dalam kode program.

Agar bisa merespon kombinasi tombol Ctrl, modifikasi kode program sebagai berikut :

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    'Cek tombol keyboard yang ditekan
    Select Case KeyCode
    Case vbKeyLeft 'Tombol panah kiri
        If Shift = vbCtrlMask Then 'Kombinasi dgn tombol Ctrl
            Call RoketKeSisiKiri
        Else
            Call RoketKeKiri
        End If
    Case vbKeyRight 'Tombol panah kanan
        If Shift = vbCtrlMask Then 'Kombinasi dgn tombol Ctrl
            Call RoketKeSisiKanan
        Else
            Call RoketKeKanan
        End If
    End Select
End Sub
```

```
Private Sub RoketKeSisiKiri()
    xpos = 0 'Geser ke sisi kiri
    picRoket.Move xpos, ypos 'Pindahkan posisi roket
End Sub
```

```
Private Sub RokatKeSisiKanan()  
    xpos = Me.ScaleWidth - picRokat.Width      `Geser ke sisi kanan  
    picRokat.Move xpos, ypos                  `Pindahkan posisi rokat  
End Sub
```

Nilai konstanta (dari argumen Shift) untuk kombinasi tombol-tombol Shift, Ctrl dan Alt :

- **vbShiftMask** : kombinasi dengan tombol Shift.
- **vbCtrlMask** : kombinasi dengan tombol Ctrl.
- **vbAltMask** : kombinasi dengan tombol Alt.

Cepat Mahir Visual Basic 6.0

Krisnha D. Oktovhiana
mail4krisna@yahoo.com

Lisensi Dokumen:

Copyright © 2005 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Bab 14

Menggunakan Drag-Drop

14.1. Apa Itu Drag-Drop?

Drag-Drop merupakan istilah umum di dalam penggunaan mouse untuk menggeser, menyalin atau memindahkan gambar, teks, file, dll. Menggunakan *drag-drop* akan mempermudah *user* saat menggunakan sebuah program.

14.2. Drag-Drop dengan VB 6.0

VB 6.0 menyediakan beberapa property, method dan event yang berhubungan dengan *drag-drop*. Operasi *drag-drop* melibatkan object source dan object target. Setiap object di dalam form bisa menjadi source ataupun target (termasuk form itu sendiri).

Property, method dan event yang berhubungan dengan *drag-drop* adalah sebagai berikut:

Property	Object	Keterangan
DragMode	Source	Menentukan bagaimana <i>drag-drop</i> dimulai: - Manual (vbManual = 0, default) atau - Otomatis (vbAutomatic = 1)
DragIcon	Source	Menentukan bentuk pointer saat operasi <i>drag-drop</i> berlangsung.
Method	Object	Keterangan
Drag [action]	Source	Digunakan untuk memulai <i>drag-drop</i> secara manual (property DragMode diset vbManual).
Event	Object	Keterangan
DragOver	Target	Saat object source di- <i>drag</i> melewati object target.
DragDrop	Target	Saat object source di-drop pada object target.

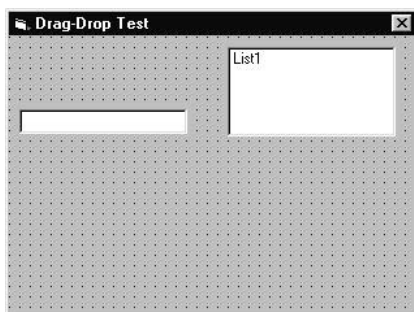
Nilai untuk argumen action adalah sebagai berikut:

Nilai	Keterangan
vbBeginDrag = 1	Memulai operasi <i>drag-drop</i>
vbCancel = 0	Membatalkan operasi <i>drag-drop</i>
vbEndDrag = 2	Mengakhiri operasi <i>drag-drop</i>

Ada tiga argumen pada event-event *drag-drop*, yaitu: **Source**, **X**, **Y** dan **State**. Argumen Source menunjukkan object yang menjadi source. Argumen X dan Y menunjukkan posisi koordinat pointer mouse. Sedangkan argumen State menunjukkan status pointer pada saat event DragOver, nilainya terdiri dari: 0 → pointer masuk ke dalam object target, 1 → pointer meninggalkan object target dan 2 → pointer bergerak di dalam object target.

14.3. Contoh Program: Drag-Drop Test

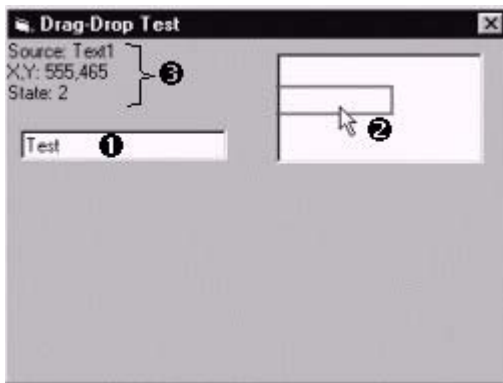
1. Aktifkan program Microsoft Visual Basic 6.0.
2. Bukalah kembali file project **Latihan.vbp**.
3. Tambahkan form baru ke dalam project.
4. Pada Jendela Object buatlah *User Interface* seperti ini:



Setting property object-nya adalah sebagai berikut:

Object	Properties	Value
Form14	BorderStyle Caption StartPosition	1 – Fixed Single Drag and Drop Test 2 – Center Screen
Text1	DragMode Text	1 – Automatic <kosong>

5. Buka Jendela Code, lalu ketikkan kode programnya seperti pada Listing-14A.
6. Simpanlah Form14 dengan nama file Lat14.frm.
Catatan:
 Anda bisa men-download file vb6-latihan.zip.
7. Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form14.
8. Coba jalankan project-nya:



Keterangan:

1. Ketikkan sesuatu pada Text1.
2. Lakukan *drag-drop* ke List1.
3. Perhatikan nilai argumen pada event DragOver.

9. Ubahlah property object Text1 sebagai berikut:

Object	Properties
Text1	DragIcon DragMode

10. Modifikasi kode programnya seperti pada Listing-14B.
11. Coba jalankan kembali project-nya.

Listing-14A

```
Private Sub List1_DragOver(Source As Control, X As Single, Y As Single, _
                          State As Integer)
    Me.Cls
    Me.Print "Source: " & Source.Name
    Me.Print "X,Y: " & X & ", " & Y
    Me.Print "State: " & State
End Sub

Private Sub List1_DragDrop(Source As Control, X As Single, Y As Single)
    If Source.Name = "Text1" Then
        List1.AddItem Text1.Text
    End If
    Me.Cls
End Sub

Private Sub Form_DragOver(Source As Control, X As Single, Y As Single, _
                          State As Integer)
    Me.Cls
End Sub

Private Sub Form_DragDrop(Source As Control, X As Single, Y As Single)
    Me.Cls
End Sub
```

Listing-14B

```
Private Sub List1_DragOver(Source As Control, X As Single, Y As Single, _
                          State As Integer)
    Me.Cls
    Me.Print "Source: " & Source.Name
    Me.Print "X,Y: " & X & ", " & Y
    Me.Print "State: " & State
    If Source.Name = "Text1" And State = 0 Then
        Text1.DragIcon = LoadPicture(App.Path & "DropYes.cur")
    End If
End Sub

Private Sub List1_DragDrop(Source As Control, X As Single, Y As Single)
    If Source.Name = "Text1" Then
        List1.AddItem Text1.Text
        Text1.Drag vbEndDrag
    End If
    Me.Cls
End Sub

Private Sub Form_DragOver(Source As Control, X As Single, Y As Single, _
                          State As Integer)
    Me.Cls
    If Source.Name = "Text1" And State = 0 Then
        Text1.DragIcon = LoadPicture(App.Path & "DropNo.cur")
    End If
End Sub

Private Sub Form_DragDrop(Source As Control, X As Single, Y As Single)
    If Source.Name = "Text1" Then
        Text1.Drag vbCancel
    End If
    Me.Cls
End Sub

Private Sub Text1_MouseDown(Button As Integer, Shift As Integer, _
                             X As Single, Y As Single)
    If Text1.Text <> "" And Button = vbLeftButton Then
        Text1.Drag vbBeginDrag
    End If
End Sub
```

Cepat Mahir Visual Basic 6.0

Krisnha D. Oktovhiana
mail4krisna@yahoo.com

Lisensi Dokumen:

Copyright © 2005 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Bab 15

Menggunakan OLE Drag-Drop

15.1. Apa Itu OLE Drag-Drop?

OLE (Object Linking and Embedding) Drag-Drop adalah jenis *drag-drop* yang memungkinkan *user* untuk menyalin atau memindahkan data dari satu bagian ke bagian yang lain di dalam satu program atau dengan program yang lain. Sebagian besar program-program buatan Microsoft (seperti Microsoft Office) ataupun perusahaan lain (seperti Adobe PhotoShop) mendukung penggunaan *OLE drag-drop*.

15.2. OLE Drag-Drop dengan VB 6.0

VB 6.0 menyediakan beberapa property, method dan event yang berhubungan dengan *OLE drag-drop*. Operasi *OLE drag-drop* melibatkan object source dan object target. Hanya beberapa komponen pada VB 6.0 yang bisa digunakan sebagai object source maupun target, yaitu: TextBox, ComboBox, ListBox, Image, PictureBox, DirListBox dan FileListBox. Sedangkan komponen lainnya hanya bisa digunakan sebagai object target, seperti: CommandButton, CheckBox, OptionButton, Label dan Form.

Property, method dan event yang berhubungan dengan *OLE drag-drop* adalah sebagai berikut:

Property	Object	Keterangan
OLEDragMode	Source	Menentukan bagaimana <i>OLE drag</i> dimulai: - Manual (vbOLEDragManual = 0, default) atau - Otomatis (vbOLEDragAutomatic = 1)
OLEDropMode	Target	Menentukan bagaimana <i>OLE drop</i> dilakukan: - Diabaikan (vbOLEDropNone = 0, default) - Manual (vbOLEDropManual = 1) atau - Otomatis (vbOLEDropAutomatic = 2)
Method	Object	Keterangan
OLEDrag	Source	Digunakan untuk memulai <i>OLE drag</i> secara manual (property OLEDragMode diset vbOLEDragManual).
Event	Object	Keterangan
OLEStartDrag	Source	Saat operasi <i>OLE drag-drop</i> dimulai.
OLECompleteDrag	Source	Saat proses <i>drop</i> pada object target selesai.
OLEDragOver	Target	Saat pointer mouse melewati object target selama operasi <i>OLE drag-drop</i> berlangsung.
OLEDragDrop	Target	Saat data di- <i>drop</i> pada object target.

Ada tiga argumen penting pada event-event *OLE drag-drop*, yaitu: **AllowedEffects**, **Effect** dan **Data**. Argumen AllowedEffects dan Effect digunakan untuk menentukan efek *OLE drag-drop* yang diperbolehkan atau yang digunakan. Nilai untuk kedua argumen tersebut adalah sebagai berikut:

Nilai	Keterangan
vbDropEffectNone = 0	Data tidak bisa diterima oleh object target.
vbDropEffectCopy = 1	Data disalin ke object target.
vbDropEffectMove = 2	Data dipindahkan ke object target.

Sedangkan argumen data merupakan object DataObject yang digunakan selama operasi *OLE drag-drop* berlangsung.

15.3. Object DataObject

Untuk menangani *OLE drag-drop* secara manual digunakan object DataObject untuk menyimpan data dan format datanya. Property dan method yang dimiliki object DataObject adalah:

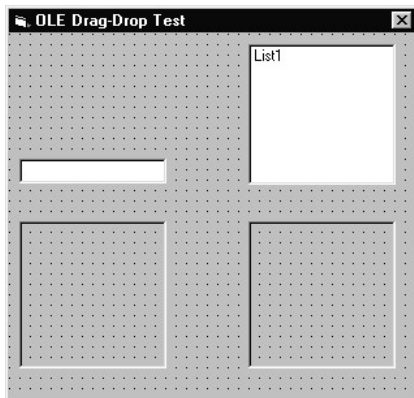
Property	Keterangan
Files	Koleksi nama-nama file (format datanya vbCFFiles)
Method	Keterangan
Clear	Menghapus semua data dan format datanya.
GetData(format)	Mengambil data yang tersimpan. Argumen format digunakan untuk menentukan format data yang diambil.
GetFormat(format)	Mengetahui format data yang tersimpan. Akan menghasilkan nilai True (bila formatnya sesuai) atau False (bila formatnya tidak sesuai).
SetData value, [format]	Menyimpan data dengan format tertentu. Argumen value adalah data yang akan disimpan.

Nilai untuk argumen format adalah sebagai berikut:

Nilai	Keterangan
Dikosongkan = 0	Format data akan ditentukan otomatis berdasarkan datanya.
vbCFText = 1	Data teks
vbCFBitmap = 2	Data bitmap (gambar)
vbCFMetafile = 3	Data metafile (gambar)
vbCFDIB = 8	Data device-independent bitmap (gambar)
vbCFFiles = 15	Data file

15.4. Contoh Program: OLE Drag-Drop Test

1. Aktifkan program Microsoft Visual Basic 6.0.
2. Bukalah kembali file project **Latihan.vbp**.
3. Tambahkan form baru ke dalam project.
4. Pada Jendela Object buatlah *User Interface* seperti ini:

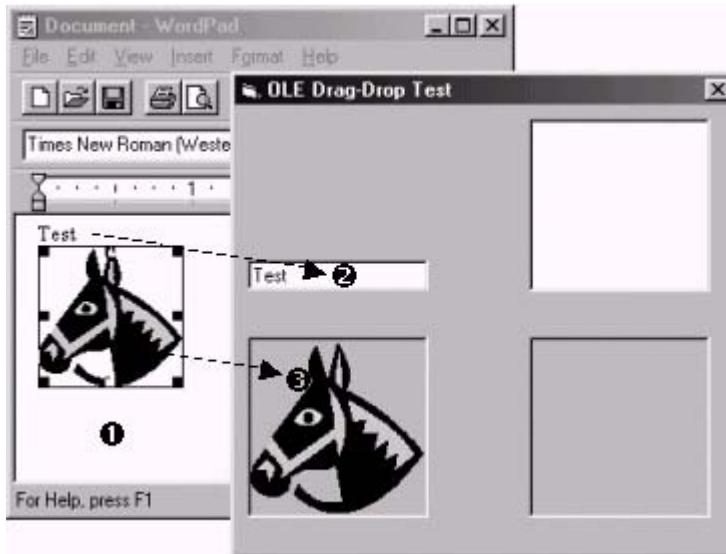


Setting property object-nya adalah sebagai berikut:

Object	Properties	Value
Form15	BorderStyle Caption	1 – Fixed Single OLE Drag-Drop Test
Text1	OLEDragMode OLEDropMode Text	1 – Automatic 2 – Automatic <kosong>
Image1	OLEDragMode OLEDropMode Stretch	1 – Automatic 2 – Automatic True

5. Simpanlah Form15 dengan nama file Lat15.frm.
Catatan:
 Anda bisa men-download file vb6-latihan.zip.

- Klik menu Project > Project1 Properties lalu klik tab General. Gantilah Startup Object-nya menjadi Form15.
- Coba jalankan project-nya dan aktifkan program WordPad:



Keterangan:

- Buatlah teks dan gambar pada program WordPad.
- Lakukan *drag-drop* teks dari WordPad ke Text1.
- Lakukan *drag-drop* gambar dari WordPad ke Image1.

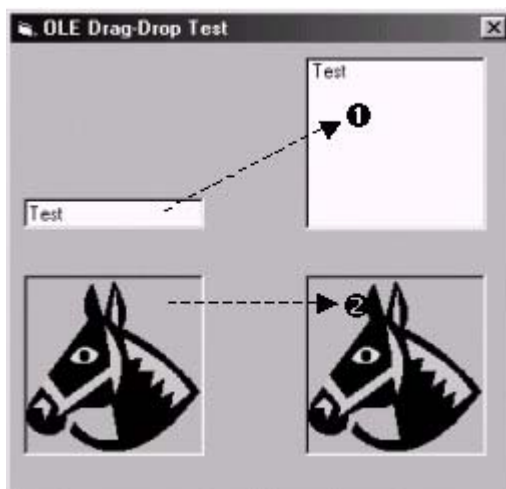
Catatan:

- *Drag-drop tanpa* tombol Ctrl → teks/gambar akan dipindahkan.
- *Drag-drop dengan* tombol Ctrl → teks/gambar akan disalin.

- Ubahlah property object-object berikut ini:

Object	Properties	Value
Text1	OLEDragMode	0 – Manual
List1	OLEDropMode	1 – Manual
Image1	OLEDragMode	0 – Manual
Image2	OLEDropMode Stretch	1 – Manual True

- Buka Jendela Code, lalu ketikkan kode programnya seperti pada Listing-15.
- Coba jalankan kembali project-nya:



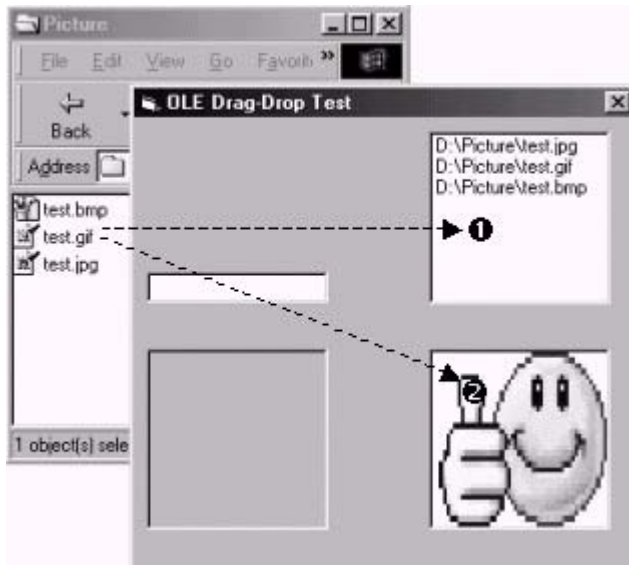
Keterangan:

- Lakukan *drag-drop* teks dari Text1 ke List1.
- Lakukan *drag-drop* gambar dari Image1 ke Image2.

Catatan:

- *Drag-drop tanpa* tombol Ctrl → teks/gambar akan dipindahkan.
- *Drag-drop dengan* tombol Ctrl → teks/gambar akan disalin.

11. Aktifkan program Windows Explorer:



Keterangan:

1. Lakukan *drag-drop* file dari program Windows Explorer ke List1.
2. Lakukan *drag-drop* file dari program Windows Explorer ke Image2.

Catatan:

Drag-drop ke List1 bisa lebih dari 1 file (pada program Windows Explorer gunakan Shift+klik atau Ctrl+klik untuk memilih lebih dari 1 file).

Listing-15

```
Dim FormatOK As Boolean

'=====
'OLE Drag-Drop ke List1
'=====
Private Sub Text1_MouseDown(Button As Integer, Shift As Integer, _
    X As Single, Y As Single)
    'Jika Text1 tdk kosong dan tombol kiri mouse ditekan
    If Text1.Text <> "" And Button = vbLeftButton Then
        Text1.OLEDrag 'Mulai operasi OLE drag-drop
    End If
End Sub

Private Sub Text1_OLEStartDrag(Data As DataObject, AllowedEffects As Long)
    'Simpan data dgn format text
    Data.SetData Text1.Text, vbCFText
    'Atur efek OLE drag-drop yg diperbolehkan
    AllowedEffects = vbDropEffectCopy Or vbDropEffectMove
End Sub

Private Sub List1_OLEDragOver(Data As DataObject, Effect As Long, _
    Button As Integer, Shift As Integer, _
    X As Single, Y As Single, State As Integer)
    'Cek format datanya
    FormatOK = Data.GetFormat(vbCFText) Or _
        Data.GetFormat(vbCFFiles)

    If FormatOK Then
        'Cek apakah tombol Ctrl ditekan
        If Shift = vbCtrlMask Then
            Effect = vbDropEffectCopy 'Data disalin
        Else
            Effect = vbDropEffectMove 'Data dipindahkan
        End If
    Else
        Effect = vbDropEffectNone 'Data diabaikan
    End If
End Sub

Private Sub List1_OLEDragDrop(Data As DataObject, Effect As Long, _
    Button As Integer, Shift As Integer, _
    X As Single, Y As Single)

    Dim i As Integer

    'Cek format datanya
    FormatOK = Data.GetFormat(vbCFText) Or _
        Data.GetFormat(vbCFFiles)

    If FormatOK Then
        'Cek apakah tombol Ctrl ditekan
        If Shift = vbCtrlMask Then
            Effect = vbDropEffectCopy 'Data disalin
        Else
            Effect = vbDropEffectMove 'Data dipindahkan
        End If
    End If
End Sub
```



```
'Tambahkan data yg tersimpan ke dalam List1
If Data.GetFormat(vbCFText) Then
    List1.AddItem Data.GetData(vbCFText)      'Data text
End If
If Data.GetFormat(vbCFFiles) Then
    List1.Clear
    For i = 1 To Data.Files.Count
        List1.AddItem Data.Files(i)          'Data file(s)
    Next i
End If
Else
    Effect = vbDropEffectNone                'Data diabaikan
End If
End Sub

Private Sub Text1_OLECompleteDrag(Effect As Long)
    If Effect = vbDropEffectMove Then
        Text1.Text = ""                      'Kosongkan Text1 jika data dipindahkan
    End If
End Sub

'=====
'OLE Drag-Drop ke Image2
'=====
Private Sub Image1_MouseDown(Button As Integer, Shift As Integer, _
    X As Single, Y As Single)
    If Image1.Picture <> 0 And Button = vbLeftButton Then
        Image1.OLEDrag
    End If
End Sub

Private Sub Image1_OLEStartDrag(Data As DataObject, AllowedEffects As Long)
    Data.SetData Image1.Picture
    AllowedEffects = vbDropEffectCopy Or vbDropEffectMove
End Sub

Private Sub Image2_OLEDragOver(Data As DataObject, Effect As Long, _
    Button As Integer, Shift As Integer, _
    X As Single, Y As Single, State As Integer)
    FormatOK = Data.GetFormat(vbCFBitmap) Or _
        Data.GetFormat(vbCFMetafile) Or _
        Data.GetFormat(vbCFDIB) Or _
        Data.GetFormat(vbCFFiles)

    If FormatOK Then
        If Shift = vbCtrlMask Then
            Effect = vbDropEffectCopy
        Else
            Effect = vbDropEffectMove
        End If
    Else
        Effect = vbDropEffectNone
    End If
End Sub
```

```
Private Sub Image2_OLEDragDrop(Data As DataObject, Effect As Long, _  
    Button As Integer, Shift As Integer, _  
    X As Single, Y As Single)  
    FormatOK = Data.GetFormat(vbCFBitmap) Or _  
        Data.GetFormat(vbCFMetafile) Or _  
        Data.GetFormat(vbCFDIB) Or _  
        Data.GetFormat(vbCFFiles)  
  
    If FormatOK Then  
        If Shift = vbCtrlMask Then  
            Effect = vbDropEffectCopy  
        Else  
            Effect = vbDropEffectMove  
        End If  
        If Data.GetFormat(vbCFBitmap) Then  
            Image2.Picture = Data.GetData(vbCFBitmap)  
        End If  
        If Data.GetFormat(vbCFMetafile) Then  
            Image2.Picture = Data.GetData(vbCFMetafile)  
        End If  
        If Data.GetFormat(vbCFDIB) Then  
            Image2.Picture = Data.GetData(vbCFDIB)  
        End If  
        If Data.GetFormat(vbCFFiles) Then  
            Image2.Picture = LoadPicture(Data.Files(1))  
        End If  
    Else  
        Effect = vbDropEffectNone  
    End If  
End Sub  
  
Private Sub Image1_OLECompleteDrag(Effect As Long)  
    If Effect = vbDropEffectMove Then  
        Image1.Picture = Nothing  
    End If  
End Sub
```