

Pemrograman Delphi untuk Pemula: IDE dan Struktur Pemrograman

Teddy Marcus Zakaria

Teddy.Mz@maranatha.edu

<http://maranatha.edu>

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.

Abstrak:

Bagi pemula seringkali bingung ketika hendak mempelajari bahasa pemrograman, apapun bahasanya. Padahal tidak sesulit yang dibayangkan untuk mempelajari sebuah bahasa pemrograman. Pada tulisan ini akan dibahas cara memulai belajar bahasa pemrograman, khususnya Delphi. Namun cara seperti ini bisa diterapkan pula untuk belajar bahasa pemrograman yang lain, seperti keluarga xbase : dbase, Clipper, Foxbase/foxpro, keluarga visual : Delphi, Visual Basic, Visual C dan sebagainya.

Kali ini akan dibahas tiga struktur perintah dalam pemrograman komputer yaitu :

1. Perintah berupa Sequence (diproses oleh kompiler secara berurutan)
2. Perintah berupa Pencabangan / Pemilihan kasus (diproses sesuai kasus yang terpenuhi)
3. Perintah berupa Pengulangan (diproses secara berulang sampai kondisi tertentu)

Keywords: Sequence, Pencabangan, Pengulangan, Delphi

1. PENDAHULUAN

Sebelum kita memulai Delphi, kita harus menginstalasi program Delphinya terlebih dahulu. Kita bisa menggunakan Delphi 5 atau 6. Sebenarnya Delphi versi berapapun tidak masalah, karena produk Borland ini selalu menjaga kompatibilitas antar versinya. Untuk menginstalasi Delphi cukup mudah, tinggal ikuti petunjuk yang diberikan pada tampilan di layar. Sesudah Delphi terinstalasi pada komputer, kita bisa memulai mengenal dahulu apa itu Delphi? Kelompok bahasa pemrograman apa? Delphi adalah kompiler / penterjemah bahasa Delphi (awalnya dari Pascal) yang merupakan bahasa

tingkat tinggi sekelas dengan Basic, C. Bahasa Pemrograman di Delphi disebut bahasa prosedural artinya bahasa/sintaknya mengikuti urutan tertentu / prosedur. Ada jenis pemrograman non-prosedural seperti pemrograman untuk kecerdasan buatan seperti bahasa Prolog. Delphi termasuk Keluarga Visual sekelas Visual Basic, Visual C, artinya perintah-perintah untuk membuat objek dapat dilakukan secara visual. Pemrogram tinggal memilih objek apa yang ingin dimasukkan kedalam Form/Window, lalu tingkah laku objek tersebut saat menerima event/aksi tinggal dibuat programnya. Delphi merupakan bahasa berorientasi objek, artinya nama objek, properti dan metode/procedure dikemas menjadi satu kemasan (encapsulate).

Sebelum mempelajari ketiga struktur pemrograman ada baiknya kenali dahulu tampilan IDE, yang merupakan editor dan tools untuk membuat program Delphi. Pada IDE akan ditampilkan Form baru yang merupakan aplikasi/program Window yang akan dibuat.

Aplikasi / program berbasis windows sering disebut dengan jendela (window). Bagaimana membuat aplikasi berbasis windows (berbasis grafik dan bukan berbasis teks seperti pada DOS)? Caranya dengan membuat sebuah form. Pada pemrograman berbasis windows, kita akan diperhadapkan pada satu atau beberapa jendela yang nampak dihadapan kita. Jendela ini dalam Delphi disebut juga dengan form. Pada pemrograman berbasis windows, kita akan diperhadapkan pada satu atau beberapa jendela yang nampak dihadapan kita. Jendela ini dalam Delphi disebut juga dengan form.

Delphi adalah sebuah perangkat lunak (bahasa pemrograman) untuk membuat program / aplikasi komputer berbasis windows. Delphi merupakan bahasa pemrograman berbasis objek, artinya semua komponen yang ada merupakan objek-objek. Ciri sebuah objek adalah memiliki nama, properti dan method/procedure. Delphi disebut juga *visual programming* artinya komponen-komponen yang ada tidak hanya berupa teks (yang sebenarnya program kecil) tetapi muncul berupa gambar-gambar.

1.1. Membuat Sebuah Form

Saat anda pertama kali masuk ke Delphi, anda akan diperhadapkan pada sebuah form kosong yang akan dibuat secara otomatis. Form tersebut diberi nama **Form1**. Form ini merupakan tempat bekerja untuk membuat antarmuka pengguna.

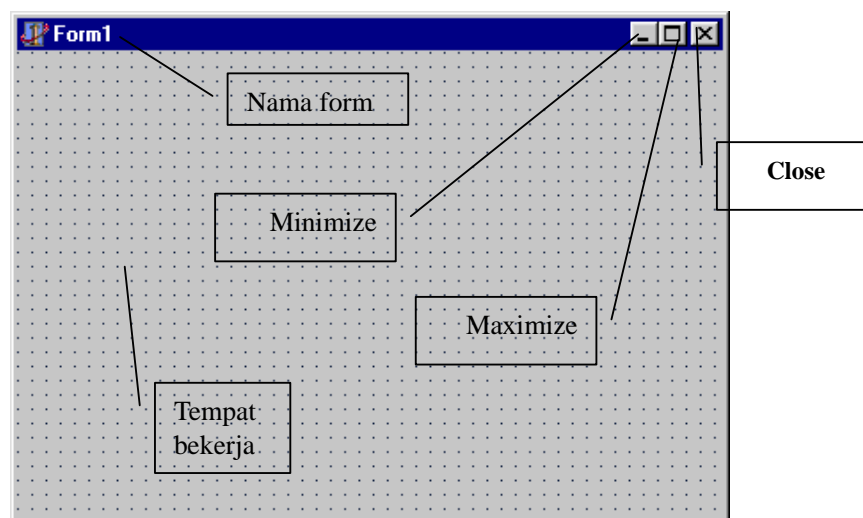


Figure 1 Form kosong yang dibuat saat anda masuk Delphi

1.2. Mengganti Nama Form dan Menambahkan Judul

Biasakan sebelum menjalankan program, sebaiknya ganti nama form dan beri judul sesuai program yang kita buat. Delphi akan secara otomatis memberi nama form1, form2, form3, dst. Nama form tersebut kurang mengandung arti dan akan menyulitkan bila form yang dibuat cukup banyak.

Saat membuka Delphi pertama kali, nampak sebuah jendela **Object Inspector**. Jika tidak muncul pilih menu *View / Object Inspector* atau tekan tombol F11.

Pada *Object Inspector* ada dua buah halaman (tab) yaitu Properties dan Events. **Properties** digunakan untuk mengganti properti (kepemilikan) sebuah objek/komponen. Sedangkan **Events** digunakan untuk membuat procedure yang diaktifkan (trigered) lewat sebuah event.

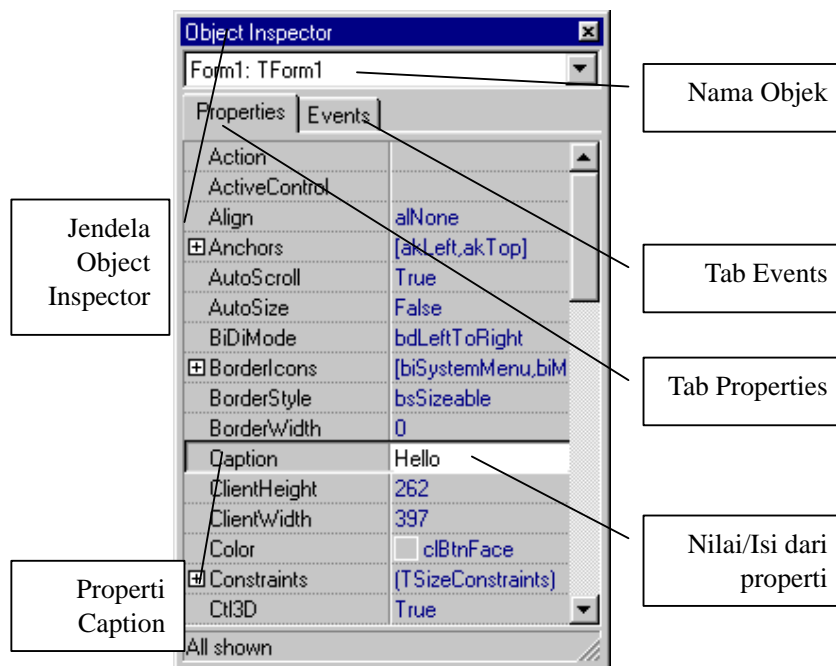


Figure 2 Jendela Object Inspector untuk mengganti properti caption

Semua properti diurutkan berdasarkan alpabetik, dan dapat juga diurutkan berdasarkan kategori. Gantilah judul form dengan Hello melalui properti **Caption**, sedangkan nama form dengan nama frmHello melalui properti **Name**. Caption digunakan untuk menyimpan keterangan yang dimunculkan pada form, sedangkan Name digunakan sebagai Nama dari objek tersebut.

Isi dari properti Name harus diawali alpabet dan tidak menggunakan spasi atau tanda baca

Anda sekarang sudah membuat aplikasi form kosong dengan tombol standar window : Minimize, Maximize, dan Close. Anda dapat mengubah ukuran form dengan menarik pada bingkai form menggunakan mouse (*drag*=klik tombol kiri mouse, tahan tombol tersebut lalu geser ke kiri/kanan atau atas/bawah). Anda dapat memindahkan form dengan meletakkan kursor pada form kemudian menggesernya (*drag*).

1.3. Menyimpan Form

Pada Delphi ada 3 buah file utama (*.dpr, *.pas dan *.dfm).

1) *.dpr adalah file proyek yang dibuat berisi program kecil untuk :

- mendefinisikan Unit yang ada dalam file proyek
- menginisialisasi data
- membangun form
- menjalankan aplikasi

```
uses
Forms,
Unit1 in 'Unit1.pas' {Form1};
begin
Application.Initialize;
Application.CreateForm(TForm1, Form1);
Application.Run;
end.
```

2) *.pas adalah unit-unit (*pascal code file*), bisa terdiri satu atau banyak file

3) *.dfm adalah file definisi Form (*special pseudo code file*), bisa terdiri satu atau banyak file

```
object Form1: TForm1
Left = 200
Top = 108
Width = 696
Height = 480
Caption = 'Form1'
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
PixelsPerInch = 96
TextHeight = 13
  object Button1: Tbutton
    Left = 176
    Top = 116
    Width = 75
    Height = 25
    Caption = 'Button1'
    TabOrder = 0
  end
end
```

Catatan:

Setiap Form (.dfm) harus memiliki sebuah Unit (.pas), tetapi anda dapat memiliki Unit tanpa sebuah Form (hanya kode saja). Jika ingin melihat kode tersebut anda dapat mengklik kanan mouse, lalu pilih **VIEW AS TEXT** atau tekan tombol **Alt- F12**. Sebaiknya anda tidak mengubah isi code tersebut, karena akan menyebabkan masalah serius. Tunggu saat anda sudah memahami maksud kode tersebut. Untuk kembali ke bentuk form, pilih **VIEW AS FORM** atau tekan tombol **Alt- F12** kembali.

Pilih submenu **Save Project** atau **Save Project As** pada menu File, dan Delphi akan menanyakan nama file source code untuk unit (*.pas) dan nama file proyeknya (*.dpr). Beri nama file form dengan HELLO.PAS dan project HELLO.DPR.

Sesudah disimpan, jalankan program dengan menekan tombol F9 atau pilih menu *Run / Run*.

1.4. Menempatkan Komponen pada Form

Karena Delphi merupakan bahasa pemrograman visual, maka komponen-komponen akan nampak pada layar. Anda tinggal menempatkan komponen yang diinginkan pada form. Ada empat cara menempatkan komponen pada form. Misal anda memilih komponen **Button** pada **Components Palette** bagian **Standard Page**. Anda dapat memilih salah satu langkah berikut:

- Klik pada komponen tersebut, pindahkan kursor ke form, sambil menekan tombol kiri mouse (drag komponen dan geser pada form) atau
- Pilih komponen (klik komponen yang diinginkan) pada **Components Palette** kemudian klik pada form dimana komponen itu akan diletakkan.
- Klik ganda pada komponen yang diinginkan, maka komponen tersebut akan ditambahkan pada form
- Anda dapat menggunakan *Copy* dan *Paste* bila ingin membuat komponen yang sama yang sudah ada pada form. Caranya Shift-Klik kiri pada komponen yang ada di form, lalu pilih menu *Copy* (Ctrl-C) kemudian pilih menu *Paste* (Ctrl-V).

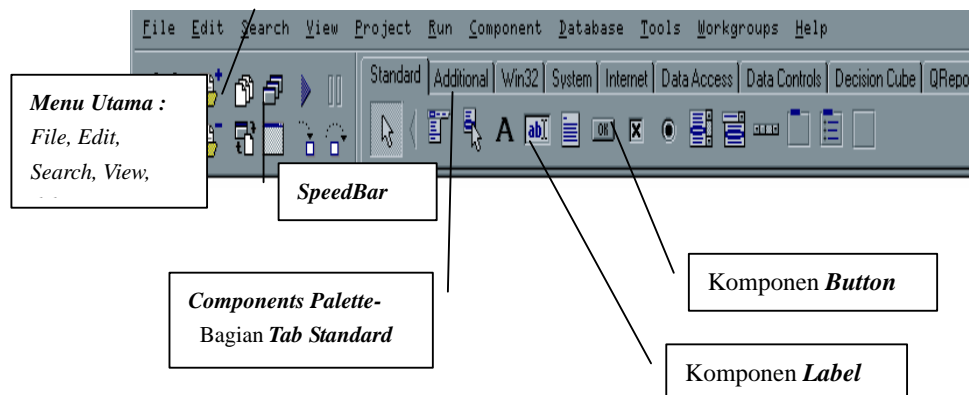


Figure 3 Kumpulan komponen

1.5. Mengatur Tataletak Komponen

Pada form ini hanya ada satu button, mungkin ada di bagian tengah form. Anda dapat mengatur letak komponen tersebut dengan menggesernya. Bila anda ingin merapihkan pilih menu **View | Alignment Palette**, maka muncul sebuah Toolbox Align dengan ikon perapihan (alignment icons).

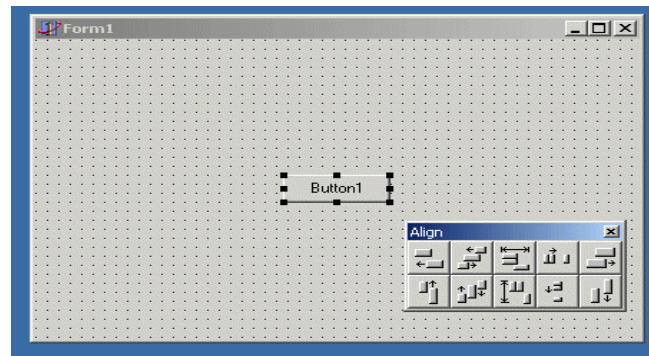


Figure 4 Form dan Toolbox Alignment

Dengan toolbox ini anda dapat merapikan beberapa komponen sekaligus, caranya buat fokus beberapa komponen, lalu klik icon pada toolbox yang diinginkan. Untuk mengetahui arti icon tersebut gerakan mouse pada tombol tersebut, lalu akan muncul penjelasan singkat kegunaan icon tersebut atau lihat Help (tekan F1). Anda bisa mempelajarinya sendiri. Cukup mudah!.

Langkah yang penting adalah mengubah nama dan keterangan komponen button tersebut. Ikuti bagian ini

1.6. Mengubah Nilai Properti

Ubah nilai properti **Caption** menjadi *Katakan Hello* dan nilai properti **Name** menjadi *btnHello*. Langkah ini mirip dengan mengubah nama dan keterangan sebuah form.

Setiap komponen sebaiknya diberinama yang memiliki arti dan diawali oleh jenis komponennya. Misal nama dari form Hello adalah "frmHello" atau nama dari button Hello adalah "btnHello". Tujuannya adalah mengelompokkan komponen-komponen sejenis, karena pada Object Inspector nama komponen diurutkan berdasarkan alpabet.

Properti **name** adalah properti internal dan digunakan untuk memberi nama pada sebuah komponen/objek. Nama ini adalah sebuah variabel yang mengacu pada komponen tersebut.

Beberapa aturan penamaan komponen atau variabel atau identifer sbb

- Diawali alpabet, berikutnya boleh angka, garis bawah.
- Tidak memakai spasi atau tanda-tanda baca atau operator
- Boleh huruf kapital atau kecil, tidak ada perbedaan
- Tidak menggunakan kata kunci (*reserve word*) yang digunakan Delphi
- Biasakan nama komponen diawali kelompok komponennya, misal btnHello, frmHello, rgrKelas.

Berikut contoh penamaan yang keliru menggunakan spasi



Figure 5 Pesan kesalahan saat mengubah properti Name.

Sesudah anda mengubah properti anda dapat melihat kode programnya (tekan **Alt-F12**) sbb :

```
object Form1: TForm1
:
Caption = 'Hello'
:
object btnHello: Tbutton
Caption = 'Katakan Hello'
End
end
```

1.7. Membuat Method/Procedure lewat Event

Saat anda menekan tombol pada sebuah form atau komponen, Windows memberitahukan bahwa aplikasi mengirim pesan yang dibangkitkan oleh event tertentu . Delphi akan menanggapi dengan menerima event atau panggilan tersebut. Hal ini yang dinamakan penanganan event (*event-handler method*). Delphi mendefinisikan sejumlah event pada setiap komponennya. Daftar event ini berbeda untuk setiap komponen. Event yang paling umum pada komponen Button adalah OnClick. Artinya jika komponen Button tersebut di Klik maka akan melakukan procedure apa.

Ada beberapa teknik yang dapat dilakukan untuk menangani event misal OnClick pada komponen button :

- Klik ganda pada button tersebut, maka sebuah method/procedure btnHelloClick
- Pilih button, kemudian pilih Object Inspector's combo box (called the Object Selector), pilih Tab Events, dan klik ganda pada area putih disebelah kanan event OnClick
- Pilih button, pilih Tab Events, dan masukkan nama method yang dikehendaki, misal btnHelloClick pada area putih di sebelah kanan event OnClick

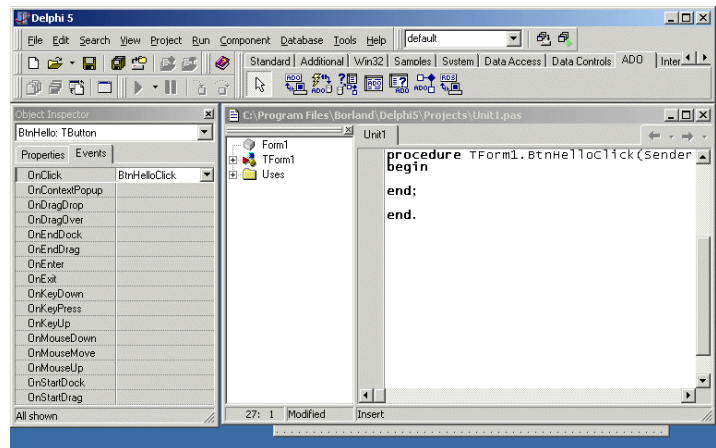


Figure 6 Tab Events pada Object Inspector, dan procedure yang dibangkitkan lewat event OnClick.

Bila anda ingin menghapus procedure atau penanganan event tersebut, anda dapat menghapus pada editor Unit tersebut. Hapus blok procedure tersebut dan hapus pada bagian definisi procedure yang ada di atasnya.

Sekarang anda mengisi procedure tersebut dengan perintah untuk menampilkan pesan sbb :

```
procedure TForm1.BtnHelloClick(Sender: TObject);  
begin  
    MessageDlg ('Hello, guys', mtInformation, [mbOK], 0);  
end;
```

Perintah ini sangat sederhana, yaitu untuk menampilkan pesan. Fungsi ini mempunyai empat parameter. Untuk rincinya anda dapat melihat Bantuan (F1).

- Parameter pertama : kalimat yang akan dimunculkan (pesannya)
- Parameter kedua : tipe message box seperti mtWarning, mtError, mtInformation, atau mtConfirmation. Coba lihat perbedaannya.
- Parameter ketiga : kumpulan tombol yang akan digunakan seperti mbYes, mbNo, mbOK, mbCancel, atau mbHelp.
- Parameter keempat : untuk help context atau nomor halaman pada Help, beri angka nol jika anda tidak mempunyai file help.



Gambar 1.1 Message box

1.8. Kompilasi dan Jalankan Program

Tekan tombol Run atau pilih menu *Run / Run*, Delphi does the following:

1. Kompilasi Pascal source code file yang mendefinisikan form-form yang ada (.pas, .dfm)
2. Kompilasi project file (.dpr)
3. Buat executable (.EXE) file
4. Jalankan executable file, biasanya pada mode pencarian kesalahan (debug mode).

2. PERINTAH SEQUENCE / DIPROSES SECARA BERURUTAN

Perintah-perintah ini akan diproses oleh kompiler secara berurutan. Contoh kita memiliki 3 baris perintah. Kompiler akan memroses mulai dari baris-1, baris-2 kemudian baris-3.

Contoh program untuk menampilkan pesan sebanyak 3 kali seperti berikut :

```
MessageDlg ('Hello, guys', mtInformation, [mbOK], 0);  
MessageDlg ('Hello, lady', mtInformation, [mbOK], 0);  
MessageDlg ('Hello, everybody', mtInformation, [mbOK], 0);
```

Mudah ...khan??

Berikut latihan ke-1 untuk tiga buah aplikasi yaitu membuat tombol Hello, saat diklik tombolnya, tombol akan beraksi dengan memunculkan pesan. Ikuti langkahnya sbb :

1. Buat form frmHello1 seperti yang sudah dijelaskan.

- Komponen yang dibutuhkan form dengan nama **frmHello1** dan button dengan nama **btnHello1**. Ubah properti *name*-nya
- Ubah properti *caption* masing-masing komponen menjadi **Membuat program Hello1** dan **Katakan Hello**
- Buat method dari komponen btnHello1 lewat event *OnClick*, seperti berikut

```
procedure TfrmHello1.btnHello1Click(Sender: TObject);  
begin  
    MessageDlg ('Hello, guys', mtInformation, [mbOK], 0);  
end;
```

- Jalankan program lihat hasilnya
- Coba ganti parameter ke-1, ke-2 dan ke-3
- Simpan dengan nama unit *uHello1.pas* dan project *pHello1.dpr* pada direktori Latih1

2. Buat aplikasi baru (File-New-Application) dengan form baru frmHello2 mirip form frmHello1 dengan tambahan sebagai berikut. Apa yang terjadi? (properti caption dari komponen tombol diganti saat program berjalan)

- Komponen yang dibutuhkan form dengan nama **frmHello2** dan button dengan nama **btnHello2**. Ubah properti *name*-nya
-

```
procedure TfrmHello2.btnHello2Click (Sender: TObject);  
begin  
    MessageDlg ('Hello, guys', mtInformation, [mbOK], 0);
```

```
BtnHello2.Caption := 'Say Hello Again';  
end;
```

- Buat method untuk mengubah ukuran formHello2 lewat event *OnResize*, perhatikan apa yang terjadi bila ukuran form diubah saat program berjalan

```
procedure TfrmHello2.FormResize(Sender: TObject);  
begin  
  BtnHello2.Top := frmHello2.ClientHeight div 2 -  
    BtnHello2.Height div 2;  
  BtnHello2.Left := frmHello2.ClientWidth div 2 -  
    BtnHello2.Width div 2;  
end;
```

- Simpan dengan nama unit *uHello2.pas* dan project *pHello2.dpr* pada direktori Latih1

3. Buat aplikasi baru (File-New-Application) dengan form baru frmHello3 seperti yang sudah dijelaskan.

Objek	Properties	Value of properties	Method lewat event
Form1	Name	FrmHello3	
Button1	Name	BtnHello3	OnClick (lihat dibawah)
	Caption	&Show Me	
Label1	Name	LblHello3	
	Font - Size	24	

- Buat method dari komponen btnHello3 lewat event *OnClick*, seperti berikut

```
procedure TForm1.btnHello3Click(Sender: TObject);  
begin  
  if btnHello3.caption = '&Show Me' then  
  begin  
    lblHello3.caption := 'Selamat Datang di Delphi 5.0';  
    btnHello3.caption := '&Hide Me';  
  end  
  else  
  begin  
    lblHello3.caption := '';  
    btnHello3.caption := '&Show Me';  
  end;  
end;
```

- Jalankan program lihat hasilnya
- Simpan dengan nama unit *uHello3.pas* dan project *pHello3.dpr* pada direktori Latih1

2.1. Menu dan Perintah pada Delphi

Ada empat cara untuk memberi perintah pada lingkungan Delphi (Delphi environment):

- Gunakan menu
- Gunakan Short Cut (misal F9, F12 dsb)
- Gunakan SpeedBar (atau toolbar).
- Gunakan SpeedMenu (lokal menu yang diaktifkan dengan tombol mouse kanan).

Berikut menu utama yang ada pada Delphi (untuk mempelajarinya gunakan Help Delphi):

Menu File. Menu ini berhubungan dengan file seperti membuat, menyimpan dan mengakhiri sebuah pekerjaan.

Menu Edit .Menu ini berhubungan dengan penyuntingan apa yang dikerjakan seperti Undo , Redo, Cut, Copy, Paste atau dapat dengan tombol Ctrl+Z, Ctrl+X, Ctrl+C, Ctrl+V.

Menu Search. Menu ini berhubungan dengan pencarian dan penggantian data.



Figure 7 Jendela Find Text

Menu View. Menu ini berhubungan dengan penampilan atau apa yang akan ditampilkan.

Menu Project. Menu ini berhubungan dengan proyek yang sedang dibuat, misal unit yang akan ditambahkan ke proyek ini, unit apa yang akan dihapus, dsb.

Menu Run. Menu ini berhubungan dengan menjalankan program, mencari kesalahan (debug), dsb.

Menu Component. Menu ini berhubungan dengan komponen, misal menambah komponen baru, menghapus komponen yang ada.

Menu Database. Menu ini berhubungan dengan Database, Database Form Wizard dan Database Explorer.

Menu Tools. Menu ini berhubungan dengan pengaturan/konfigurasi, tool-tool pembantu Delphi.

Menu Help. Menu ini berhubungan dengan informasi mengenai Delphi, Help / bantuan

2.2. Component, Property, Method, Event

Kode yang akan dilihat, serupa dengan struktur Bahasa Pascal. Delphi adalah bahasa pemrograman berbasis objek, artinya pendekatan pembuatan program melalui objek-objek yang ada. Misalnya objek form, text dsb. Setiap objek akan memiliki properti (atribut) dan method yang diaktifkan / dipicu oleh event. Mari kita lihat penjelasan berikut.

Apakah Objek (COMPONENT) itu?

Ingat sebuah komponen adalah sebuah objek pada Palette, :

- sebuah Objek, adalah sebuah komponen dalam Component Palette,
- atau sesuatu yang dibuat melalui kode-kode / bahasa pemrograman

Jadi sebuah objek adalah secara umum kelas dari kumpulan sesuatu. Komponen pasti objek namun tidak selalu merupakan komponen, misal TStringList adalah sebuah objek (kumpulan karakter), dan bukan sebuah komponen. Penjelasan mengenai komponen di Lampiran A.

Apakah sebuah PROPERTY itu?

Sebuah Property tidak lain adalah sebuah nama/variabel milik sebuah objek/komponen misal Caption, Text yang dapat diubah nilai baik melalui object Inspector atau melalui program.

Beberapa istilah/ nama berikut yang mirip, dan sering digunakan:

- **Procedure** adalah kumpulan perintah yang melakukan suatu proses tertentu
- **Function** adalah sama dengan procedure, tetapi proses tersebut dapat mengembalikan suatu hasil / nilai misal hasilnya = 1
- **Method** adalah procedure atau function yang tergabung pada sebuah komponen
- **Subroutine** adalah istilah umum dari semuanya (procedure/function/method) misal pada bahasa Basic.

Apakah sebuah METHOD itu?

Sebuah method adalah sebuah function/fungsi yang tergabung dalam sebuah objek. Contoh ListBox (dapat berarti sebuah array of strings) yang memiliki Method (Clear) yang membuat Listbox tersebut menjadi kosong. CLEAR adalah sebuah Method pada ListBox tersebut.

```
Begin
  ListBox1.Clear; // Mengosongkan isi ListBox
  ListBox1.Items.LoadFromFile('c:\Data1.txt');
  //properti Items (bertipe string) memiliki method untuk LoadFromFile
end;
```

Apakah sebuah EVENT itu?

Sebuah Event adalah sebuah aksi pengguna (User Action) misal Mouse Click, KeyPressed. Setiap Events diawali dengan kata 'On'.

Contoh :

Nama event	Nama method
OnClick ..	Button1Click(Sender : TObject)
OnKeyDown ..	Button1KeyDown(Sender : TObject)
OnMouseMove ..	Button1MouseMove(Sender : TObject)

2.3. Cara Delphi Bekerja

Saat anda menambahkan Components pada Form1 dan merubah nilai properti, Delphi akan membuat (pseudo) code (dalam Unit1.dfm) untuk mendefinikan apa yang anda lakukan. Secara normal, ANDA TIDAK DIHARAPKAN MENGUBAHNYA Unit1.dfm File; Anda hanya bekerja pada Form1 secara Visual. Ini yang dinamakan bahasa pemrograman Visual (Visual Programming)

Apakah Delphi itu?

- Delphi (IDE) adalah sebuah Visual Interface antara anda (the User) dan Komputer anda (yang berjalan diatas Windows).
- Delphi menterjemahkan Visual Components (Buttons, Panels,..) yang ada pada Form kedalam sebuah kode-kode komputer (pseudo in Unit1.dfm) yang mendefinisikan bagaimana dibentuknya form dan komponennya termasuk juga propertinya.

Saat anda mengkompilasi program, apa yang terjadi?

- Delphi akan memanggil file .dpr file (file proyek anda)
- Delphi meminta program yang ada dalam proyek tersebut dan file dpr memberikan sbb :

```
uses
  Forms,
  Unit1 in 'Unit1.pas' {Form1};
```

- Delphi meminta, Apa yang dilakukan pertama kali ? .dpr file memberikan sbb :

```
begin
  Application.Initialize; itializes stuff
  Application.CreateForm(Tform1, Form1);
  Application.Run;
end.
```

2.4. Forms, Dialog Boxes, Messages , ...

Menampilkan Form atau Window

Anda dapat bekerja dengan beberapa form pada sebuah *project*. Saat anda berada di Form1 untuk menampilkan form yang lain misal Form2 sbb :

Form2.Show;

Membuka (Shows) Form2 (tetapi user diijinkan untuk dapat mengklik Form1)

Form2.ShowModal;

Membuka (Shows) Form2 (tetapi user TIDAK DIIJINKAN mengklik Form1)

Menampilkan Pesan (Message)

ShowMessage('Ini kotak pesan);

Tampilan sederhana sebuah baris/teks; User dapat menekan tombol OK untuk keluar dari kotak pesan (message window).

MessageDlg('Msg',mtConfirmation,[mbYes],0);

Mirip ShowMessage tapi dapat lebih dari satu tombol pilihan.

if **MessageDlg('Please say YES or NO',mtConfirmation,[mbYes,mbNo],0)=mrYES** then

```
begin  
  Label1.Text:="Tekan tombol YES";  
end;
```

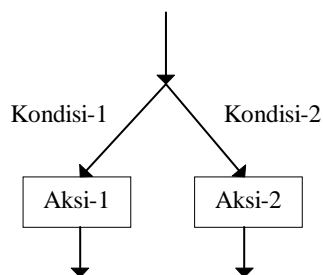
```
TMsgDlgType = mtWarning, mtError, mtInformation, mtConfirmation, mtCustom  
  
TMsgDlgBtn  
=  mbYes,    mbNo,    mbOK,    mbCancel,    mbAbort,    mbRetry,    mbIgnore,    mbAll,    mbHelp  
  
Return values  
=  mrNone,    mrYes,    mrNo,    mrOk,    mrCancel,    mrAbort,    mrRetry,    mrIgnore,    m  
rAll
```

Meminta Masukan (Input Box)

```
InputBox(...);  
  var  
    InputString: string;  
begin  
  InputString:= InputBox('Masukkan Nama', 'Please Enter your Name','');  
end;
```

3. PERINTAH PENCABANGAN / STRUKTUR PEMILIHAN

Pada beberapa kasus terkadang kita menginginkan komputer melakukan suatu aksi tertentu bila suatu kondisi terpenuhi. Keberadaan perintah bersyarat pada suatu program memberikan pencabangan proses seperti ditunjukkan oleh **Error! Not a valid link..** Bahasa Pascal menyediakan dua cara penyajian perintah bersyarat, yaitu If...Then...Else dan Case...of.



Gambar 3.1 Pencabangan pada program

Pada prinsipnya pemilihan aksi dapat dikelompokkan menjadi 3 bagian :

```
Pemilihan 1 pilihan/kasus. Gunakan perintah IF - THEN  
Pemilihan 2 pilihan/kasus. Gunakan perintah IF - THEN - ELSE  
Pemilihan N pilihan/kasus. Gunakan Case - of (Catatan sebenarnya dapat menggunakan if-then - else  
bersarang)
```

3.1. IF ... THEN : Pemilihan 1 kasus

Perintah bersyarat If – then digunakan hanya melakukan 1 aksi bila kondisi dipenuhi. Bentuk sintaks dari perintah bersyarat ini adalah:

```
If <kondisi_pencabangan> then  
Begin  
    ... {aksi-1}  
end;
```

3.2. IF ... THEN ... ELSE : Pemilihan 2 kasus

Perintah bersyarat If umumnya digunakan untuk melakukan pencabangan sederhana (antara 2 atau 3 cabang). Atau untuk pencabangan yang banyak, dimana kondisi yang menjadi prasyaratnya melibatkan lebih dari satu parameter. Bentuk sintaks dari perintah bersyarat ini adalah:

```
If <kondisi_pencabangan> then  
Begin  
    ... {aksi-1}  
end  
else  
Begin  
    ... {aksi-2}  
end;
```

3.3. IF ... THEN ... ELSE : Pemilihan N kasus

```
If <kondisi_pencabangan1> then  
Begin  
    ... {aksi-1}  
end  
else  
    If <kondisi_pencabangan2> then  
        Begin  
            ... {aksi-2}  
        end  
    else  
        Begin  
            ... {aksi-3}  
        End;
```

3.4. CASE ... OF ... : Pemilihan N kasus

Perintah bersyarat Case umumnya digunakan untuk kondisi dengan banyak percabangan. Syarat percabangan pada bentuk ini hanya boleh melibatkan satu buah parameter dengan tipe data bukan Real. Pemeriksaan kondisi di sini lebih tepat disebutkan dalam hubungan relasi samadengan (=). Dengan demikian bila parameter bernilai tertentu maka dilakukan suatu aksi terkait, bila bernilai lain maka dilakukan aksi yang lain juga, demikian seterusnya.

```

Case <Parameter> Of
    <nilai_1> : <aksi_1> ;
    <nilai_2> : <aksi_2> ;
    ...
    <nilai_n> : <aksi_n> ;
    Else <aksi_n+1> ;
End;
    
```

Latihan ke-2

Buat aplikasi baru dengan nama form frmWelcome, kemudian ubah nama komponen/objek sbb:

Objek	Properti	Nilai properti	Method lewat event
Form1	Name	FrmWelcome	OnActivate
	Caption	Welcome	
Button1	Name	BtnQuit	OnClick, isi dengan <i>application.terminate;</i>
	Caption	&Quit	
Label1	Name	LblIWP	
	Caption	Internasional Welcome Program	
	Font	Times New Roman, Bold, Size = 18	
Label2	Name	LblChoose	
	Caption	Choose a Country	
Label3	Name	LblCountry	
	Caption	(kosong)	
	Font	Times New Roman, Bold, Size = 12, Color = clTeal	
Label4	Name	LblComment	
	Caption	(kosong)	
	Font	Times New Roman, Bold-Italic, Size = 24, Color = clTeal	
ListBox1	Name	LbxCountry	OnClick

- Buat method dari komponen frmWelcome lewat event *OnActivate*, seperti berikut

```
procedure TForm1.FormActivate(Sender: TObject);  
begin  
    lbxCountry.Hint := 'Please choose...';  
    lbxCountry.Items[0]:= 'England';  
    lbxCountry.Items[1]:= 'Germany';  
    lbxCountry.Items[2]:= 'Spain';  
    lbxCountry.Items[3]:= 'Italy';  
end;
```

- Buat method dari komponen lbxCountry lewat event *OnClick*, seperti berikut

```
procedure TForm1.lbxCountryClick(Sender: TObject);  
begin  
    lblCountry.caption := lbxCountry.Items[lbxCountry.ItemIndex];  
  
    case lbxCountry.ItemIndex of  
        0 : lblComment.caption := 'Hello, Programmer' ;  
        1 : lblComment.caption := 'Hallo, Programmierer' ;  
        2 : lblComment.caption := 'Hola, Programador' ;  
        3 : lblComment.caption := 'Ciao, Programmtori' ;  
    end;  
end;
```

- Simpan dengan nama unit *uWelcome1.pas* dan project *pWelcome1.dpr* pada direktori Latih3

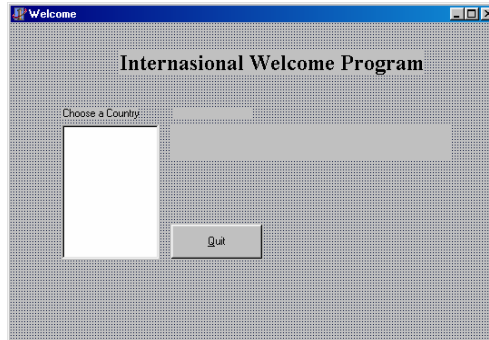


Figure 8 Desain form :Pemilihan kasus



Figure 9 Hasil : Pemilihan kasus

4. STRUKTUR PENGULANGAN

Dalam menyelesaikan masalah, terkadang kita harus melaku suatu proses yang sama lebih dari satu kali. Untuk itu perlu dibuat suatu algoritma pengulangan. Pascal memberikan tiga alternatif pengulangan, yaitu dengan For, While, atau Repeat. Masing-masing memiliki karakteristik, yang akan dipelajari pada modul ini. Ada dua hal yang penting dalam melakukan merancang perintah pengulangan, yaitu:

- Inisialisasi awal.
- Nilai akhir pengulangan atau kondisi berhenti.

4.1. FOR – TO - DO

Pada pengulangan dengan For, inisialisasi awal dan kondisi akhir ditentukan dengan menggunakan suatu variable kendali yang nilainya dibatasi dalam suatu range tertentu. Sintaks untuk perintah ini adalah:

```
For <variable_kendali> := <nilai_awal> to <nilai_akhir> do  
Begin  
    ... {aksi}  
End ;
```

atau

```
For <variable_kendali> := <nilai_awal> downto <nilai_akhir> do  
Begin  
    ... {aksi}  
End ;
```

Perbedaan antara to dan downto adalah pada kondisi nilai awal dan akhir. Pada to: nilai awal lebih kecil dari nilai akhir, sedangkan pada downto nilai awal lebih besar dari nilai akhir. Perlu diingat, bahwa variable kendali harus dideklarasikan dengan tipe data integer.

Contoh program pengulangan menggunakan For-to-do

```
procedure TfrmPengulangan.btnPengulanganClick(Sender: TObject);  
var I : integer;  
begin  
    lbxHasil.Items.Clear;  
    For I := 1 to 5 do  
        Begin  
            lbxHasil.Items.Add(inttoStr(i));  
        End;  
end;
```

Hasilnya adalah : 1, 2, 3, 4, 5

Contoh program yang sama jika tanpa perintah tanpa pengulangan

```
procedure TfrmPengulangan.btnPengulanganClick(Sender: TObject);  
begin  
  lbxHasil.Items.Clear;  
  
  lbxHasil.Items.Add(inttoStr(1));  
  lbxHasil.Items.Add(inttoStr(2));  
  lbxHasil.Items.Add(inttoStr(3));  
  lbxHasil.Items.Add(inttoStr(4));  
  lbxHasil.Items.Add(inttoStr(5));  
  
End;
```

Hasilnya adalah : 1, 2, 3, 4, 5

4.2. WHILE - DO

Pada metoda pengulangan ini aksi hanya akan diproses bila kondisi pengulangan terpenuhi, bentuk sintaks dari pengulangan ini adalah:

```
While <kondisi_pengulangan> do  
  
Begin  
  ... {aksi}  
  
End ;
```

Selama *kondisi_pengulangan* bernilai true maka aksi akan dilakukan, dan baru akan berhenti setelah kondisi pengulangan bernilai false. Karena kondisi pengulangan diperiksa pada bagian awal, maka ada kemungkinan aksi tidak pernah dilakukan, yaitu bila kondisi pengulangan tidak pernah bernilai true.

Contoh program pengulangan menggunakan While-do

```
procedure TfrmPengulangan.btnPengulanganClick(Sender: TObject);  
var I : integer;  
begin  
  lbxHasil.Items.Clear;  
  
  I := 1;  
  While I<=5 DO  
  Begin  
    lbxHasil.Items.Add(inttoStr(i));  
    I := I + 1;  
  End;  
  
End;
```

4.3. REPEAT - UNTIL

Metoda pengulangan ini juga melakukan pengulangan berdasarkan pemeriksaan kondisi pengulangan. Hanya saja natur dari pengulangan ini adalah sistem seakan-akan memaksa untuk melakukan pengulangan, sampai di ketahui adanya kondisi berhenti. Bentuk sintaks dari pengulangan ini adalah:

```
Repeat  
  
    ...    {aksi}  
  
Until <kondisi_berhenti>
```

Berlawanan dengan While, yang akan memproses aksi hanya bila kondisi_pengulangan bernilai true, pada pengulangan Repeat, sistem akan memproses aksi selama kondisi_berhenti bernilai false. Dengan demikian aksi pasti akan selalu diproses (minimal satu kali). Pada tipe ini, pengulangan dapat terjadi terus-menerus (tidak pernah berhenti), yaitu bila kondisi berhenti tidak pernah bernilai true.

Contoh program pengulangan menggunakan Repeat- Until

```
procedure TfrmPengulangan.btnPengulanganClick(Sender: TObject);  
var I : integer;  
begin  
    lbxHasil.Items.Clear;  
  
    I := 1;  
    Repeat  
        lbxHasil.Items.Add(inttoStr(i));  
        I := I + 1;  
    Until I > 5;  
End;
```

Latihan ke-3

Buat aplikasi bari dengan nama form frmPengulangan, kemudian ubah nama komponen/objek sbb:

Objek	Properti	Nilai properti	Method lewat event
Form1	Name	FrmPengulangan	
	Caption	Pengulangan	
Button1	Name	BtnPengulangan	OnClick
	Caption	&Pengulangan	
Label1	Name	LblComment	
	Caption	(kosong)	
	Font	Times New Roman, Bold-Italic, Size = 12	
ListBox1	Name	LbxHasil	OnClick
	Font	Times New Roman, Bold-Italic, Size = 20, Color = clTeal	

- Buat method dari komponen btnPengulangan lewat event *OnClick*, seperti berikut

```
procedure TfrmPengulangan.btnPengulanganClick(Sender: TObject);  
var I : integer;  
begin  
  lbxHasil.Items.Clear;  
  For I := 0 to 20 do  
  Begin  
    lbxHasil.Items.Add(inttoStr(i));  
  End;  
end;
```

- Buat method dari komponen lbxHasil lewat event *OnClick*, seperti berikut

```
procedure TfrmPengulangan.lbxHasilClick(Sender: TObject);  
begin  
  lblComment.caption :=  
    'Dicetak saat I bernilai ' + IntToStr(lbxHasil.ItemIndex);  
end;
```

- Simpan dengan nama unit *uPengulangan1.pas* dan project *pPengulangan1.dpr* pada direktori Latih4

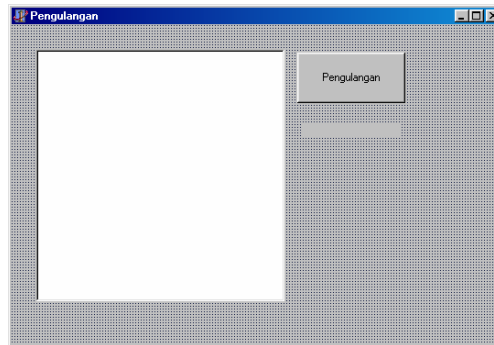


Figure 10 Desain Form : Pengulangan

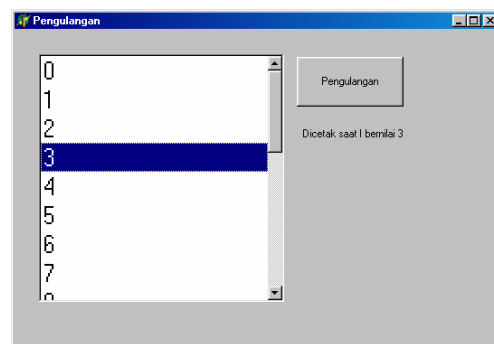


Figure 11 Hasil : Pengulangan

Cukup dulu yah belajarnya, nanti kita lanjutkan dengan seri lainnya.

BIOGRAFI PENULIS

Teddy Marcus Zakaria. Lahir di Bandung, 18 Maret 1969. Menamatkan SMA di SMA Trinitas Bandung pada tahun 1987. Menyelesaikan program S1 Teknik Informatika di ST.Inten Bandung tahun 1992 dan S2 pada jurusan Teknik Informatika ITB Bandung tahun 1998. Saat ini bekerja sebagai karyawan di Departemen Teknologi Informasi bidang Pengembangan Perangkat Lunak di Universitas Kristen Maranatha. Waktu luang digunakan mengajar di Universitas Kristen Maranatha dan STMIK-LIKMI Bandung. Kalau mengajarnya diberi sedikit ya cari proyek untuk tambah penghasilan, buat program aplikasi bisnis Inventori, Penjualan-Piutang, Pembelian-Hutang, Jurnal-Neraca ya seperti itulah. Jika ada tawaran buat program yang aneh, ya dicoba saja kenapa tidak, seperti Absensi Karyawan pakai foto, Sistem Informasi Perparkiran, SMS Gateway, maklum sambil belajar ya cari penghasilan. Cita-citaku yaitu memajukan bangsa ini, jadi selain mengajar, pembimbing juga coba-coba jadi penulis kecil-kecilan di beberapa penerbit. Soalnya aku lihat kita sangat tertinggal dengan negara-negara tetangga dalam bidang pendidikan dan teknologi, ya kalau bukan kita yang membangun bangsa ini siapa lagi yah....?

Informasi lebih lanjut tentang penulis ini bisa didapat melalui:
Email: Teddy.Mz@maranatha.edu

Untuk mempelajari pemrograman Delphi lebih lanjut anda dapat melihat/berlangganan artikel di beberapa situs berikut. Cari yang gratis supaya tidak menusuk mata maksudnya mata pencarian. Seperti :

- About Delphi Programming From Zarko Gajic, your Guide to Delphi Programming <http://delphi.about.com>,
- 3k Delphi Tips by Jounlai Cho <http://3kdt.delpphi.co.kr>,
- Weekly newsletter of SwissDelphiCenter developers knowledge base www.swissdelphicenter.com,
- Borland Community <http://community.borland.com>
- DELPHI SUPER PAGE <http://sunsite.icm.edu.pl/delphi/>
- TORRY"S DELPHI PAGES <http://www.torry.ru/index.htm>