

Perkuliahan Pemrograman II (Teori / Praktikum)

BORLAND DELPHI

Materi Hari Ini

- ▶ Pengenalan Borland Delphi
- ▶ Dasar–Dasar OOP dengan Delphi
- ▶ Pengenalan IDE Delphi
- ▶ Form dan Komponen

PENGENALAN DELPHI

Pengenalan Delphi

- ▶ Borland Delphi
- ▶ Kelebihan Menggunakan Borland Delphi
- ▶ Aplikasi yang dibangun dengan Delphi

Borland Delphi

- ▶ Borland Delphi adalah sebuah alat pengembangan aplikasi–aplikasi untuk sistem operasi Microsoft Windows.
- ▶ Delphi merupakan bahasa pemrograman pertama yang memecahkan batasan antara bahasa tingkat tinggi, pengembangan aplikasi dengan cepat (Rapid Application Development/RAD).

Kelebihan Menggunakan Borland Delphi

- ▶ Delphi mendukung Pemrograman Berorientasi Objek (Object Oriented Programming/OOP)
- ▶ Pengembangan aplikasi secara cepat (Rapid Application Development/RAD)
- ▶ Menggunakan bahasa tingkat tinggi
- ▶ Hasil dari proses kompilasi berupa sebuah file yang dapat dieksekusi (executable file) sehingga mempermudah dalam pendistribusian program dan mengurangi banyaknya file pendukung DLL
- ▶ Delphi menyediakan banyak sekali komponen yang dapat digunakan. Selain itu banyak juga komponen yang bersumber dari pihak ketiga yang biasanya disertai dengan dokumentasi, source code dan lain-lain. Komponen dari pihak ketiga bisa yang komersil atau free.
- ▶ Mendukung banyak database server (MySQL, SQL Server, Interbase, Oracle dll) sehingga dapat mempermudah dalam membuat aplikasi database.

Aplikasi yang dibangun dengan Delphi

- ▶ Produk Borland : Borland Delphi, Borland C++ Builder, Borland JBuilder versi 1 dan 2
- ▶ Perangkat Lunak Akunting Panggilan : PhoneControl
- ▶ Game : Astral Masters, Astral Tournament, Smugglers series, [Soldat](#), [Quake conversion from the C source](#), Space Rangers, [Space Rangers 2: Dominators](#)
- ▶ Management Database : Tool MySQL (Administrator, Query Browser, Migration Toolkit)
- ▶ Internet Messaging : [Skype](#) (VoIP and IM), [The Bat!](#) (e-mail client), PopTray (e-mail check tool), [FeedDemon](#) (RSS/Atom feed viewer), [XanaNews](#) (newsgroup reader), [Xnews](#) (newsgroup reader)Customer relationship management: Sage SalesLogix
- ▶ Produksi Musik : [FL Studio](#)
- ▶ Pengembangan Software : : [Dev-C++](#) (IDE), [DUnit](#), [Help & Manual](#) (help system authoring), [Inno Setup](#) (installer engine), [ConTEXT](#) (Programmers editor)
- ▶ Pengembangan Web : [Macromedia HomeSite](#) (HTML editor), [TopStyle Pro](#) (CSS editor), [Macromedia Captivate](#) ([screencast](#))
- ▶ Browser Web : [Avant Browser](#), [Netcaptor](#)
- ▶ Utility : [Spybot - Search & Destroy](#), [Ad-Aware](#) (anti-spyware), [Total Commander](#) (file manager), [Copernic Desktop Search](#), [PowerArchiver](#), ASuite

DASAR-DASAR OOP DENGAN DELPHI

Dasar-Dasar OOP Dengan Delphi

- ▶ Overview Object Oriented Programming
- ▶ Definisi Class
- ▶ Definisi Object
- ▶ Ilustrasi Data Abstraction
- ▶ Ilustrasi Encapsulation
- ▶ Ilustrasi Inheritance
- ▶ Ilustrasi Polimorphism
- ▶ Menghidupkan Object
- ▶ Mematikan Object

Overview Object Oriented Programming

- ▶ Object Pascal merupakan bahasa dasar yang digunakan oleh Delphi. Object Pascal merupakan bahasa pemrograman yang berorientasi objek.
- ▶ 4 prinsip dasar dari pemrograman berorientasi objek yaitu :
 - Abstraksi Data (Data Abstraction)
 - Enkapsulasi (Encapsulation)
 - Pewarisan (Inheritance)
 - Polimorpisme (Polymorphism)

Definisi Class (1/2)

- ▶ Sebuah Class mendefinisikan karakteristik abstrak dari sebuah benda (obyek), termasuk karakteristik benda (atribut atau ciri benda) dan perilaku benda tersebut (sesuatu yang bisa dilakukan oleh benda tersebut atau dikenal dengan method atau operasi).
- ▶ Banyak juga yang mendefinisikan class sebagai cetak biru yang menjelaskan tentang sesuatu.
- ▶ Contoh adalah obyek manusia. Manusia mempunyai atribut tinggi, berat badan, nama, warna rambut serta memiliki method atau operasi seperti manusia bisa melakukan makan, minum, berbicara dan lain-lain. Properti atau atribut dan method yang mendefinisikan suatu Class disebut dengan member (anggota).

Definisi Class (2/2)

Contoh Pendefinisian Class dengan Delphi

```
TManusia=class    {atau TManusia=class(TObject) }
    Nama:String;
    Tinggi:Integer;
    Berat:Double;
    procedure UcapNama;
end;
{Implementasi method UcapNama}
procedure TManusia.UcapNama;
begin
    writeln('Saya adalah '+nama);
end;
```

Definisi Object (1/2)

- ▶ Object adalah sebuah instansiasi (instance) khusus dari sebuah class.
- ▶ Contoh : orang yang bernama Irma merupakan sebuah instansiasi dari sebuah class TManusia.

Definisi Object(2/2)

Contoh Pendefinisian Object dengan Delphi

```
var  
    Orang:TManusia;  
    Irma:Tmanusia;
```

Ilustrasi Data Abstraction (1/2)

- ▶ Abstraksi adalah penyederhanaan dari kenyataan yang kompleks dengan memodelkan class sesuai dengan masalah yang dihadapi.
- ▶ Abstraksi data bisa berarti bahwa memodelkan suatu benda dengan mewakili karakteristiknya dengan suatu data.
- ▶ Contoh : Benda seperti manusia bisa diabstraksikan dengan memiliki beberapa atribut seperti tinggi badan, berat badan, warna rambut dan lain-lain.

Ilustrasi Data Abstraction (2/2)

```
TManusia=class
```

```
    Nama:String;
```

```
    Tinggi:Integer;
```

```
    Berat:Double;
```

```
end;
```

```
TMahasiswa=class (TObject)
```

```
    NIM:String;
```

```
    Nama:String;
```

```
    TempatLahir:String;
```

```
    TanggalLahir:TDate;
```

```
end;
```

Ilustrasi Encapsulation(1/2)

- ▶ Enkapsulasi berarti membungkus atribut dan method yang digunakan ke dalam class.
- ▶ Contoh : Manusia mempunyai method/operasi makan. Proses makan suatu objek mungkin berbeda dengan proses makan pada objek yang lain.

Ilustrasi Encapsulation(2/2)

```
TOrangIndonesia=class
    Nama:String;
    Tinggi:Integer;
    Berat:Double;
    procedure UcapNama;
{Method disisipkan bersama atribut dalam sebuah class}
    end;
{Implementasi method UcapNama}
procedure TOrangIndonesia.UcapNama;
begin
    writeln('Saya adalah '+nama);
end;
```

Ilustrasi Inheritance (1/2)

- ▶ Kemampuan suatu class untuk mewariskan atribut dan perilakunya kepada anak classnya (sub class).
- ▶ Class anak boleh direstruktur programnya sehingga mempunyai atribut dan perilaku tambahan sehingga tidak persis dengan class induknya.

Ilustrasi Inheritance (2/2)

Type

```
TOrangIndonesia=class (TObject)
```

```
    Nama: String;
```

```
    Tinggi: Integer;
```

TOrangIndonesia Turunan Dari
TObject

```
TOrangBetawi=class (TOrangIndonesia)
```

```
end;
```

TOrangBetawi Turunan Dari
TOrangIndonesia

TOrangSunda Turunan Dari
TOrangIndonesia

TOrangJawa Turunan Dari
TOrangIndonesia

```
Procedure TOrangIndonesia.UcapNama;
```

```
Begin
```

```
    Writeln('Nama Saya '+Nama);
```

```
end;
```

Cara UcapNama Orang Indonesia

```
end;
```

Cara UcapNama Orang Sunda

```
end;
```

Cara UcapNama Orang Jawa

Ilustrasi Polimorphism(1/2)

- ▶ Polimorpisme memperbolehkan anda untuk memperlakukan anggota(member) suatu class yang merupakan class turunan sebagai anggota class induknya.
- ▶ Kemampuan objek-objek yang mempunyai tipe data yang berbeda untuk merespon terhadap pemanggilan method dari method-method yang mempunyai nama yang sama.

Ilustrasi Polimorphism (2/2)

```
var
  Orang1:TOrangIndonesia;
  Sunda1:TOrangSunda;
  Jawa1:TOrangJawa;
begin
  Orang1:=TOrangIndonesia.Create;
  Orang1.Nama:='Andri';
  Orang1.UcapNama;

  Sunda1:=TOrangSunda.Create;
  Sunda1.Nama:='Cecep';
  Sunda1.UcapNama;

  Jawa1:=TOrangJawa.Create;
  Jawa1.Nama:='Bejo';
  Jawa1.UcapNama;

  Orang1:=Sunda1;
  Orang1.UcapNama;

  Orang1:=Jawa1;
  Orang1.UcapNama;
end.
```

Nama Saya Andri

Nami Kuring Cecep

Nama Saya Bejo

Menghidupkan Object (1/2)

- ▶ Ketika sebuah object dideklarasikan (Contoh `Sunda1:TOrangSunda`) bukan berarti objek tersebut bisa digunakan.
- ▶ Agar suatu objek dapat digunakan, maka objek tersebut harus dihidupkan terlebih dahulu.
- ▶ Jika sebuah objek digunakan tanpa dihidupkan terlebih dahulu maka akan menghasilkan error.

Menghidupkan Object (2/2)

```
var
    Orang1:TOrangIndonesia;
begin
    Orang1.UcapNama;
end.
```



ERROR.
OBJECT BELUM DIBUAT

```
var
    Orang1:TOrangIndonesia;
begin
    Orang1:=TOrangSunda.Create;
    Orang1.Nama:='Andri';
    Orang1.UcapNama;
end.
```



HIDUPKAN OBJECT



MEMANIPULASI OBJECT

Mematikan Object (1/2)

- ▶ Sebuah objek membutuhkan alokasi di memori.
- ▶ Objek yang sudah tidak digunakan lagi harus dimatikan sehingga akan mengembalikan memori yang dipakainya.
- ▶ Untuk mematikan sebuah objek maka metode yang digunakan merupakan metode khusus yang disebut destructor.
- ▶ Destructor yang digunakan adalah destructor Destroy. Tetapi Borland sendiri menyarankan untuk mematikan objek adalah dengan menggunakan metode/procedure Free.

Mematikan Object (2/2)

```
var
  Orang1:TOrangIndonesia;
begin
  Orang1:=TOrangSunda.Create;
  { Hidupkan objek }
  ...
  ...
  Orang1.Free;
  { Matikan Objek / Hapus dari Memori }
end.
```



HIDUPKAN OBJECT



MATIKAN OBJECT

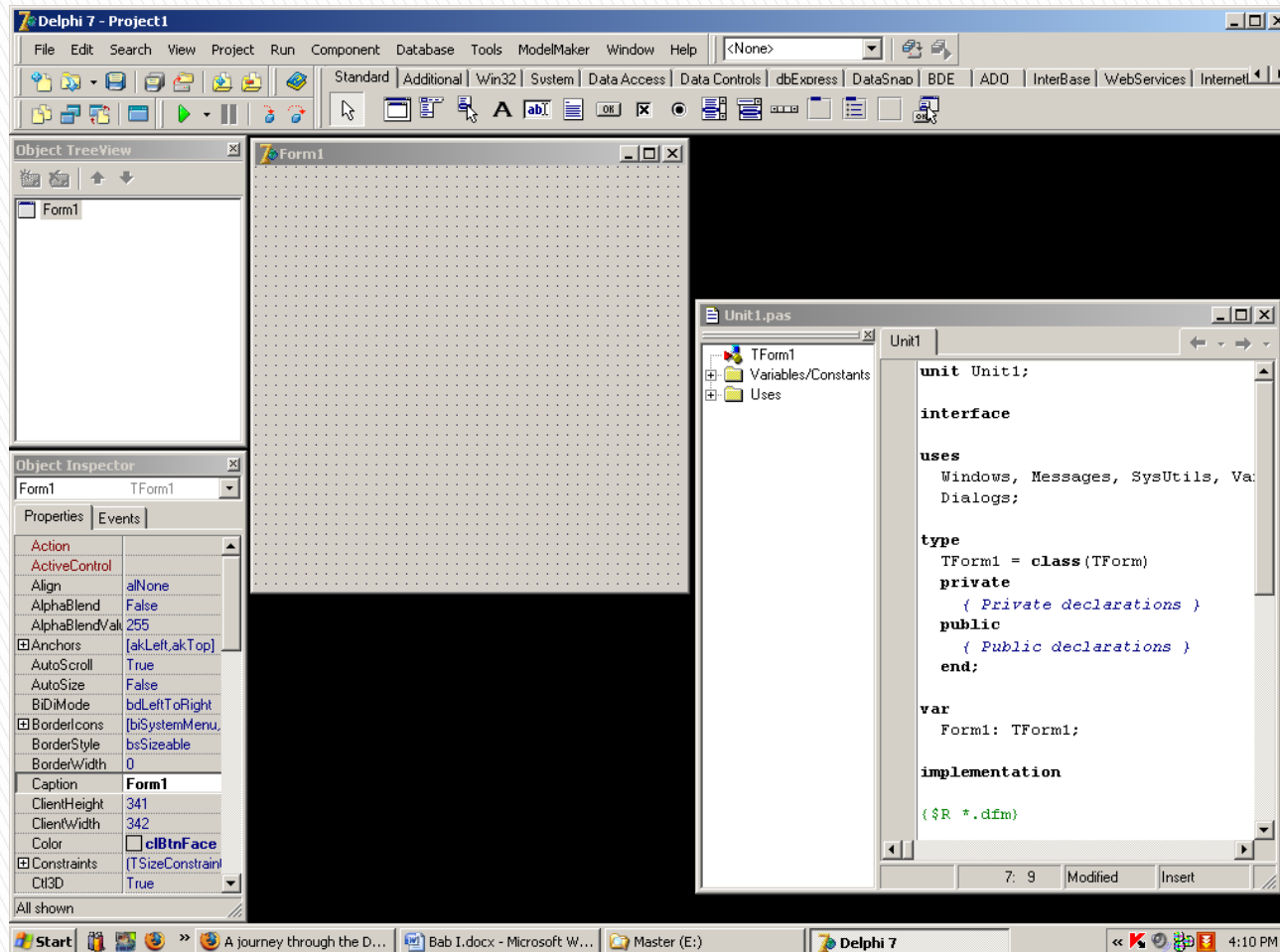
PENGENALAN IDE DELPHI

Pengenalan IDE Delphi

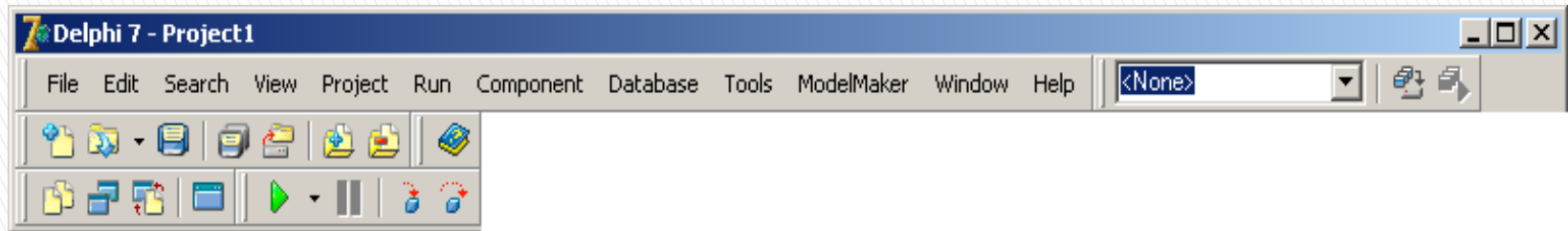
- ▶ Window Utama
- ▶ Component Palette
- ▶ Object Inspector
- ▶ Object TreeView
- ▶ Form Designer
- ▶ Code Editor

Memulai Delphi

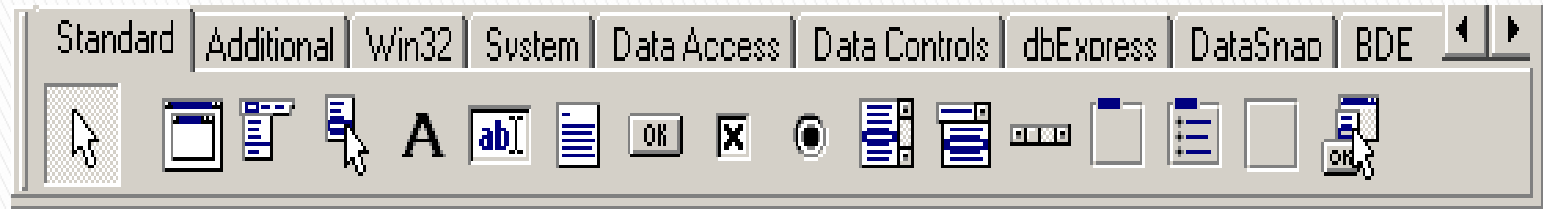
Start → Programs → Borland Delphi 7 → Delphi 7



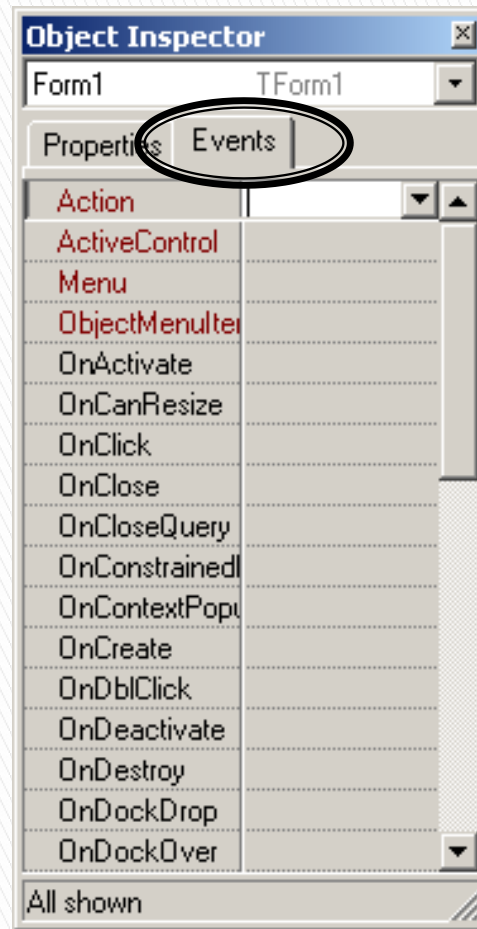
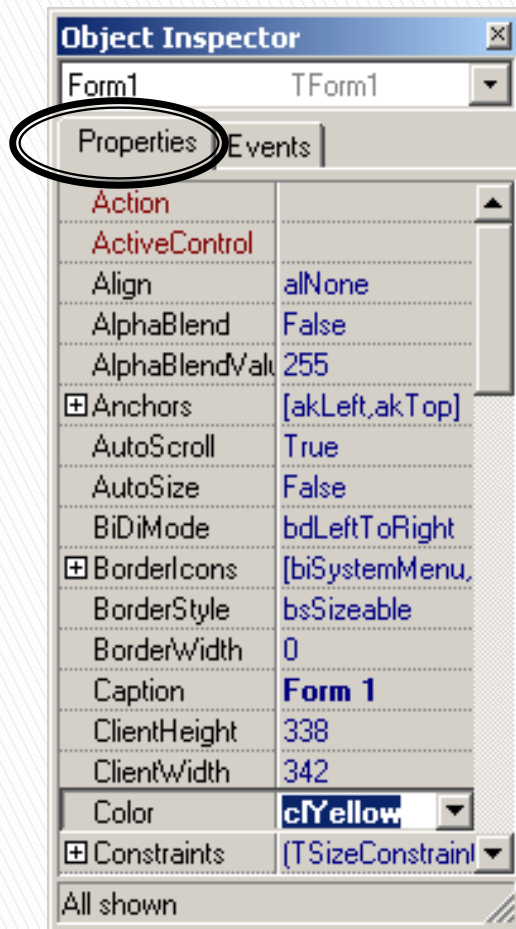
Window Utama



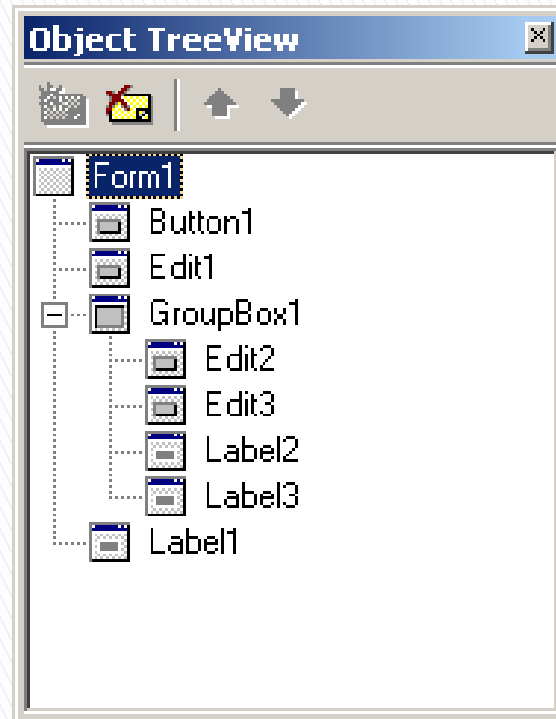
Component Palette



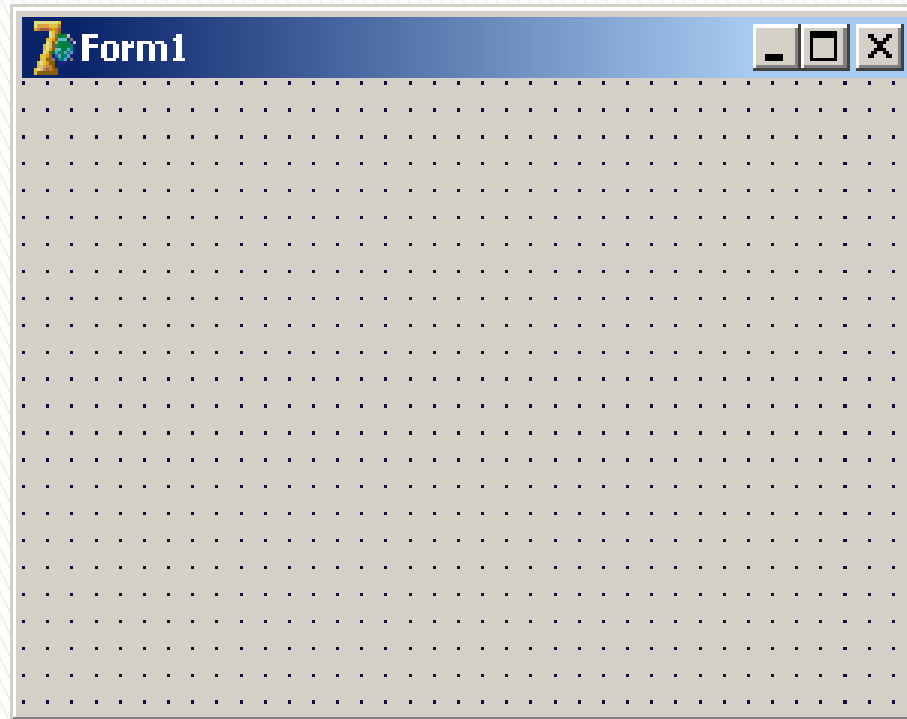
Object Inspector



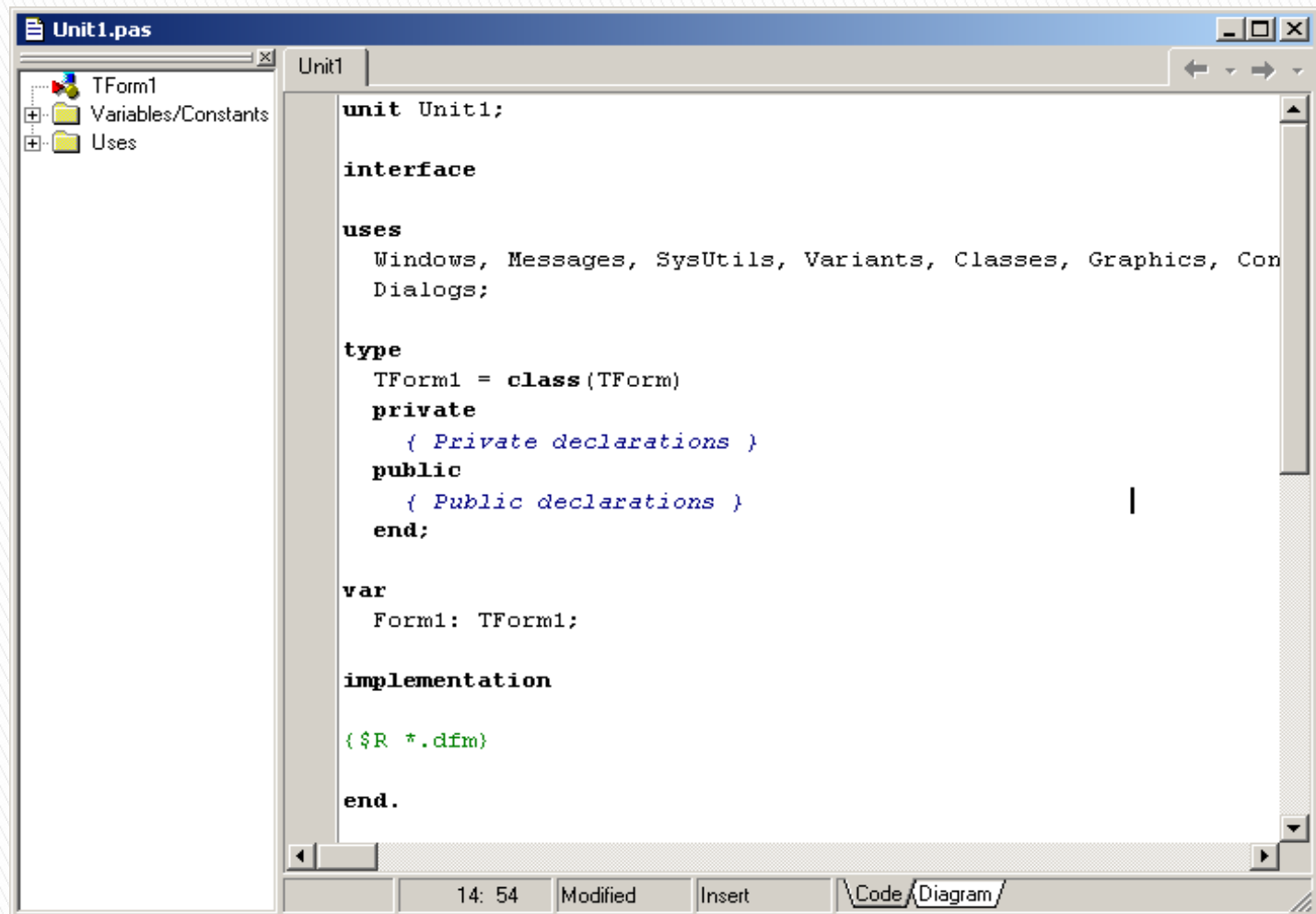
Object TreeView



Form Designer



Code Editor



FORM DAN KOMPONEN

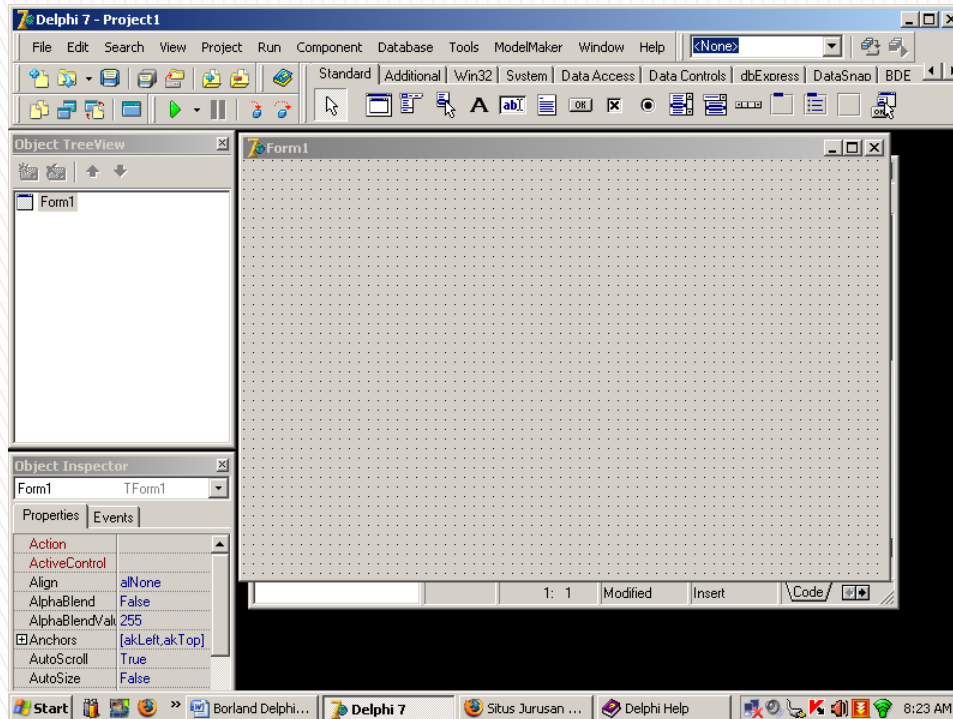
Form dan Komponen

- ▶ **Membuat Program Pertama**
 - Membuat Aplikasi Baru
 - Menyimpan Aplikasi
 - Mengeksekusi Aplikasi
- ▶ **Memodifikasi Form**
 - Memodifikasi Form Dengan Object Inspector
 - Memodifikasi Form Dengan Kode Program
- ▶ **Menambahkan Objek Lain Ke Dalam Form**

Membuat Program Pertama

Membuat Aplikasi Baru

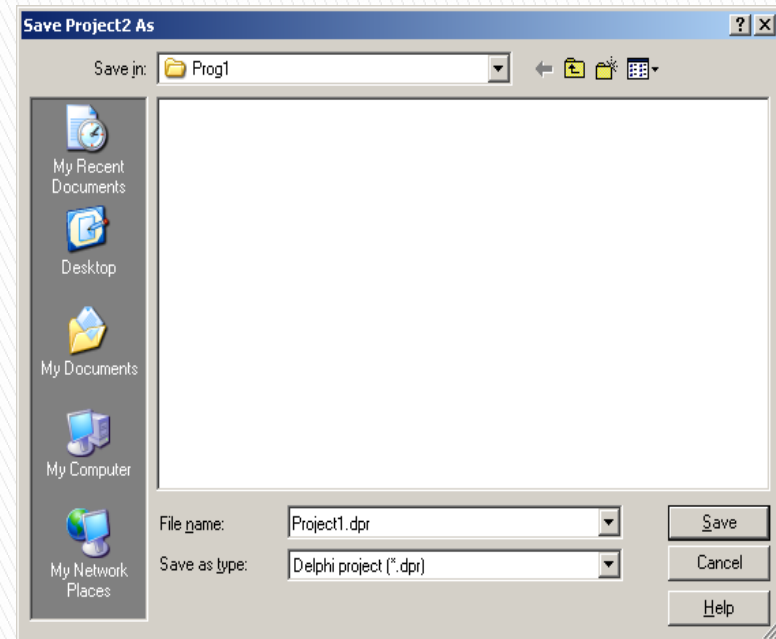
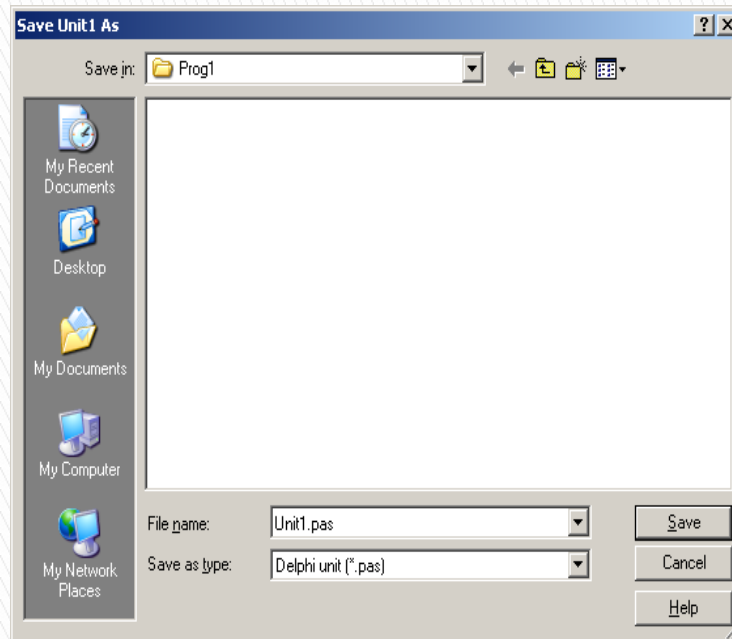
Klik menu File → New → Application



Membuat Program Pertama

Menyimpan Aplikasi

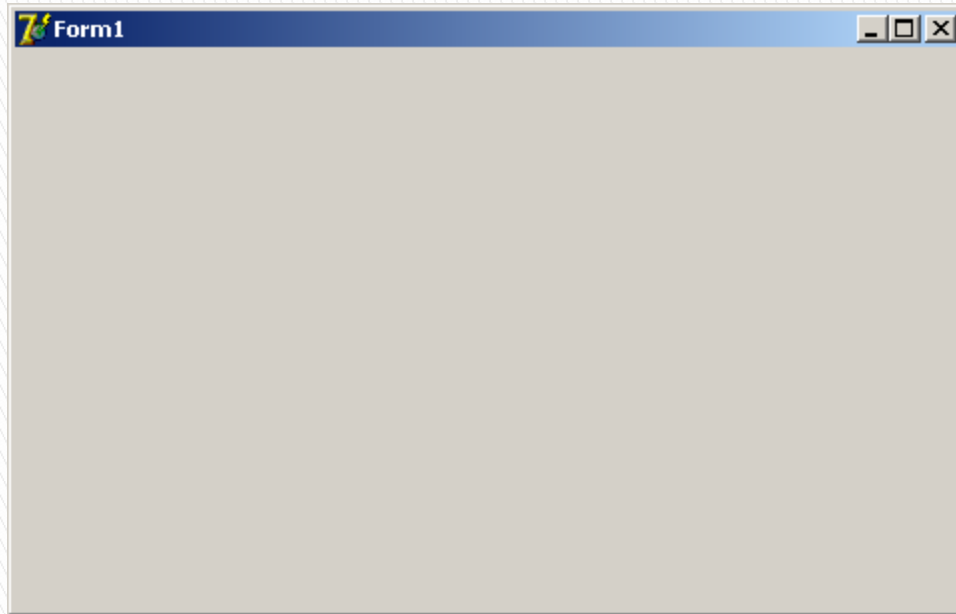
Klik menu File → Save All



Membuat Program Pertama

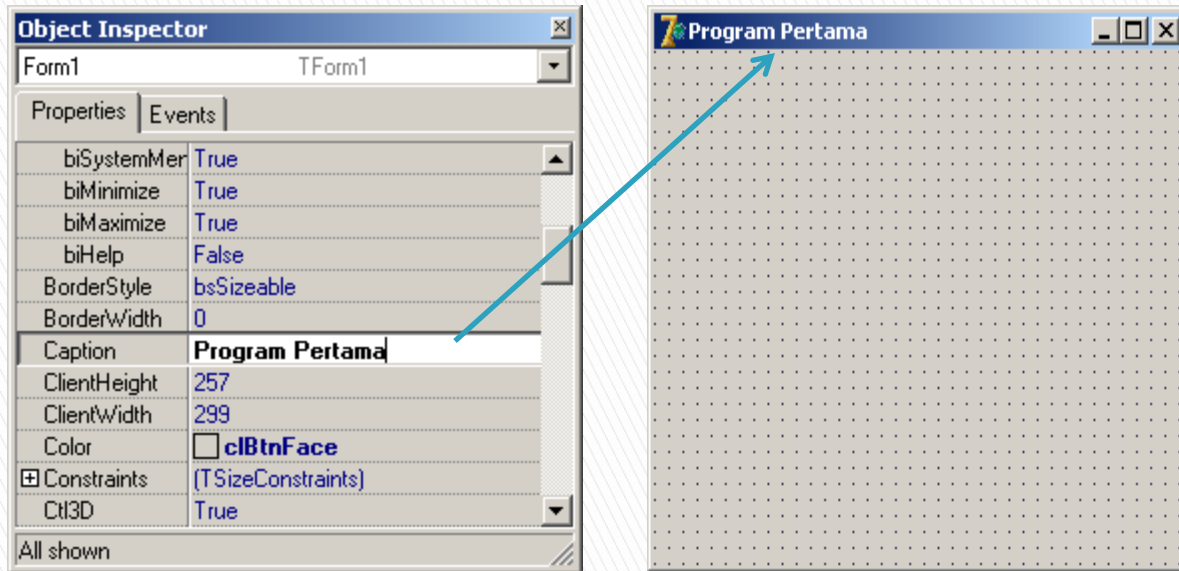
Menjalankan Aplikasi

Klik menu Run → Run (F9)



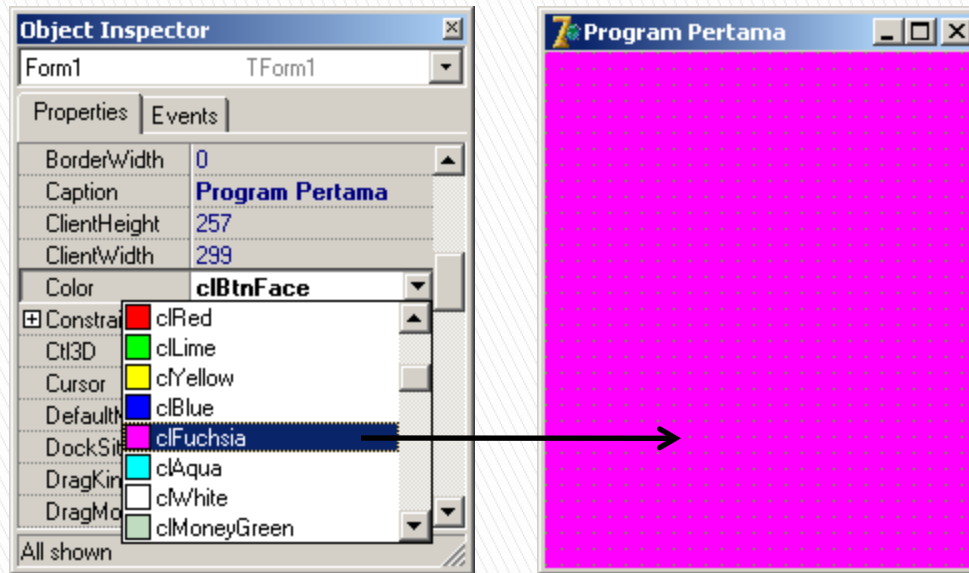
Memodifikasi Form

- ▶ Menggunakan Kode Program
 - Mengganti Judul Form



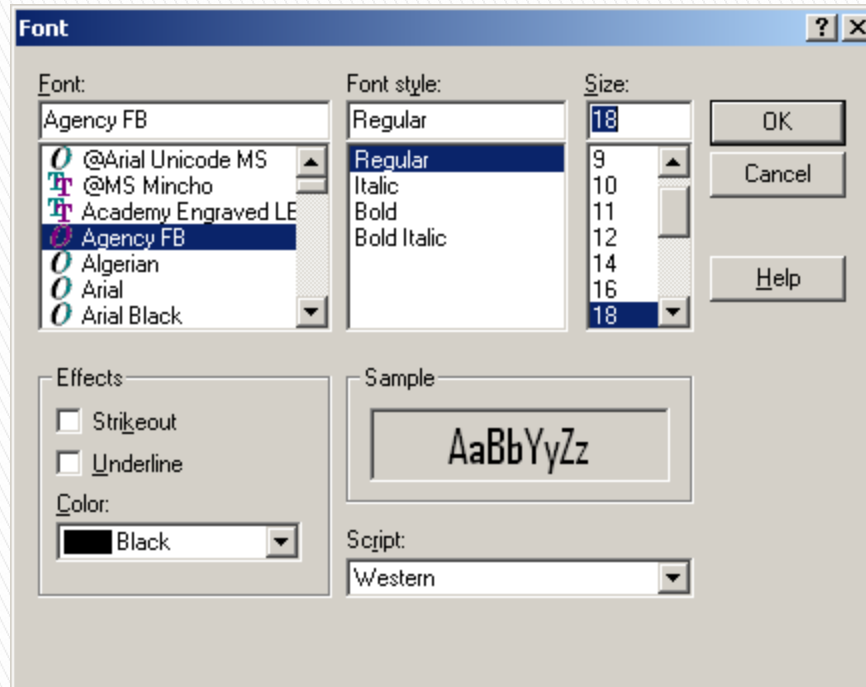
Memodifikasi Form

- ▶ Menggunakan Object Inspector
 - Mengganti Warna Form



Memodifikasi Form

- ▶ Menggunakan Object Inspector
 - Mengganti Font



Memodifikasi Form

- ▶ Menggunakan Kode Program

Contoh Kasus :

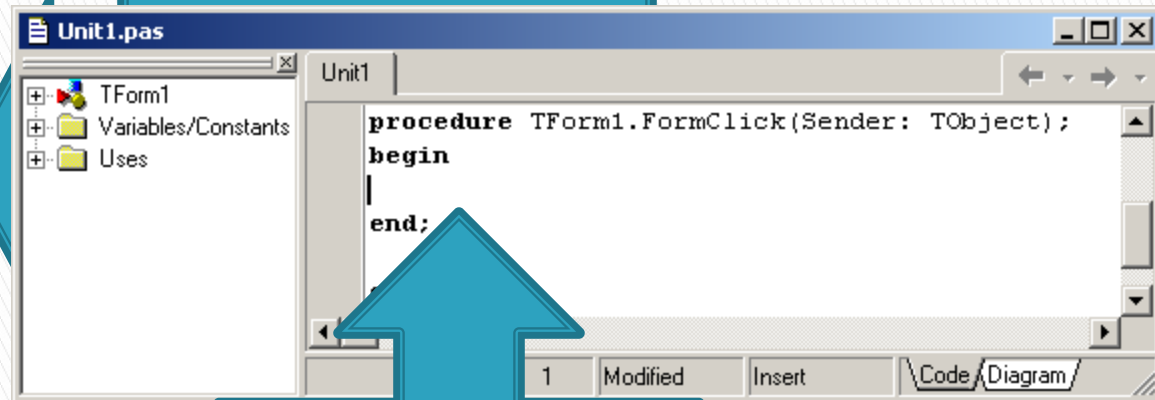
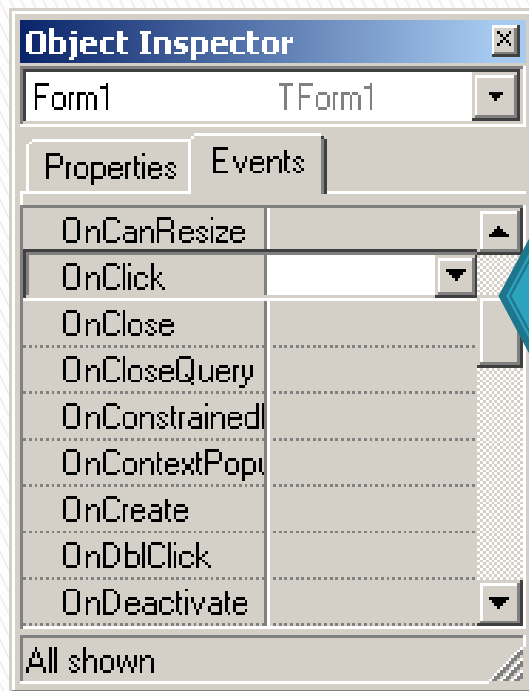
“Buatlah sebuah program yang ketika form diklik, maka warna form akan menjadi merah”

Perhatikan hal-hal berikut :

- Ketika Form diklik (Event OnClick)
- Warna Form akan menjadi merah (Color diisi merah)

Memodifikasi Form

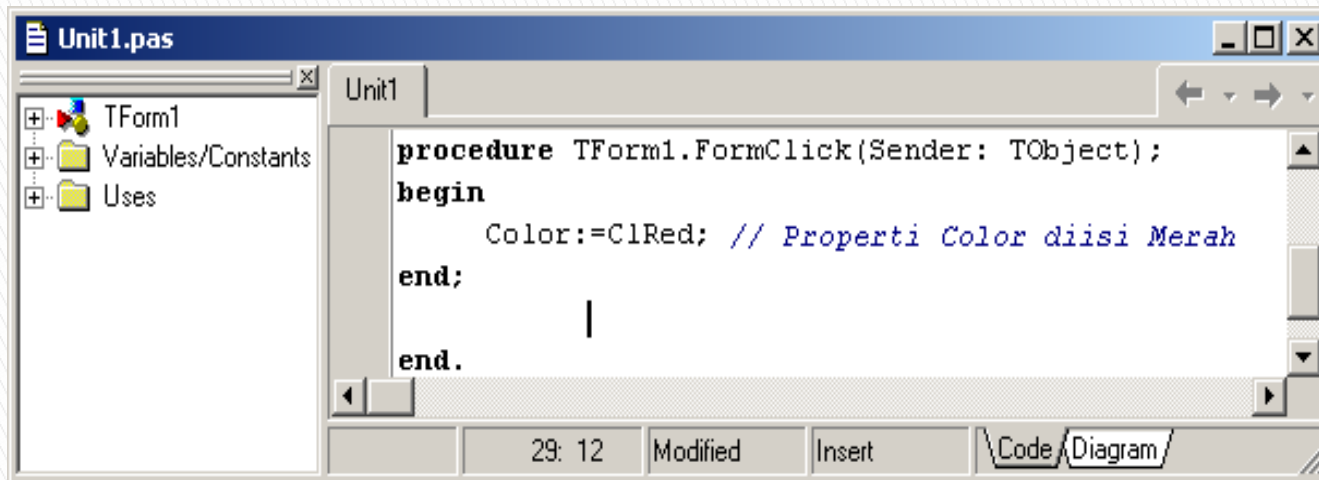
- ▶ Cari Event OnClick di Form



Isi kode programnya
disini (diantara
begin dan end)

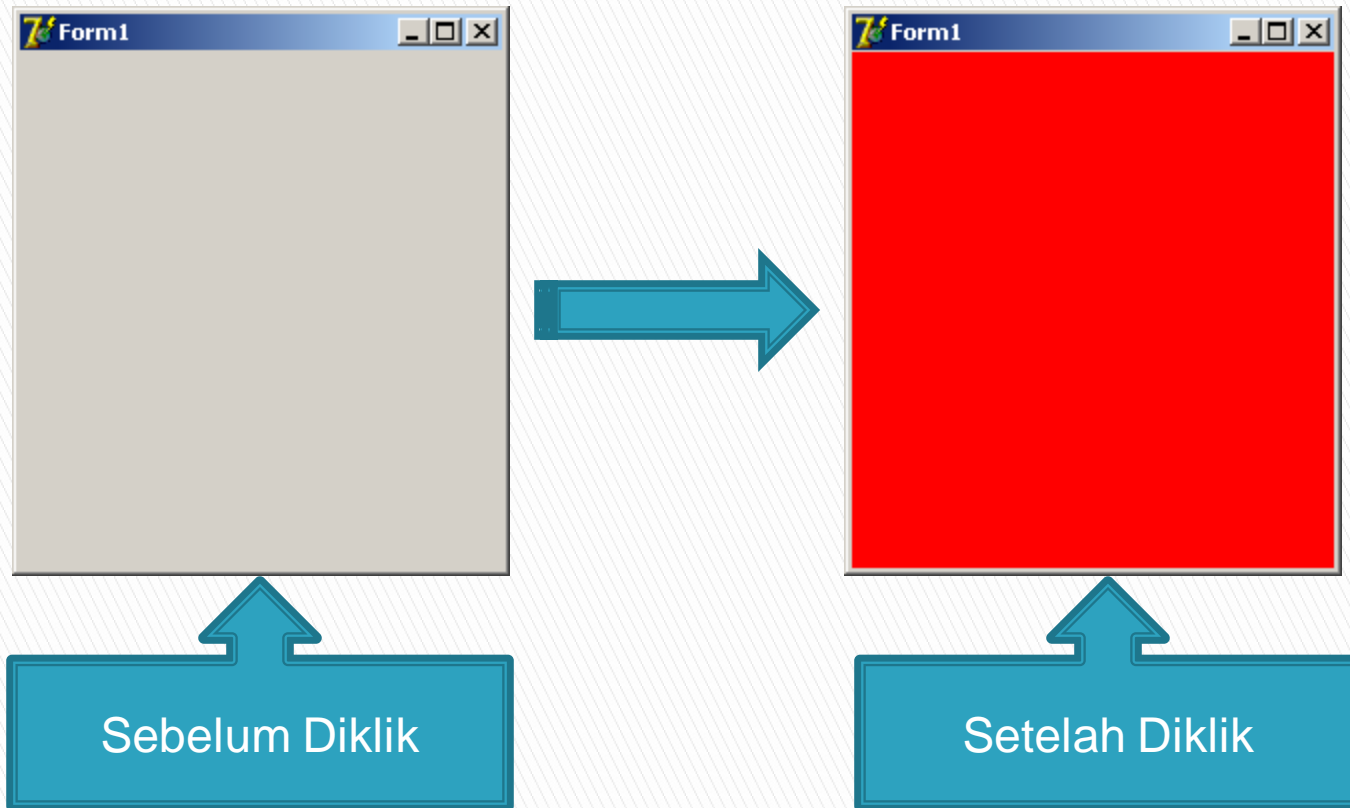
Memodifikasi Form

- ▶ Isi dengan kode program yang akan mengubah warna latar form menjadi merah.



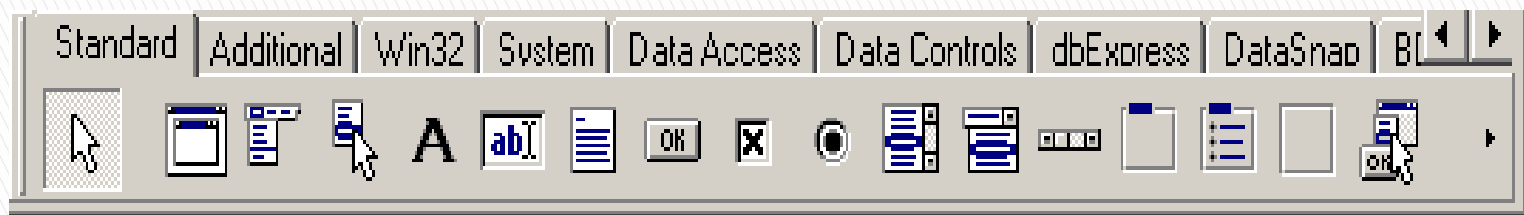
Memodifikasi Form

- ▶ Run aplikasi



Menambahkan Object Lain Ke Form

- ▶ Form merupakan “kertas kosong” yang dapat ditempati oleh komponen lain
- ▶ Komponen-komponen yang dapat digunakan dapat dilihat pada Component Palette
- ▶ Komponen yang umum banyak dipakai dalam suatu aplikasi terdapat pada tab Standar

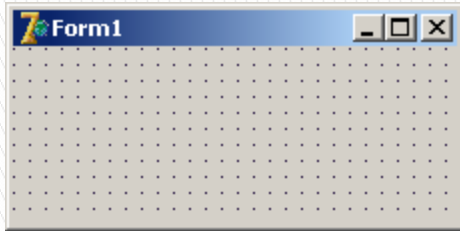


Cara Menempatkan Komponen

Contoh Kasus :

Buat sebuah program yang akan menginputkan judul untuk form pada sebuah **edit** dan jika tombol **“Ganti Judul”** diklik maka judul tersebut akan terupdate di judul Form, dan jika tombol **Keluar** diklik maka program akan ditutup

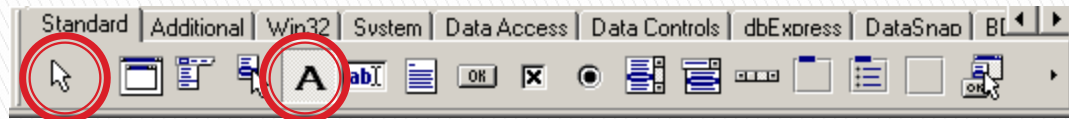
Langkah-Langkah



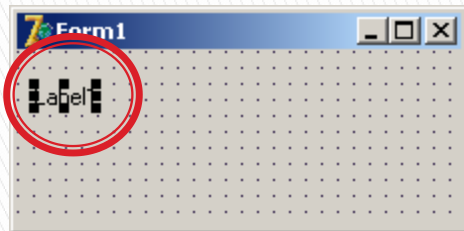
Form masih kosong



Kondisi Component Palette
Sebelum Ada Komponen
Yang Dipilih

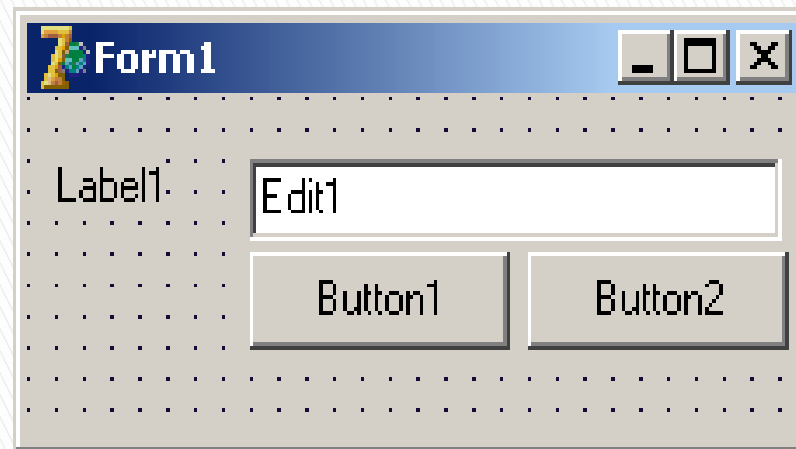


Kondisi Component Palette
setelah Ada Komponen
Yang Dipilih



Form yang telah disisip komponen Label

Ulangi untuk komponen lainnya



Ganti Properti

► Objek **Label1**

- Ganti properti **Caption** dengan **Judul Baru**
- Silahkan atur **Font**, warna dan yang lainnya, dipersilahkan.

► Object **Edit1**

- Ga
- Ga
- Sil



Pada object Edit1 anda diminta untuk mengganti Name dari sebuah object. Ini **sangat diperlukan** karena ketika object yang ada di dalam sebuah form hanya menggunakan penomoran (edit1, edit2, edit3 dst), maka akan sulit untuk mengingat fungsi dari sebuah object. Oleh karena itu gantilah Name dari sebuah object sesuai dengan kegunaannya.

► Objek

- Ga
- Ga
- Ju

Ganti

► Object **Button2**

- Ganti properti **Caption** dengan **Keluar**
- Ganti properti **Name** dengan **TblKeluar** (Tombol Keluar)

Isi Kode Program

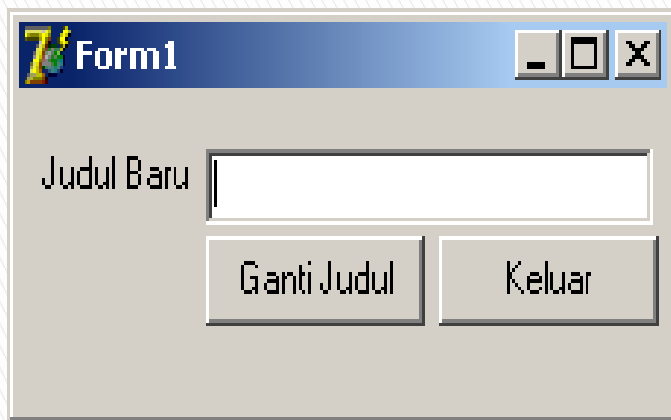
- ▶ Cari event OnClick di TblGantiJudul, isi dengan kode berikut :

```
procedure TForm1.TblGantiJudulClick(Sender: TObject);  
begin  
    Caption:=EJudul.Text;  
    // isikan isi Ejudul (text) ke judul (Caption) form  
end;
```

- ▶ Cari event OnClick di TblKeluar, isi dengan kode berikut

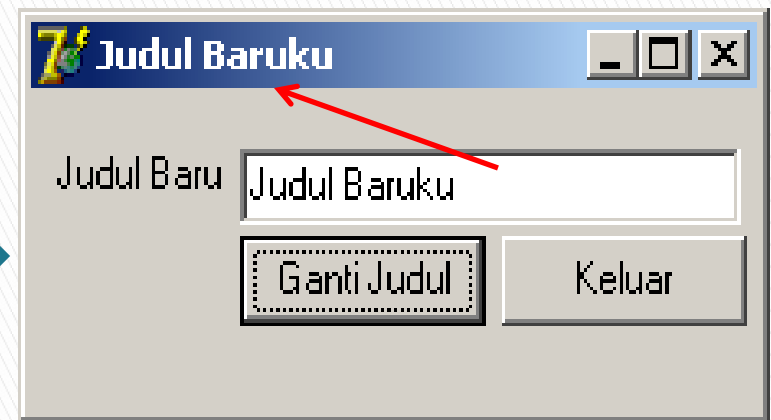
```
procedure TForm1.TblKeluarClick(Sender: TObject);  
begin  
    Close; // atau Application.Terminate;  
end;
```

Run Aplikasi



A screenshot of a Windows application window titled "Form1". It contains a text box labeled "Judul Baru" which is currently empty. Below the text box are two buttons: "Ganti Judul" and "Keluar".

Sebelum melakukan aksi
penggantian judul



A screenshot of a Windows application window titled "Judul Baruku". The text box labeled "Judul Baru" now contains the text "Judul Baruku". The "Ganti Judul" button is highlighted with a dashed border, and a red arrow points from the text box to the button title bar.

Setelah mengisi judul baru dan
mengklik tombol Ganti Judul