

MODUL PRAKTIKUM PEMROGRAMAN VISUAL



**Oleh :
Wahyu Widodo, S.Kom**

**UNIT PELAKSANA TEKNIS LABORATORIUM
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN ILMU KOMPUTER
EL RAHMA
YOGYAKARTA
2009**

BAB I

Materi :

1. Sekilas tentang Delphi
2. Mengenal IDE Delphi

Maksud dan Tujuan :

1. Mengenal tentang pemrograman Delphi sebagai visual programming
2. Mengenal tipe file utama dalam project Delphi
3. Mengenal tipe data dalam Delphi.
4. Mengenal lingkungan kerja delphi

I. Sekilas tentang Delphi

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. Dengan demikian delphi lebih mudah digunakan dibandingkan dengan pascal, karena anda hanya tinggal mengatur properti atau sifat dari objek dan menambahkan sedikit pengkodean.

Didalam delphi ada beberapa pengertian yang memerlukan penjelasan untuk memahami bagaimana dan apa saja komponen yang ada dalam delphi, adapun diantaranya adalah:

a. Project (*.dpr)

Project delphi adalah sekumpulan file yang akan membangun aplikasi atau library yang dapat disebarkan kepada pemrogram lain. Pada saat memulai delphi, delphi akan membuat sebuah project baru. Setelah itu terserah pengguna apakah ingin memodifikasi proyek tersebut, membuat proyek baru lagi atau mungkin membuka proyek yang sudah ada.

Project aplikasi terdiri atas :

1. File project : file ini tersimpan dengan ekstensi .dpr, hanya ada satu file proyek untuk satu proyek.
2. File form : file form tersimpan dengan ekstensi .dfm, adalah file biner yang berisi gambaran grafis dari form. Setiap file .dfm selalu paralel dengan satu unit file yang berekstensi .pas.
3. Unit (source) : file unit tersimpan dengan ekstensi .pas, dimana setiap unit berupa file source code dalam bahasa pascal. Tidak setiap file unit memiliki atau terhubung file .dfm

File proyek menggabung semua file diatas yang digunakan oleh delphi untuk melakukan kompilasi dan membuat file target (.exe atau .dll).

b. Form (*.dfm)

Form merupakan tempat kita merancang program. Jika seorang pemogram diibaratkan sebagai seorang pelukis maka form inilah yang menjadi kanvasnya. Form yang dirancang dalam delphi bisa digunakan kembali oleh proyek delphi yang lain. Form dapat juga disimpan dalam bentuk dll, sehingga dapat digunakan oleh aplikasi yang lain seperti c++, paradox, atau dbase. Delphi menawarkan kemudahan pembuatan form melalui model template.

c. Unit (*.pas)

Unit merupakan modul yang terpisah yang dapat dikompilasi tersendiri, yang berisi deklarasi bagian publik (antarmuka) dan bagian privat (implementasi). Setiap form dalam delphi mempunyai satu unit yang bersesuaian. Source code unit tersimpan dengan ekstensi pas, sedangkan yang sudah terkompilasi tersimpan dengan ekstensi .dcu. proses link akan menggabung file .dcu menjadi satu file exe atau dll.

Tipe data

Setiap data yang disimpan di memori komputer harus memiliki tipe data.

a. Tipe Integer

Type	Rentang Nilai	Byte
Byte	0...255	1
Word	0...55535	2
LongWord	0...424967295	4
ShortInt	-128...127	1
SmallInt	-32768...32767	2
LongInt	-2147483648...2147483647	4
Integer	-2147483648...214783647	4
Cardinal	0...4294967295	4
Int64	-2 ⁶³ ...2 ⁶³⁻¹	8

b. Tipe Real

Type	Rentang Nilai	Ketelitian (digit)	Byte
Real48	2.9 x 10 ⁻³⁹ ... 1.7 x 10 ³⁸	11 – 12	6
Single	1.5 x 10 ⁻⁴⁵ ... 3.4 x 10 ³⁸	7 – 8	4
Double	5.0 x 10 ⁻³²⁴ ...1.7 x 10 ³⁰⁸	15 – 16	8
Real	5.0 x 10 ⁻³²⁴ ... 1.7 x 10 ³⁰⁸	15 – 16	8
Extended	3.6 x 10 ⁻⁴⁹⁵¹ ...1.1 x 10 ⁴⁹³²	19 – 20	10
Comp	-2 ⁶³⁺¹ ... 2 ⁶³⁻¹	19 – 20	8
Currency	-922337203685477.5808... 922337203685477.5807	19 - 20	8

c. Tipe Boolean

Tipe	Byte
Boolean	1
ByteBool	1
WordBool	2
LongBool	4

d. Tipe Character

Tipe	Byte
Char	1
AnsiChar	1
WideChar	2

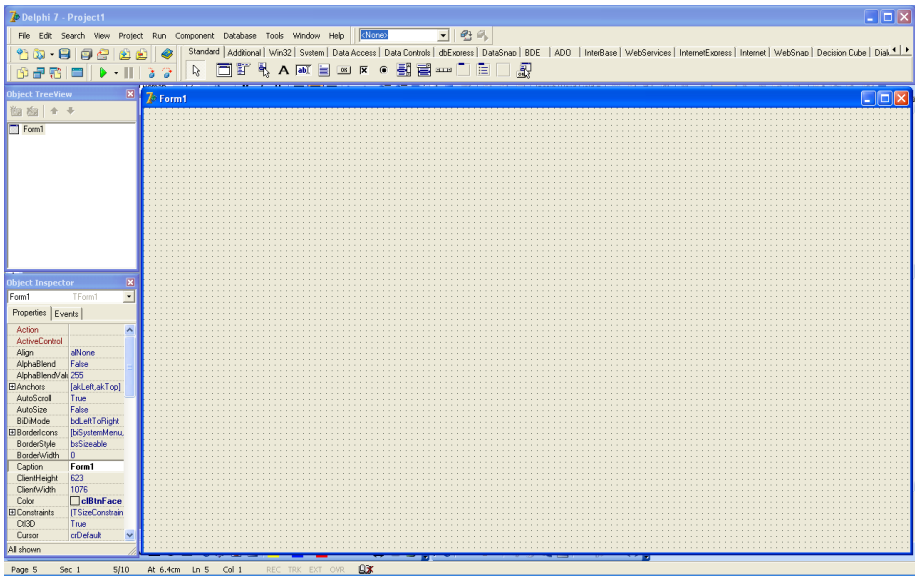
e. Tipe String

Tipe	Byte	Isi Maksimum
ShortString	2 sd. 256	256 karakter
AnsiString	4 sd. 2 GB	2 ³¹ karakter
WideString	4 sd. 2 GB	2 ³⁰ karakter

II. Mengetahui lingkungan kerja delphi

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.

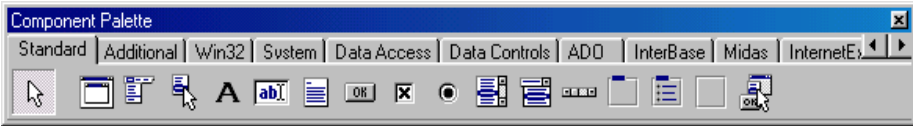
Tampilan interface delphi adalah sebagai berikut :



Gambar 1.0 .Tampilan IDE delphi.

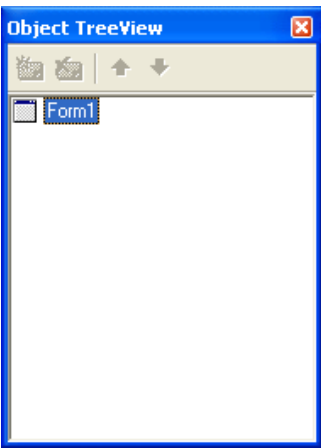
Untuk menjalankan:

1. Klik tombol **Start**, pilih menu **Programs → Borland Dephi 7 → Delphi 7**
2. Tunggu hingga program delphi aktif
3. Perhatikan komponen yang ada, diantaranya component pallete , berisi sekumpulan objek yang akan kita gunakan dalam pembuatan program. Kita juga dapat menambah komponen tambahan yang tidak disertakan dalam paket delphi atau yang lebih sering dikenal dengan *plugin vcl (visual component library) eksternal*.



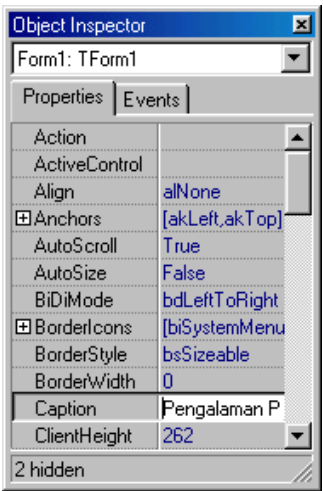
Gambar 1.1.Toolbar Component Pallete.

4. Berikut ini adalah object treeview, digunakan untuk memonitoring suatu object yang terdapat pada project yang sedang kita buat. Semua komponen yang diletakan pada lembaran form akan terlihat di dalam object treeview.



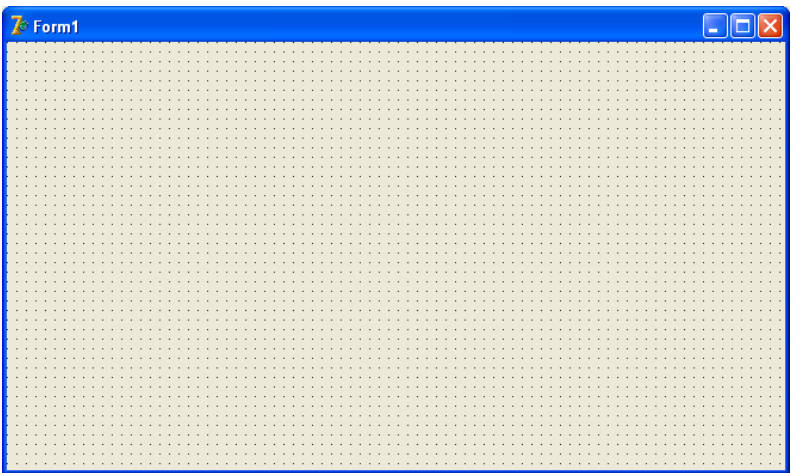
Gambar 1.2.Object treeview.

5. Object inspector (berisi daftar property atau sifat dari objek yang aktif yang dapat kita ubah sesuai dengan kebutuhan dalam program kita). Kedua objek ini merupakan komponen yang paling sering dipakai dalam pembuatan program visual.



Gambar 1.3. Object Inspector

6. Pada gambar dibawah ini adalah form atau lembar kerja yang digunakan untuk meletakkan komponen.



Gambar 1.4. Lembar kerja form

BAB II

Materi :

- 1. Instalasi zeos
- 2. Koneksi Delphi ke mysql dengan zeos

Maksud dan Tujuan :

- 1. Setelah bab ini berakhir diharapkan siswa dapat melakukan instalasi zeos di Delphi atau komponen tambahan lainnya (*additional vcl*)
- 2. Studi kasus mengkoneksikan database mysql dengan menggunakan zeos

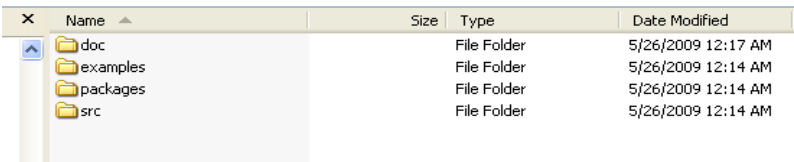
I. Instalasi Zeos

Delphi merupakan rapid programming dimana terdapat banyak komponen-komponen add-in yang dapat ditambahkan di delphi. komponen-komponen tersebut ada yang gratis maupun berbayar.

Zeos merupakan salah satu komponen gratisan yang digunakan sebagai konektor untuk menghubungkan aplikasi dengan database, fasilitasnya cukup komplit karena dapat handle koneksi mulai dari MySQL sampai Oracle. komponen ini dapat didownload di <http://sourceforge.net/projects/zeoslib>, penggunaannya pun sangat mudah dan tidak memerlukan banyak waktu untuk melakukan koneksi dengan database. dalam penggunaannya, zeos tidak memerlukan **ODBC (Open Database Connectivity)**. Untuk masalah kehandalan, zeos dapat dikatakan cukup handal sebagai komponen konektor database, selain itu para pengembang zeos juga masih selalu aktif untuk melakukan perubahan-perubahan versi yang dapat dikatakan bahwa zeos sendiri masih hidup.

Langkah –langkah instalasi zeos :

- 1. Ekstrak zeos dalam sebuah folder sehingga terdapat 4 folder seperti pada gambar dibawah ini :

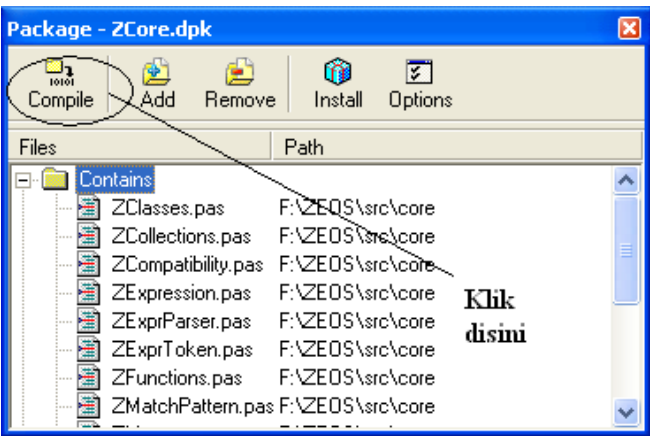


Name	Size	Type	Date Modified
doc		File Folder	5/26/2009 12:17 AM
examples		File Folder	5/26/2009 12:14 AM
packages		File Folder	5/26/2009 12:14 AM
src		File Folder	5/26/2009 12:14 AM

Gambar 2.5. Ekstrak zeos

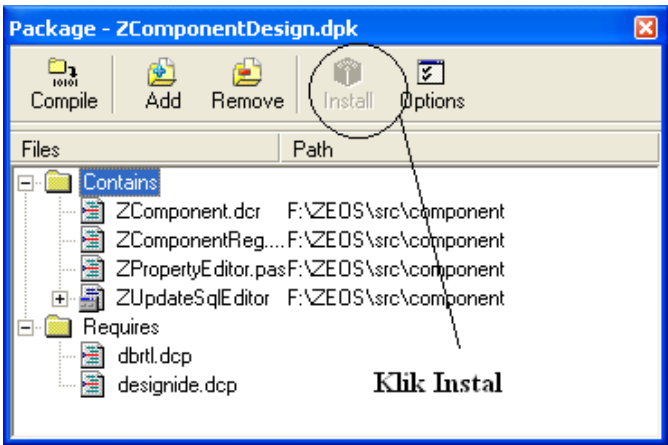
- 2. Compile semua file dibawah ini, yang terdapat pada folder package versi delphi 7, dengan mendouble klik :

- ZCore.bpl
- ZParseSql.bpl
- ZPlain.bpl
- ZDbc.bpl
- ZComponent.bpl



Gambar 2.6. Compile package delphi

3. Setelah semua list tersebut diatas selesai dcompile, selanjutnya install ZComponentDesign.bpl. yang terdapat pada package yang sama.



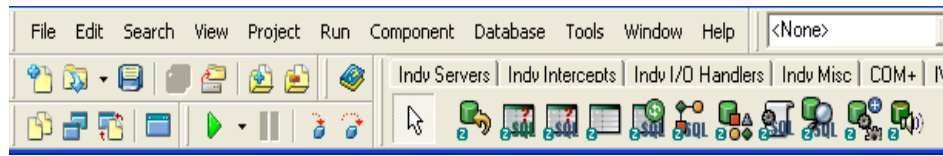
Gambar 2.7. Install ZComponent design

Apabila berhasil akan muncul kotak dialog yang menunjukkan bahwa penginstalan telah selesai dijalankan seperti gambar berikut :



Gambar 2.8. Pesan instalasi zeos berhasil

4. Setelah selesai melakukan proses penginstalan, arahkan library path delphi ke dalam direktori zeos, untuk delphi 7 dapat dilakukan dengan memilih menu *Tools->Environment Options* lalu pilih tab *library* dan tekan tombol yang menunjukkan *library path* dan akan muncul kotak diaog dan klik tombol dengan caption “...” lalu arahkan ke direktori <Folder Zeos>packages>delphi7>build lalu tekan OK dan klik tombol Add.
5. Selanjutnya anda dapat melihat pada toolbar komponen pallette, di tab zeos telah terinstall :



Gambar 2.9. Component palette zeos yang sudah terinstal

II. Koneksi database mysql dengan zeos

Untuk membuat aplikasi terkoneksi dengan database, kita gunakan TZConnection sebagai pusat koneksi dengan database. Berikut ini adalah gambar skemanya koneksi zeos dan mysql :



Gambar 2.10. Skema diagram zeos

Langkah – langkah untuk mengimplementasikan koneksi :

1. Buat database di mysql dengan nama : *pakar_db*.
2. Restore dump file, yang berisi data dan struktur table penyakit.
3. Setelah table terpasang dalam database, klik file > new > application.
4. Siapkan komponen berikut ini kedalam form :
 - ZConnection1 dan ZQuery yang terdapat pada tab zeos acces.
 - Datasource1 yang terdapat pada tab data acces.
 - DBGrid1 yang terdapat pada tab data controls
5. Klik komponen ZConnection dan atur propertinya pada object inspector sebagai berikut :

- User : *root* <atau sesuai dengan konfigurasi user MySQL anda>
- Protocol : *mysql*
- Port : *3306* <atau sesuai dengan konfigurasi port MySQL anda>
- HostName : *localhost* <bisa diisi dengan IP komputer>
- Database : *pakar_db* <contohnya>
- Connected : *True*

Apabila Connected dapat *True* dan tidak ada pesan error maka selamat anda telah berhasil mengkoneksikan aplikasi dengan MySQL dengan baik tetapi apabila tidak maka silahkan periksa kembali settingan yang anda buat apakah telah sesuai atau belum.

6. Klik komponen TZQuery kemudian aturlah property pada object inspector sebagai berikut :
 - Name : QPenyakit (atau sesuai selera anda :D)
 - Connecton : ZConnection1 (sesuai dengan nama yang anda buat pada komponen TZConnection)

6. Klik komponen TDataSource ke dalam aplikasi dan set properti seperti berikut :
- Name : dsPenyakit
 - DataSet : QPenyakit
7. Lalu atur juga komponen TDBGrid set properti DataSource menjadi dsPenyakit, untuk melakukan aksi pengaksesan kita tambahkan Button dengan properti Name menjadi btnTampil, rancangan Formnya adalah sebagai berikut :

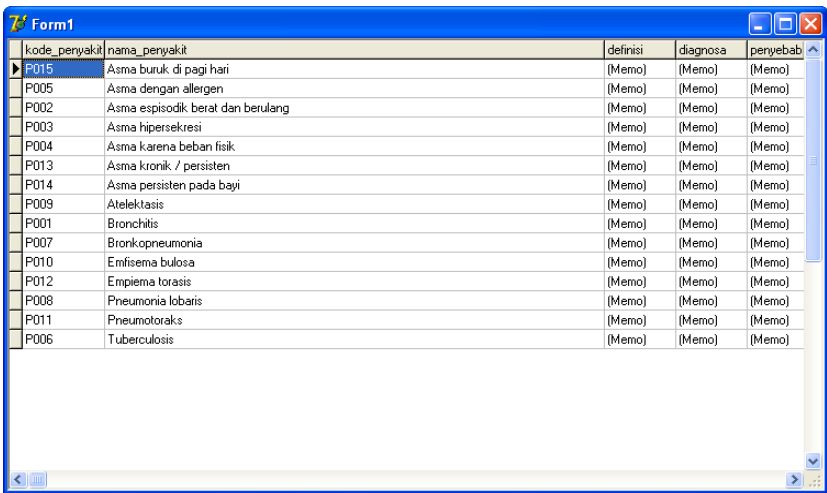


Gambar 2.11. DbGrid di form

8. Berikan kode pada event OnClick milik btnTampil sebagai berikut :

```
procedure TForm1.btnTampilClick(Sender: TObject);
begin
    with QPenyakit do
    begin
        Close;
        SQL.Clear;
        SQL.Text := 'SELECT * FROM penyakit ORDER BY
nm_penyakit ASC';
        Open;
    end;
end;
```

Lalu coba jalankan aplikasi dengan menekan tombol pada keyboard F9 dan apabila telah berjalan tekan tombol Tampilkan dan apabila telah benar maka akan keluar data pada grid yaitu sebagai berikut :



Gambar 2.12. Hasil menampilkan data penyakit

BAB III

Materi :

- 1. Manajemen file
- 2. Mendesain tampilan aplikasi
- 3. Mengenal eksepsi

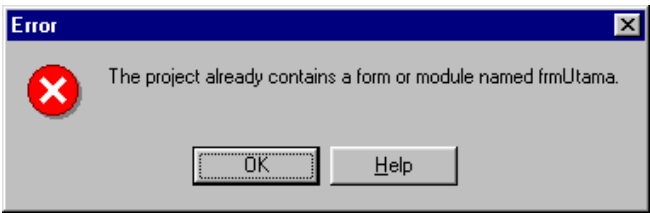
Maksud dan Tujuan :

- 1. Setelah bab ini berakhir diharapkan siswa memanajemen file project yang akan dibuat / didevelop agar lebih terstruktur.
- 2. Diharapkan mahasiswa dapat mendesain tampilan aplikasi dengan memanfaatkan property object komponen pada object inspector.
- 3. Diharapkan mahasiswa dapat mengenal ekpsepsi dan mengimplementasikan pada beberapa kasus dipemrograman.

I. Manajemen File

Dalam setiap pembuatan program delphi, saat pertama kali kita menyimpan, minimal kita melakukan 2 kali penyimpanan. Yang pertama adalah penyimpanan Project dan yang kedua adalah form. Untuk memastikan semua form dalam object tersimpan, pastikan anda selalu memilih Save All. Berikut ini langkah-langkahnya:

- 1. Pilih menu File – Save All.
- 2. Isikan nama form yang akan disimpan. Catatan: Ekstensi dari form adalah pas, dan anda tidak boleh menyimpan nama file sama dengan nama property. Misalkan jika nama form anda adalah frmUtama, maka jika anda menyimpan form tersebut dengan nama frmUtama juga, maka akan muncul pesan kesalahan.



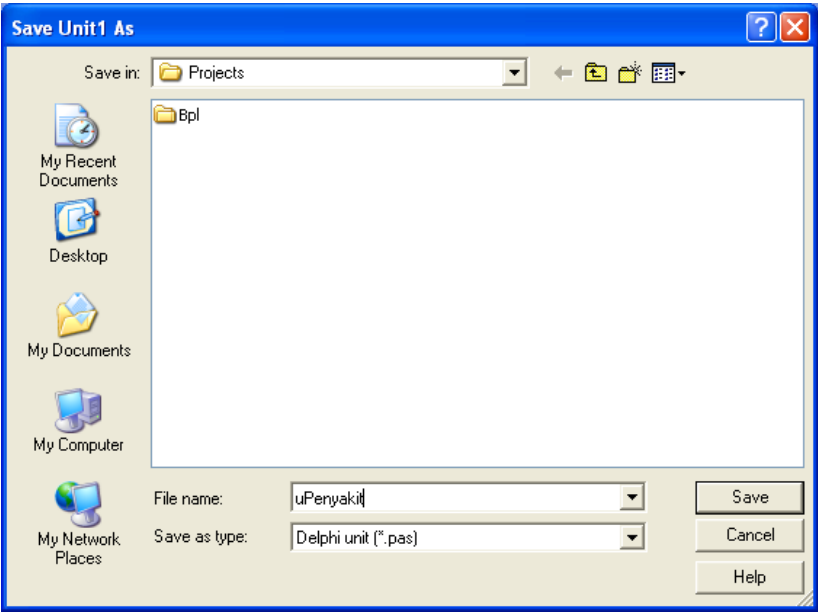
Gambar 3.13. Jendela Pesan Kesalahan

Aturan penyeragaman penamaan :

Kaidah aturan penamaan tidak ada keharusan, namun untuk lebih mudah dalam menyelesaikan sebuah project yang dilakukan secara team, penyeragaman penamaan ini sangat digunakan.

No.	Komponen	Penamaan	Contoh
1.	Project	Diawali prNama	prPakar
2.	Unit	Diawali uNama	uMain, uPenyakit
3.	Form	Diawali frmNama	frmUtama, frmGejala
4.	TEdit	Diawali edNama	edGejala
5.	TCombobox	Diawali cbNama	cbPenyakit
6.	TMemo	Diawali meNama	mePenyakit
7.	TButton	Diawali btnNama	btnSimpan

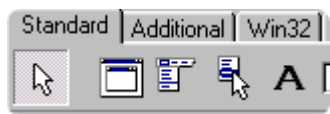
3. Untuk menyimpan unit, bisa menambah awalan huruf u dan dilanjutkan nama modulasi yang sedang dibuat. Misal : *uPenyakit*.



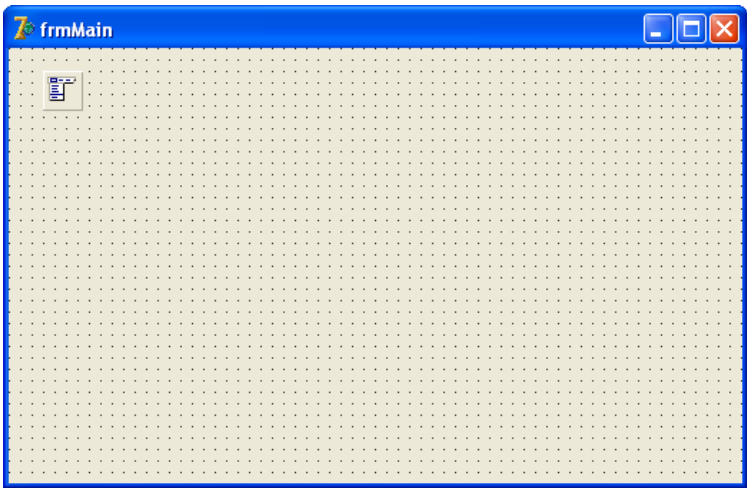
Gambar 3.14. Jendela Pesan Kesalahan

II. Mendesain tampilan aplikasi

- 1. Buatlah sebuah aplikasi dengan property sebagai berikut :
Nama project : prPakar
Nama form : frmMain
Nama unit : uMain.
- 2. Dari tab standart component palate, klik komponen MainMenu dan letakkan pada frmMain yang telah anda buat, seperti pada gambar dibawah ini.

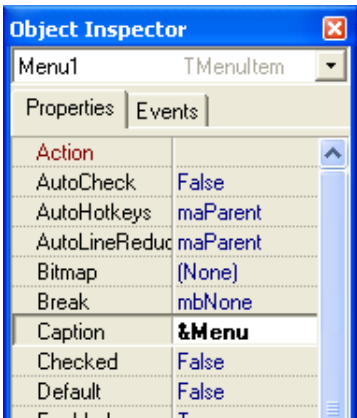


Gambar 3.15. Komponen main menu



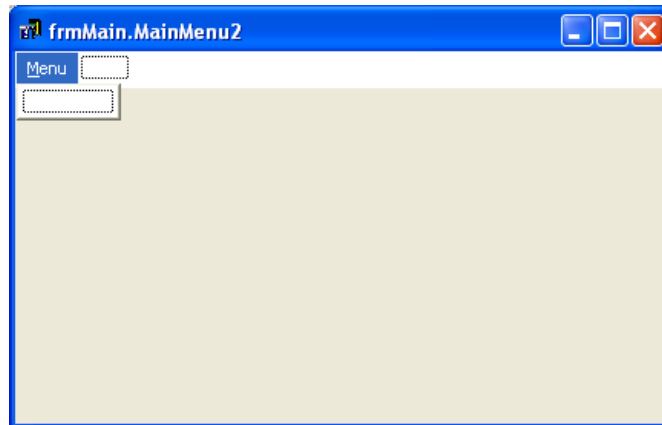
Gambar 3.15. Gambar form

- 3. Double klik komponen MainMenu yang ada pada form sehingga muncul cendela baru.
- 4. Aktifkan object inspector, kemudian ketik: &Menu pada bagian caption, kemudian tekan tombol Enter.



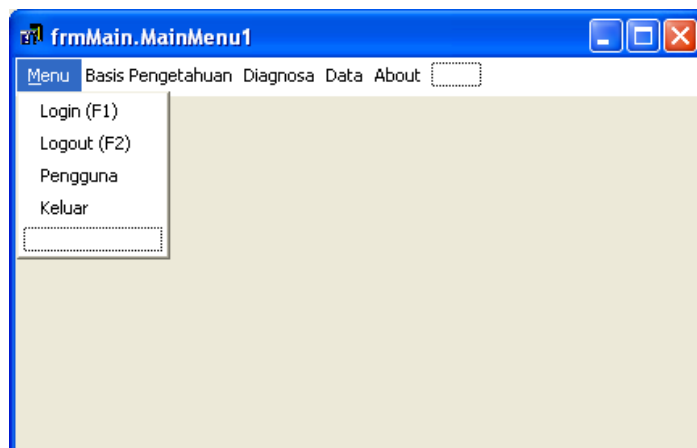
Gambar 3.16. Caption pada object inspector

5. Perhatikan pada jendela disebelah kanannya. Seharusnya muncul tulisan Menu.



Gambar 3.17. Jendela pembuatan menu

6. Selanjutnya, klik pada bagian bawah tulisan Menu, kemudian pada caption di object inspector ketik Login (F1).



Gambar 3.17. Menu yang telah dilengkapi

7. Ulangi langkah diatas untuk pembuatan menu selanjutnya. Struktur menu pada gambar diatas adalah sebagai berikut :
- a. Menu
 1. Login (F1)
 2. Logout (F2)
 3. Pengguna
 4. Keluar
 - b. Basis Pengetahuan
 1. Data
 - 1.1. Penyakit
 - 1.2. Gejala
 2. Basis Aturan
 - c. Diagnosa
 1. Registrasi
 2. Diagnosa Penyakit
 - d. Data
 1. Pasien

- 2. Laporan
 - 2.1. Pasien
 - 2.2 Hasil Diagnosa
 - e. About
8. Untuk kembali keform utama anda tutup jendela pembuatan menu tadi.

III. Mengenal eksepsi

Eksepsi merupakan sebuah mekanisme untuk mengatasi kesalahan sewaktu program di eksekusi. Dengan memberikan kode pada penanganan eksepsi, suatu tindakan dapat diatur sendiri oleh pemrogram sekiranya kesalahan – kesalahan yang tidak normal terjadi.

Delphi menyediakan dua konstruksi yang terkait dengan eksepsi, yaitu berupa :

- **try...except** dan
- **try...finally**

Bentuk umum try...except adalah sebagai berikut :

```
try
    pernyataan_1;
except
    pernyataan_2;
end;
```

Pada bentuk diatas, sekiranya terjadi kesalahan pada pernyataan 1, maka yang akan dijalankan adalah pernyataan 2. Contoh penerapan try ...except, akan kita terapkan pada proses menjalankan sintak query seperti dibawah ini :

```
function CommandSQL(Query: TQuery ; _SQL: string ; cari: Boolean):
boolean;
begin
    try
        result:=True;
        with Query do begin
            Close;
            SQL.Clear;
            SQL.Add(_SQL);
            if cari then
                Open
            else
                ExecSQL;
        end;
    except
        result:=False;
    end;
end;
```

Bentuk umum try...finally adalah sebagai berikut :

```
try
    pernyataan_1;
finally
    pernyataan_2;
end;
```

Pada bentuk diatas, sekiranya pada pernyataan 1 selesai dijalankan, maka pernyataan 2 dilanjutkan. Contoh penerapan try ...finally, akan kita terapkan pada proses pemanggilan form seperti dibawah ini :

```
procedure BukaForm(nmForm : TForm; TnmForm : TFormClass);
begin
    nmForm := TnmForm.Create(Application);
    try
        nmForm.ShowModal;
```

```
    finally  
        nmForm.Free;  
    end;  
end;
```

Pemilihan penggunaan eksepsi tersebut diatas, disesuaikan dengan keadaan yang dihadapi oleh pemrogram. Bila ingin meyakinkan bahwa apapun yang terjadi suatu kode harus dieksekusi (missal pembebasan memory seperti sintak diatas), maka gunakan *try...finally*.

Sedang untuk mengatur agar suatu kode dieksekusi manakala sesuatu eksepsi terjadi, maka gunakan *try...except*.

BAB IV

Materi :

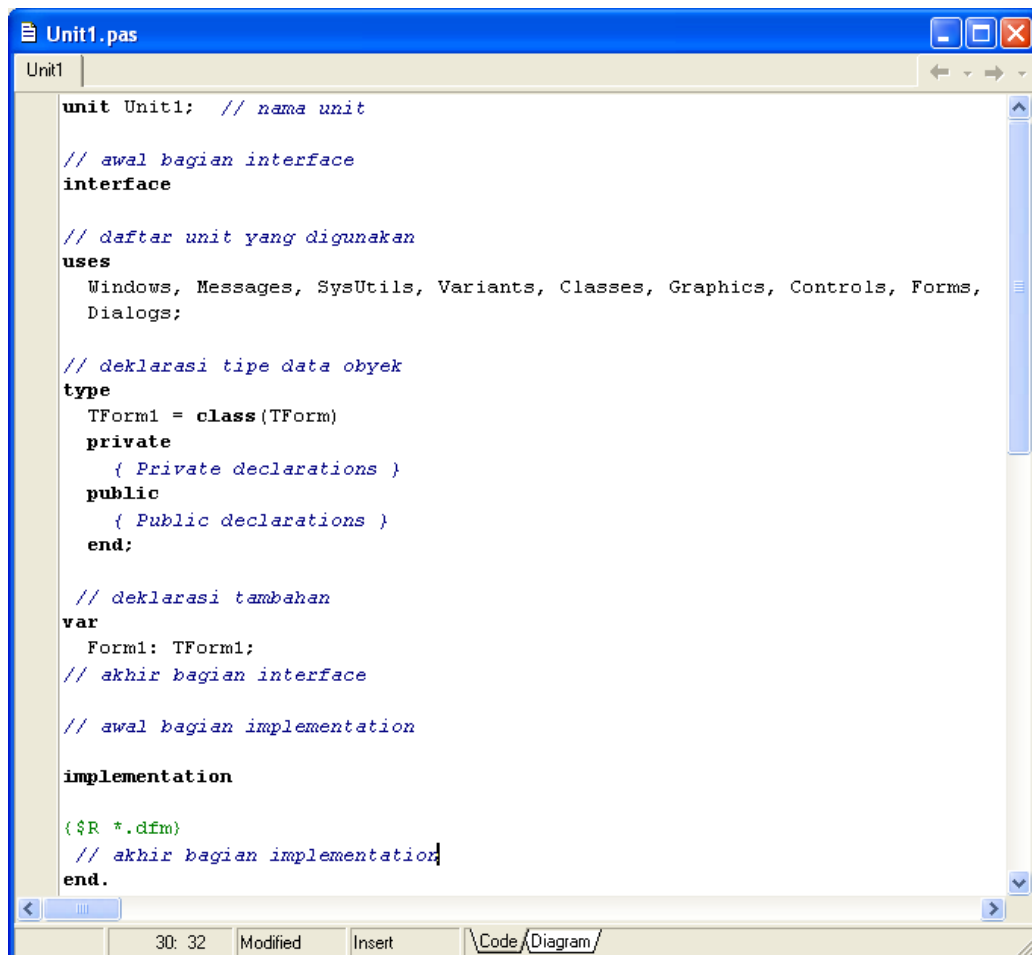
1. Memahami struktur unit dan cara memanggil unit
2. Membuat prosedur dan fungsi

Maksud dan Tujuan :

1. Setelah bab ini berakhir diharapkan mahasiswa dapat memahami struktur unit dan metode pemanggilan unit.
2. Diharapkan mahasiswa dapat membuat prosedur , fungsi dan menerapkan pada aplikasi project yang dibuat.

I. Struktur unit dan cara pemanggilan

Unit merupakan modul yang terpisah yang dapat dikompilasi tersendiri, yang berisi deklarasi bagian publik (antarmuka) dan bagian privat (implementasi). Setiap form dalam delphi mempunyai satu unit yang bersesuaian. Source code unit tersimpan dengan ekstensi pas, sedangkan yang sudah terkompilasi tersimpan dengan ekstensi .dcu. proses link akan menggabung file .dcu menjadi satu file exe atau dll.



Gambar 4. 18. File unit aplikasi delphi

File unit terdiri dari beberapa bagian :

1. Deklarasi nama Unit yaitu Unit Unit1;
2. Interface. Bagian ini memiliki sub bagian :

- a. Uses : mendaftarkan semua class yang diperlukan untuk membangun aplikasi sesuai dengan komponen – komponen yang digunakan didalam aplikasi.
 - b. Type : mengandung beberapa tipe data yang digunakan didalam unit. Terdiri dari bagian Private (hanya diakses diunit ini) dan Public (dapat diakses di unit lain).
 - c. Var : berisi deklarasi variable yang digunakan didalam unit.
3. Implementation. Kode yang ditulis dibawah kata kunci Implementation merupakan tugas – tugas rinci yang ditangani oleh user ini.

II. Membuat procedure dan function

Function (fungsi) dan procedure adalah suatu rutin yang biasanya dipakai sebagai alat untuk melakukan tugas tertentu dan atau mendapatkan nilai tertentu. Procedure dan function juga telah disediakan oleh delphi secara otomatis tinggal digunakan.

Sebagai contoh :

- **Str** : procedure yang digunakan untuk mendapatkan nilai string dari data numeric.
Contoh :

```

procedure TForm1.FormCreate(Sender : TObject);
Var
    I : Single;
    S : String;
Begin
    I := 1500;
    Str(I:8:2, S);
    Form.caption := S;
End;

```

S bernilai 1500, jadi properties caption form akan tertulis 1500.

- **StrToCurr** : function yang dipakai untuk mendapatkan nilai berupa data numeric bertipe currency dari suatu data string. Sebagai contoh :
Contoh :

```

procedure TForm1.FormCreate(Sender : TObject);
Var
    I : Currency;
    S : String;
Begin
    S := '1500';
    I := StrToCurr(S);
End;

```

I bernilai 1500, bertipe currency.

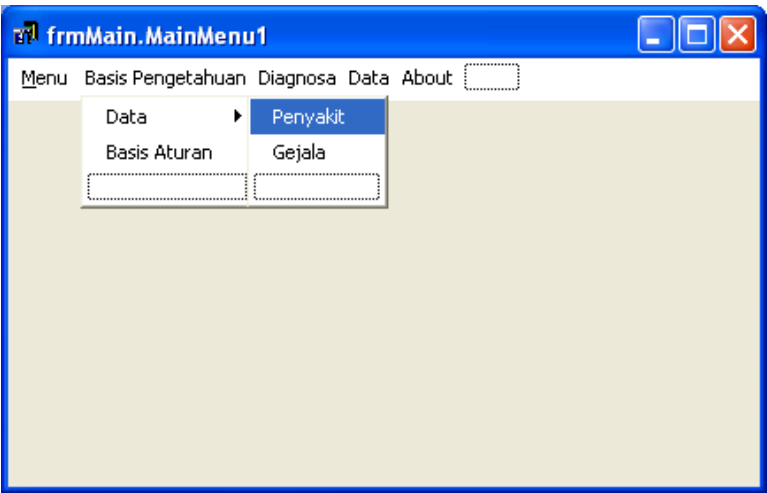
Latihan membuat procedure.

1. Bukalah project yang telah anda simpan.
2. Buat Form baru, klik File > new form
Atur properties form pada object inspector sebagai berikut :

Properti	Pengaturan
Caption	Data Penyakit
Nama	frmPenyakit
Position	Desktop Center

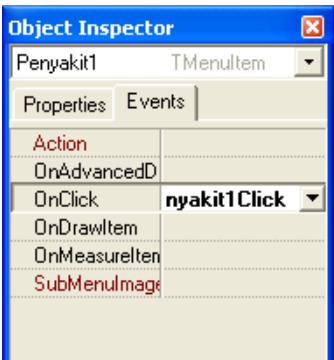
3. Selanjutnya simpanlah dalam 1 folder dimana project anda disimpan dengan nama : uPenyakit.(menyimpan sebuah unit)
4. Selanjutnya, kita akan membuat sebuah procedure yang digunakan untuk memanggil form ini dari form utama. Double klik komponen MainMenu1

pada form utama (frmMain), dan klik menu Basis pengetahuan > Data > Penyakit. Seperti pada gambar dibawah ini :



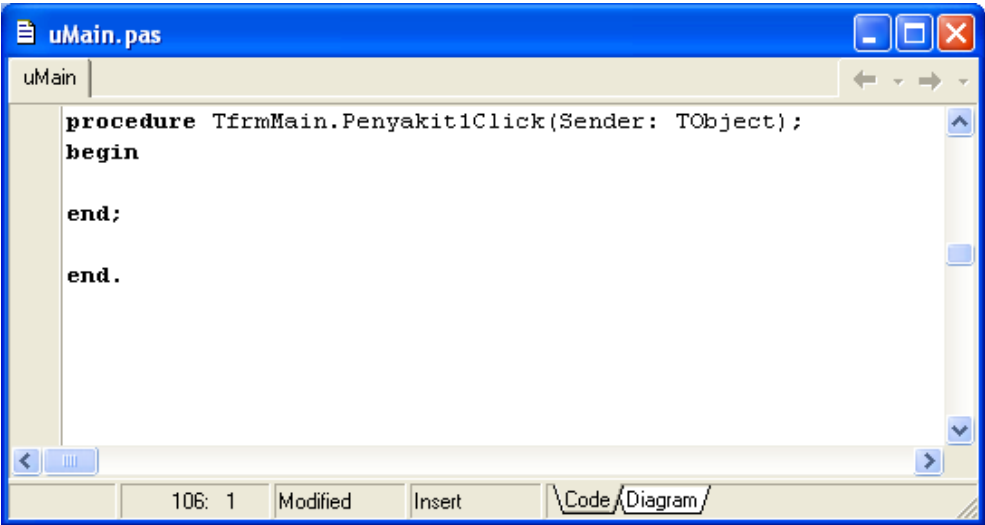
Gambar 4. 19. Main menu pada form utama

5. Selanjutnya, Klik tab event pada object inspector dan double klik pada event OnClick, seperti pada gambar dibawah ini :



Gambar 4. 20. Event OnClick pada Object Inspector

6. Sehingga tampil pada code editor seperti gambar dibawah ini :



Gambar 4. 21. Code editor pada event OnClick

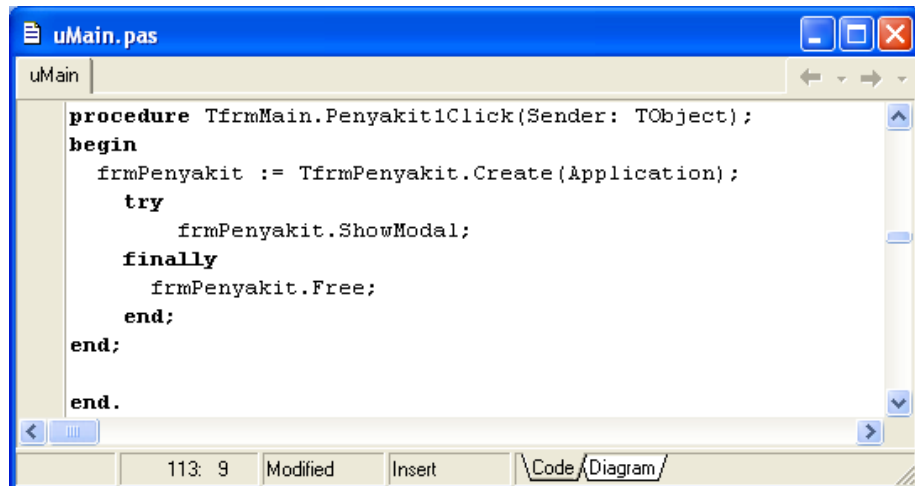
7. Selanjutnya kita akan membuat procedure pemanggilan form frmPenyakit yang telah dibuat, dengan menyisipkan kode dibawah ini, antara begin dan end; sebagai berikut :

```

frmPenyakit := TfrmPenyakit.Create(Application);
try
    frmPenyakit.ShowModal;
finally
    frmPenyakit.Free;
end;

```

Sehingga seperti gambar dibawah ini :



Gambar 4. 22. Procedure untuk menampilkan form

8. Kemudian jalankan aplikasi anda, dengan menekan F9, cobalah klik Basis pengetahuan > Data > Penyakit. Apabila berhasil, maka form penyakit anda akan tampil.

Latihan :

Cobalah buat procedure yang digunakan untuk menutup aplikasi. Dengan menggunakan perintah.

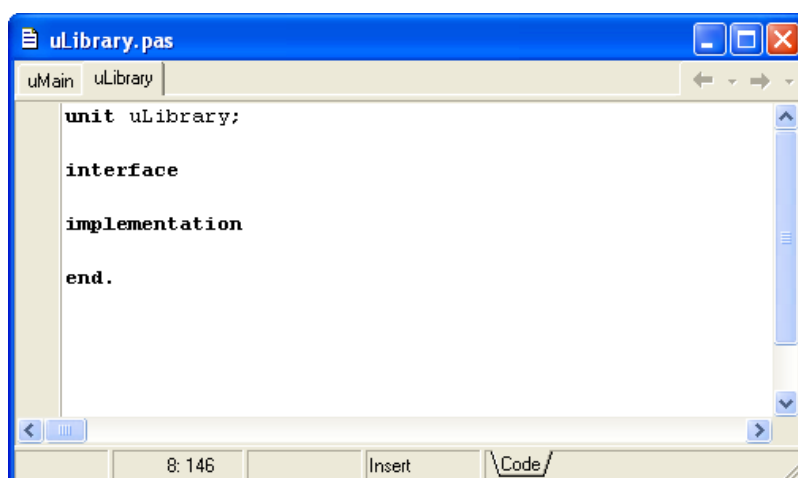
```

Application.terminate;

```

Latihan membuat function.

1. Bukalah project yang telah anda simpan.
2. Buat Form baru, klik File > new unit, kemudian simpan unit tersebut dengan nama uLibrary didalam satu folder dimana project anda disimpan.



Gambar 4. 23. unit baru

3. Dibawah kata kunci **interface**, tambahkan kode dibawah ini :

```
uses Messages, Graphics, Menus, Forms, SysUtils, Variants,
Classes, Controls, Dialogs, commctrl, StdCtrls, ExtCtrls, ComCtrls,
Dbtables, CheckLst, DB, windows ;
```

Kode diatas digunakan untuk memanggil class – class dari library bawaan delphi yang terkait dengan function dan procedure yang akan kita kembangkan.

4. Dibawah kode **uses**, kita akan membuat sebuah function dialog dengan cara menulis pendeklarasian function sebagai berikut :

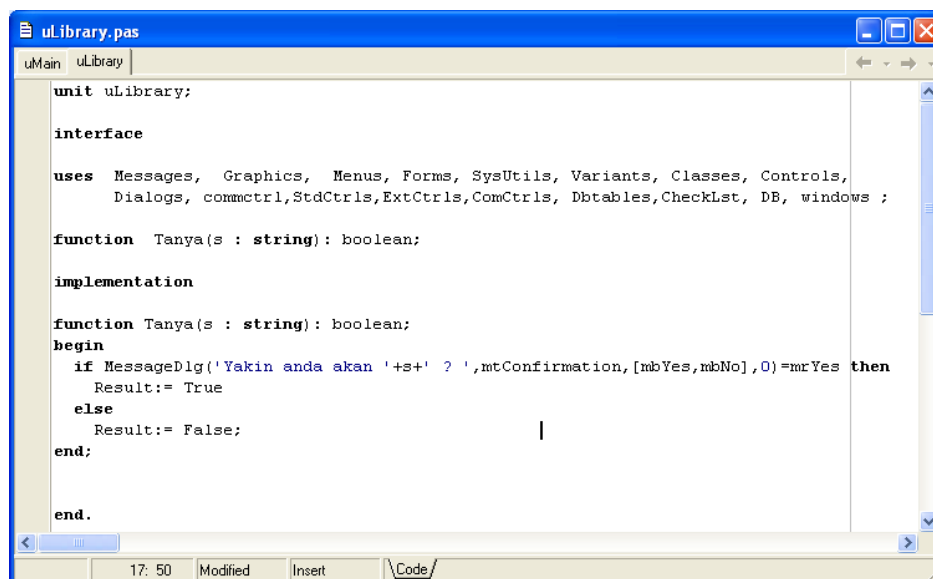
```
function Tanya(s : string): boolean;
```

Function diatas diberi nama “Tanya” dengan parameter bertipe string (huruf atau karakter) dan memiliki nilai balik bertipe Boolean.

5. Kemudian penjabaran dari function yang telah dideklarasikan, ada dibawah kata kunci **implementation**.

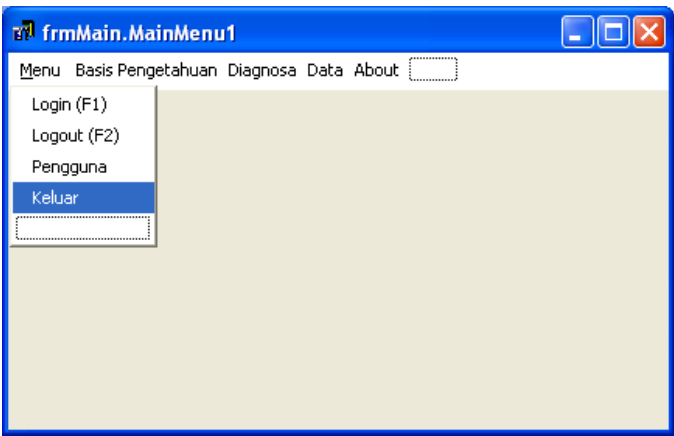
```
function Tanya(s : string): boolean;
begin
    if MessageDlg('Yakin anda akan '+s+' ?
', mtConfirmation, [mbYes, mbNo], 0) = mrYes then
        Result:= True
    else
        Result:= False;
end;
```

Selengkapnya seperti pada gambar dibawah ini :



Gambar 4. 24. Function Tanya pada unit library

6. Pemanggilan function tersebut, double-click komponen MainMenu1 pada form utama, pilih menu klik keluar seperti pada gambar dibawah ini :

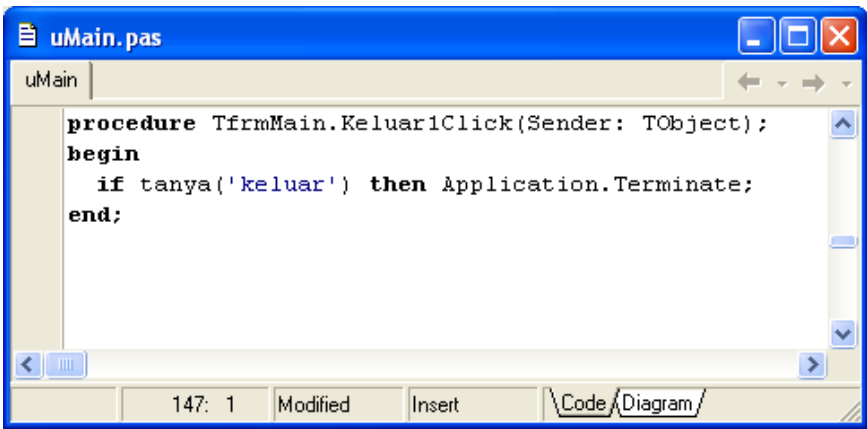


Gambar 4. 25. Main menu pada form utama

7. Pilih Onclick tab event diobject inspector, sisipkan kode dibawah ini diantara begin dan end;

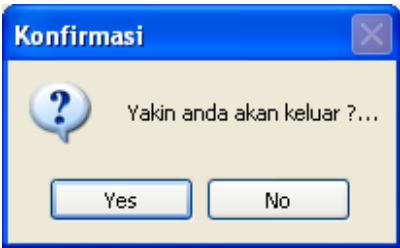
```
if tanya('keluar') then Application.Terminate;
```

Seperti pada gambar dibawah ini :



Gambar 4. 26. Methode pemanggilan function tanya

8. Jalankan aplikasi dengan menekan F9 dan cobalah klik keluar pada form menu anda. Jika berhasil, maka akan muncul konfirmasi dialog untuk menutup aplikasi.



Gambar 4. 27. Konfirmasi dialog menutup aplikasi

BAB V

Materi :

- 1. Studi kasus membuat pengolahan data penyakit.

Maksud dan Tujuan :

- 1. Setelah bab ini berakhir diharapkan mahasiswa dapat membuat validasi form.
- 2. Diharapkan mahasiswa melakukan pengolahan data (insert, update dan delete) dan menampilkan (select).

Dari hasil teori dan praktek dari bab sebelumnya, sekarang kita akan mencoba mengimplementasikan pada sebuah kasus pengolahan data penyakit. Adapun langkah – langkahnya sebagai berikut :

- 1. Buat table *penyakit* pada database pakar_db dengan menggunakan sql tool atau console. Struktur table sebagai berikut :

No.	Field	Type	Length
1.	penyakitId	Tinyint	3
2.	nm_penyakit	Varchar	255
3.	penyebab	Varchar	
4.	diagnosa	Varchar	
5.	pengobatan	Varchar	
6.	pencegahan	varchar	

- 2. Desain form seperti pada gambar dibawah ini :

Gambar 5. 28. Tampilan form penyakit

Daftar komponen yang terdapat pada form penyakit sebagai berikut :

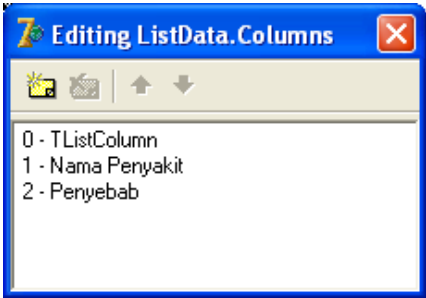
No.	Nama Komponen	Pengaturan Properti di Object Inspector
1.	Form1	Name : frmPenyakit Position : poDesktopCenter KeyPreview : true
2.	Panel1	Caption : Penyakit Align : altop
3.	Edit1	Name : edPenyakit

4.	Memo1 Memo2 Memo3 Memo4	Name : edPenyebab Name : edDiagnosa Name : edPengobatan Name : edPencegahan
5.	ListView1	Name : ListData RowSelect : True ViewStyle : vsReport Align : alBottom
6.	GroupBox1	Align : alBottom
7.	Button1 Button2 Button3 Button4	Name : BtnOperasi, Caption : &Simpan Name : BtnBatal , Caption : &Batal Name : BtnHapus, Caption : &Hapus Name : BtnTutup, Caption : &Tutup

Klik kanan pada komponen listview dan pilih colom editor, kemudian klik Addnew list sampai 3 kali.

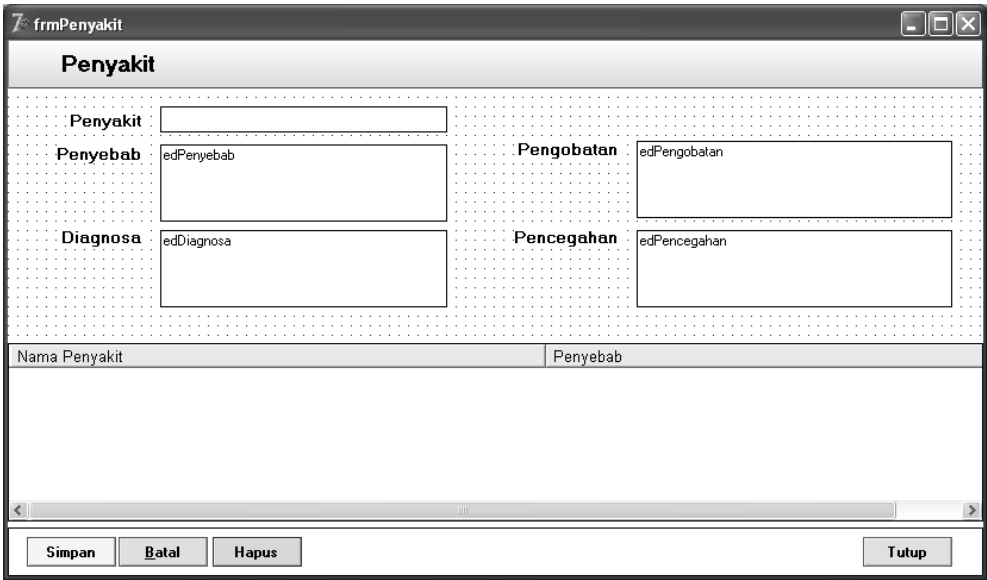
Atur property listindex pada object inspector sebagai berikut :

- Listindex ke -0 : width : 1
- Listindex ke -1 : width : 450, caption : Nama Penyakit.
- Listindex ke -2 : width : 430, caption : Penyebab.



Gambar 5. 29. Editing kolom pada ListData

Hasil akhir list view yang telah anda atur menjadi seperti gambar dibawah ini :



Gambar 5. 30. Hasil akhir listview pada form penyakit

3. Buat DataModule, Pilih File > New > Data module. Ubah property nama menjadi : DM pada object inspector dan simpan pada folder project anda dengan nama uDM.
4. Kemudian dari tab component palette zeos, isikan 1 komponen ZConnection1 dan 3 ZQuery kedalam data module yang baru saja anda buat.



Gambar 5. 31. Komponen Zeos pada data module

5. Kemudian atur property pada object inspector sebagai berikut.

Komponen	Pengaturan
ZConnection	Name : Koneksi Database : pakar_db Port : 3306 Protocol : mysql User : root Connected : true
ZQuery1	Connection : Koneksi Name : Q1
ZQuery2	Connection : Koneksi Name : Q2
ZQuery3	Connection : Koneksi Name : Q3

6. Setelah semua pengaturan selesai, kemudian simpan semua perubahan dengan cara klik file > saveall.

Memulai proses koding.

1. Buatlah procedure dengan nama *tampil* dibawah *{ \$R *.dfm }* dengan code sebagai berikut :

```

procedure TfrmPenyakit.Tampil;
begin
    CommandSQL(DM.Q2,'SELECT * FROM penyakit ORDER BY
    penyakitID DESC ',True);
    ListData.Items.Clear;
    ListData.Items.BeginUpdate;
    with DM.Q2 do
    begin
        First;
        while not eof do
        begin
            with ListData.Items.Add do
            begin
                Caption:= Fields[0].AsString;
                SubItems.Add(Fields[1].AsString);
                SubItems.Add(Fields[2].AsString);
            end;
            Next;
        end;
    end;

```



```

end;
ListData.Items.EndUpdate;
end;

```

2. Pada event form OnShow diobject inspector, tuliskan kode berikut ini :

```

procedure Tfrmpenyakit.FormShow(Sender: TObject);
var i : Shortint;
begin
    Caption:= judul +' - Data Penyakit';
    edpenyakit.Clear;
    for i:=0 to ComponentCount-1 do
        begin
            if Components[i] is TMemo then
                TMemo(Components[i]).Clear;
            end;
            edpenyakit.SetFocus;
            BtnHapus.Enabled:= False;
            stsEdit          := False;
            id:='';
            BtnOperasi.Caption :='Simpan';
            Tampil;
        end;
    end;

```

3. Pada event form OnKeyDown, sisipkan kode berikut ini :

```

if Key=VK_RETURN then
    Perform(WM_NEXTDLGCTL, 0, 0);

```

4. Buat procedure untuk validasi penyimpanan dengan nama *cekKosong* :

```

procedure TfrmPenyakit.cekKosong;
begin
    CekNullEdit(edPenyakit,'nama penyakit');
    cekNullMemo(edPenyebab,'penyebab');
    cekNullMemo(edDiagnosa,'diagnosa');
    cekNullMemo(edPengobatan,'pengobatan');
    cekNullMemo(edPencegahan,'pencegahan');
end;

```

5. Pada event OnClick btnOperasi tuliskan kode berikut ini :

```

procedure Tfrmpenyakit.BtnOperasiClick(Sender: TObject);
begin
    cekKosong;
    if Tanya('menyimpan') then
        begin
            if stsEdit then
                _SQL:= 'UPDATE penyakit SET
nm_penyakit="'+edPenyakit+"",
diagnosa="'+edDiagnosa.Text+"", '+'
                'pengobatan="'+edPengobatan.Text+"",
pencegahan="'+edPencegahan.Text+" " '+'
                ' WHERE penyakitId="'+id+" " '
            else
                _SQL:= 'INSERT INTO penyakit(nm_penyakit,
diagnosa, pengobatan, pencegahan) VALUES '+'
                ' ("'+edPenyakit.Text+"", "'"+edDiagnosa.Text+
                '"', "'"+edPengobatan.Text+"", "'"+edPencegahan+"") '";
                CommandSQL(DM.Q1,_SQL,False);
                FormShow(Self);
                Berhasil('Penyimpanan sukses');
            end;
        end;
    end;

```

6. Pada tombol batal, isikan kode berikut pada event OnClick di codeinspector :

```
FormShow(Self);
```

7. Klik pada ListData, kemudian pilih tab event pada object inspector pada event OnDoubleClick, sisipkan kode berikut ini :

```
procedure Tfrmpenyakit.ListDataDbClick(Sender:
TObject);
begin
    if ListData.ItemIndex < 0 then exit;
    Id := ListData.Selected.Caption;
    CommandSQL(DM.Q3,'select * from penyakit where
penyakitID="'+id+'"',True);
    with DM.Q3 do
    begin
        edpenyakit.Text := ListData.Selected.SubItems[0];
        edPenyebab.Text := Fields[2].AsString;
        edDiagnosa.Text := Fields[3].AsString;
        edPengobatan.Text:= Fields[4].AsString;
        edPencegahan.Text:= Fields[5].AsString;
    end;

    stsEdit := True;
    BtnOperasi.Caption := 'Ubah';
    BtnHapus.Enabled := True;
end;
```

8. Kemudian pada komponen edPenyakit, properti event OnChange isikan kode berikut ini :

```
procedure Tfrmpenyakit.edpenyakitChange(Sender:
TObject);
begin
    if Trim(edpenyakit.Text)='' then
    BtnBatal.Enabled := False else
        BtnBatal.Enabled := True;
end;
```

9. Event OnClick pada BtnBatal, isikan kode berikut ini :

```
procedure Tfrmpenyakit.btnBatalClick(Sender: TObject);
begin
    FormShow(Self);
end;
```

10. Kemudian pada btnHapus, isikan kode berikut ini :

```
procedure Tfrmpenyakit.BtnHapusClick(Sender: TObject);
begin
    if Tanya('menghapus penyakit '+edpenyakit.Text) then
        CommandSQL(DM.Q1,'DELETE FROM penyakit WHERE
penyakitId="'+id+'"',False);
        Berhasil('Penghapusan sukses');
        FormShow(Self);
end;
```

Dan pada event OnClick tombol BtnClose, isikan kode :

```
Close;
```

Jalankan program dengan menekan F9, dan cobalah memasukan data, mengedit dan mendelete untuk testing aplikasi yang telah anda buat pada module data penyakit.

BAB VI

Materi :

- 1. Studi kasus membuat pengolahan data gejala penyakit.

Maksud dan Tujuan :

- 1. Melanjutkan dari bab sebelumnya, dan diharapkan setelah bab ini berakhir, mahasiswa semakin mudah dan lancar dalam pengolahan data.

Lanjutkan ke modul pengolahan data gejala, sesuai dengan referensi sesuai dengan bab sebelumnya.

- 1. Buat table *gejala* pada database pakar_db dengan menggunakan sql tool atau console. Struktur table sebagai berikut :

No.	Field	Type	Length
1.	gejalald	Tinyint	3
2.	nm_gejala	varchar	255

- 2. Desain form seperti pada gambar dibawah ini :

The image shows a screenshot of a Windows application window titled "frmGejala". The window has a standard Windows title bar with minimize, maximize, and close buttons. Below the title bar is a header area with the word "Gejala" in bold. The main area of the form is divided into two sections. The top section has a label "Gejala:" followed by a single-line text input box. The bottom section has a label "Nama Gejala" followed by a multi-line list box. At the bottom of the form, there are four buttons: "Simpan", "Batal", "Hapus", and "Tutup".

Gambar 6.32. Tampilan desain form gejala

BAB VII

Materi :

- 1. Studi kasus membuat pengolahan data basis aturan

Maksud dan Tujuan :

- 1. Melanjutkan dari bab sebelumnya, dan diharapkan setelah bab ini berakhir, mahasiswa semakin mudah dan lancar dalam pengolahan data.

Lanjutkan ke modul pengolahan data basis aturan, sesuai dengan referensi sesuai dengan bab sebelumnya.

- 1. Buat table *basisaturan* pada database pakar_db dengan menggunakan sql tool atau console. Struktur table sebagai berikut :

No.	Field	Type	Length
1.	penyakitId	Tinyint	3
2.	gejalaId	Tinyint	3

- 2. Desain form seperti pada gambar dibawah ini :

The image shows a screenshot of a Windows application window titled "frmAturan". The window has a standard Windows title bar with minimize, maximize, and close buttons. Below the title bar is a header area with the text "Basis Aturan". The main content area contains a "Penyakit" label followed by a dropdown menu. Below that is a "List Gejala" label followed by a large empty list box. At the bottom left of the main area is a checkbox labeled "Select All". At the bottom of the window are two buttons: "Simpan" on the left and "Tutup" on the right.

Gambar 7.33. Tampilan desain form basis aturan

- 3. Buatlah procedure tampil pada uAturan dengan kode berikut ini :

```
procedure TfrmAturan.Tampil;
begin
  CommandSQL(dm.Q1,'select * from gejala',True);
  With DM.Q1 do
  begin
    ListGejala.Items.Clear;
```

```

slGejala := TStringList.Create;
slGejala.Clear;
First;
while not Eof do
begin
    ListGejala.Items.Add(Fields[1].AsString);
    slGejala.Add(Fields[0].AsString);
    Next;
end;
end;
end;

```

4. Klik form, kemudian pilih event OnFormShow pada object inspector, tuliskan kode berikut ini :

```

procedure TfrmAturan.FormShow(Sender: TObject);
begin
    Tampil;
    cmbPenyakit.Clear;
    TampilKeCombo(cmbPenyakit,DM.Q1,'select * from
penyakit');
    Caption:= judul +' - Data Basis Aturan';
end;

```

5. Kemudian pada combo penyakit, pilih event onchange, ketik kode berikut ini :

```

procedure TfrmAturan.cmbPenyakitChange(Sender: TObject);
var
    _i : Integer;
begin
    if cmbPenyakit.ItemIndex=-1 then
    begin
        Tampil;
        Exit;
    end;

    penyakitID:= FetchCodeCombo(cmbPenyakit.Text,1);
    Tampil;
    CommandSQL(DM.Q2,'select * from aturan where
penyakitID="'+penyakitID+'"',True);
    DM.Q2.First;
    while not (DM.Q2.Eof) do
    begin
        for _i:=0 to ListGejala.Items.Count-1 do
        begin
            //ShowMessage(DM.Q2.Fields[1].AsString+' pod
'+slGejala[_i]);
            if (DM.Q2.Fields[1].AsString=slGejala[_i]) then
ListGejala.Checked[_i]:= true;
            end;
            DM.Q2.Next;
        end;
    end;
end;

```

6. Kemudian pada event OnClick BtnSimpan tulis kode berikut :

```

procedure TfrmAturan.BtnOperasiClick(Sender: TObject);
var
    jum,i : integer;
begin
    jum:= ListGejala.Items.Count-1;
    CekStrings(cmbPenyakit,'Penyakit harus dipilih');

```

```

    CekStrings(ListGejala,'Gejala harus dipilih');
    CommandSQL(DM.Q1,'delete from aturan where
penyakitID="'+penyakitID+'"',False);
    for i:= 0 to jum do
    begin
        if ListGejala.Checked[i] then
            CommandSQL(DM.Q2,'insert into aturan
values ("'+penyakitID+'","'+slGejala[i]+'") ',False);
        end;
        Berhasil('Penyimpanan sukses');
        FormShow(Self);
    end;
end;

```

BAB VIII

Materi :

- 1. Studi kasus membuat system pelacakan penyakit

Maksud dan Tujuan :

- 1. Melanjutkan dari bab sebelumnya, dan diharapkan setelah bab ini berakhir, mahasiswa dapat bereksplorasi dengan teknik logika query sql untuk melacak suatu penyakit pada penerapan ilmu kecerdasan buatan.

Ikuti langkah – langkah berikut ini :

- 1. Buat table *tpenyakit* dan *taturan* pada database pakar_db. Struktur table sebagai berikut :
Struktur table tpenyakit :

No.	Field	Type	Length
1.	penyakitId	Tinyint	3

Struktur table taturan :

No.	Field	Type	Length
1.	gejalaId	Tinyint	3

- 2. Desainlah tampilan form seperti pada gambar dibawah ini :

frmKonsult

Diagnosa Penyakit

Pilih gejala bayi anda :

Hasil Diagnosa Penyakit :

LABELPENYAKIT

Reset

Detail

Tutup

Gambar 8.33. Tampilan desain form basis aturan

Daftar komponen yang terdapat pada form konsultasi sebagai berikut :

No.	Nama Komponen	Pengaturan Properti di Object Inspector
1.	Form1	Name : frmKonsult Position : poDesktopCenter KeyPreview : true
2.	Panel1	Caption : Diagnosa Penyakit Align : altop
3.	Label1	Caption : Pilih gejala bayi anda :
4.	Timer1	Name : timer1
5.	ListView1	Name : ListData RowSelect : True ViewStyle : vsReport Align : alNone
6.	GroupBox1	Align : alBottom
7.	Button1 Button2 Button3	Name : BtnOperasi, Caption : &Reset Name : BtnDetail , Caption : &Detail Name : BtnTutup, Caption : &Tutup

3. Tambahkan pada unit Library, sebuah function dibawah ini yang digunakan untuk mengecek keberadaan data dalam sebuah table :

```
function CekData(Query:TQuery; nmTabel:String):boolean;
begin
    CommandSQL(Query,'select * from '+nmTabel,True);
    if Query.IsEmpty then
        result := true
    else
        result := false;
end;
```

4. Tambahkan sebuah procedure pada unit library yang berfungsi untuk mengosongkan table sebagai berikut :

```
procedure KosongTabel(const Query : TQuery ; nmTabel :
String);
begin
    CommandSQL(Query,'TRUNCATE '+nmTabel,False);
end;
```

5. Kemudian pada code editor, buat procedure ViewGejala dengan sintak sebagai berikut :

```
procedure TfrmKonsult.ViewGejala;
begin
    if CekData(DM.Q1,' taturan ') then
        _SQL:='SELECT * FROM gejala ORDER BY gejalaNama
ASC'
    else
        _SQL:='SELECT DISTINCT gp.gejalaID, g.gejalaNama
FROM gejala AS g, tpenyakit AS t, aturan AS gp '+'
        'WHERE g.gejalaID=gp.gejalaID AND
gp.penyakitID=t.penyakitID AND gp.gejalaID NOT IN
(SELECT gejalaID FROM taturan) ORDER BY g.gejalaNama ASC
';
    CommandSQL(DM.Q2,_SQL,True);
    ListData.Items.Clear;
    ListData.Items.BeginUpdate;
```



```

with DM.Q2 do
begin
    First;
    while not eof do
    begin
        with ListData.Items.Add do
        begin
            Caption:= Fields[0].AsString;
            SubItems.Add(Fields[1].AsString);
        end;
        Next;
    end;
end;
ListData.Items.EndUpdate;
if DM.Q2.RecordCount=0 then
begin
    CommandSQL(DM.Q4,'select distinct t.penyakitID,
p.penyakitNama from tpenyakit as t, penyakit as p where
t.penyakitID=p.penyakitID',True);
    Timer1.Enabled      := True;
    Label3.Visible      := True;
    labelPenyakit.Visible := true;
    labelPenyakit.Caption :=
DM.Q4.FieldName('penyakitNama').AsString;
    penyakitID:=
DM.Q4.FieldName('penyakitID').AsString;
    BtnOperasi.Enabled  := True;
end;

    CommandSQL(DM.Q5,'SELECT t.penyakitID, p.penyakitNama
FROM penyakit as p, tpenyakit as t where
p.penyakitID=t.penyakitID ORDER BY t.penyakitID DESC
',True);
    ListView1.Items.Clear;
    ListView1.Items.BeginUpdate;
    with DM.Q5 do
    begin
        First;
        while not eof do
        begin
            with ListView1.Items.Add do
            begin
                Caption:= Fields[0].AsString;
                SubItems.Add(Fields[1].AsString);
            end;
            Next;
        end;
    end;
end;
ListView1.Items.EndUpdate;
end;

```

6. Pada event OnFormShow formKonsult, tuliskan kode dibawah ini :

```

procedure TfrmKonsult.FormShow(Sender: TObject);
begin
    KosongTabel(DM.Q1, 'tpenyakit');
    KosongTabel(DM.Q4, 'taturan');
    ViewGejala;
    Label1.Caption := 'Saudara '+namaPasien+' silahkan
pilih gejala bayi anda ';
    labelPenyakit.Visible:=False;
    Caption:= judul +' - Diagnosa';
    BtnOperasi.Enabled := False;
end;

```

7. Pada event OnDoubleClick pada komponen ListData , isikan kode dibawah ini :

```
procedure TfrmKonsult.ListDataDbClick(Sender: TObject);  
var  
    _sl : string;  
begin  
    if ListData.ItemIndex < 0 then exit;  
    gejalaID:= ListData.Selected.Caption;  
  
    if CekData(DM.Q5,' taturan ') then  
        _sl:='SELECT penyakitID FROM aturan WHERE  
gejalaID="'+gejalaID+'"  
    else  
        _sl:='SELECT gp.penyeakitID FROM aturan AS gp,  
tpenyakit AS t WHERE gp.penyeakitID=t.penyeakitID AND  
gp.gejalaID="'+gejalaID+'";  
  
    CommandSQL(DM.Q5,_sl,True);  
    KosongTabel(DM.Q3,'tpenyakit');  
    with DM.Q5 do  
    begin  
        First;  
        while not eof do  
        begin  
            CommandSQL(DM.Q4,'insert into tpenyakit  
values ("'+FieldByName('penyeakitID').AsString+'" ),False)  
        ;  
            Next;  
        end;  
    end;  
    CommandSQL(DM.Q2,'INSERT INTO taturan(gejalaID)  
VALUES ("'+gejalaID+'") ',False);  
    ListData.Clear;  
    ViewGejala;  
end;
```

8. Pada event onClick tombol btnReset, isikan kode : FormShow(Self);
9. Pada event onclick BtnDetail, digunakan untuk mencetak laporan hasil diagnosa. Cara membuat laporan akan dibahas pada bab selanjutnya.

BAB IX

Materi :

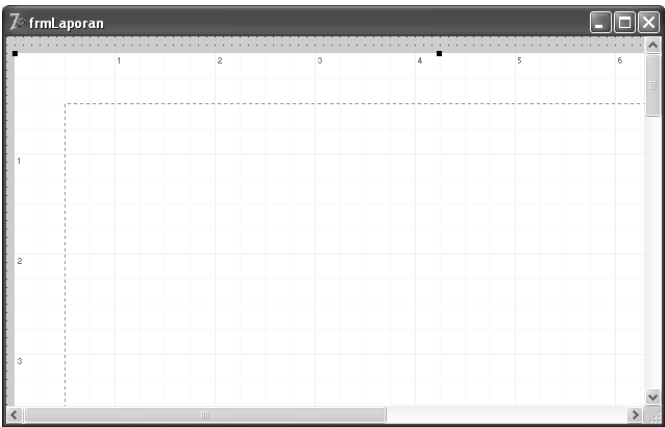
- 1. Studi kasus membuat laporan data penyakit

Maksud dan Tujuan :

- 1. Setelah bab ini berakhir, mahasiswa dapat membuat laporan dengan component Quick Report.

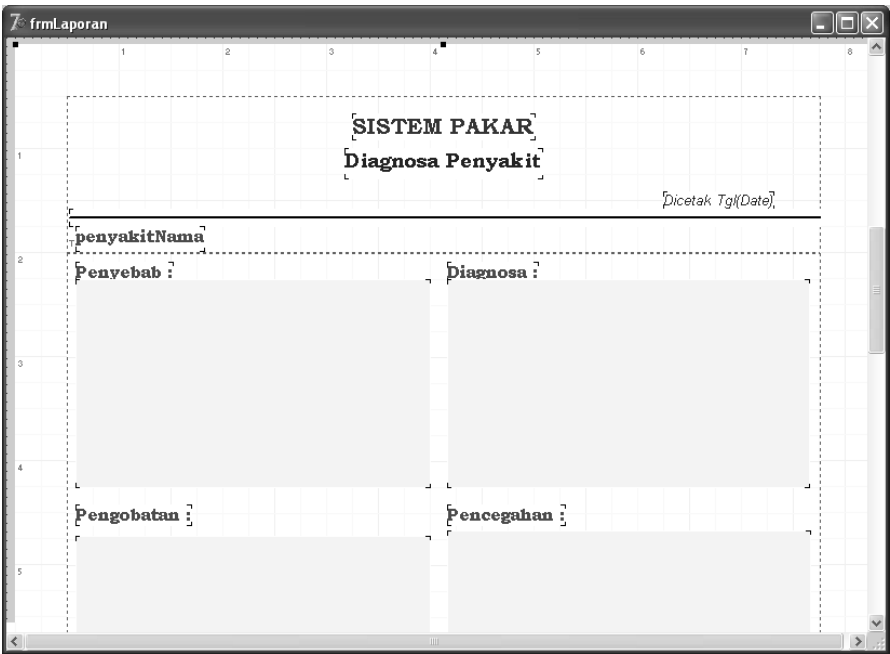
Langkah-langkah membuat laporan adalah sebagai berikut :

- 1. Buat sebuah form yang diberi nama pada frmLaporan.
- 2. Letakkan komponen QuickRep1 pada form, seperti pada gambar dibawah ini.



Gambar 9.34. Tampilan desain form laporan

- 3. Atur properti QuickRep1 pada object inspector, sebagai berikut :
 - DataSet : DM.Q1
 - DetailBand : true
 - PageFooterBand : true
 - TitleBand : true
- 4. Kemudian desain tampilan laporan seperti pada gambar dibawah ini, dengan menggunakan QRLabel yang ada pada tab Qreport component palette :



Gambar 9.35. Tampilan desain form laporan

5. Buka kembali frmKonsult, pada OnClick BtnDetail, isikan kode berikut ini :

```
procedure TfrmKonsult.BtnOperasiClick(Sender: TObject);  
begin  
    CommandSQL(DM.Q2,'insert into  
    hasil_konsultasi(penyakitID,kd_pasien,tgl_konsultasi)  
    values (''+penyakitID+'',''+IDpasien+'',CURDATE()),False  
    );  
    CommandSQL(DM.Q1,'select * from penyakit where  
    penyakitID="'+penyakitID+'"',True);  
    frmLaporan := TfrmLaporan.Create(Application);  
    try  
        frmLaporan.QuickRep2.Preview;  
    finally  
        frmLaporan.Free;  
    end;  
end;
```

Latihan

Buatlah laporan seluruh data penyakit dan laporan seluruh data gejala

BAB X

Materi :

1. Pengaturan otoritas untuk mengakses ke sistem

Maksud dan Tujuan :

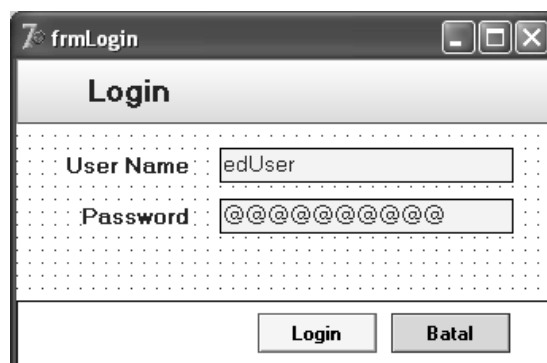
1. Setelah bab ini berakhir, mahasiswa dapat membuat proses login dan logout pada sebuah system aplikasi.

Langkah – langkah sebagai berikut :

1. Buat procedure yang digunakan untuk menonaktifkan menu sebelum proses login. Procedure tersebut beri nama Session dengan parameter bertipe Boolean. Pastikan pada unit utama, sisipkan kode berikut ini :

```
procedure TfrmMain.Session(Login:Boolean);  
begin  
  if Login then  
  begin  
    Login1.Visible      := not login;  
    Logout1.Visible     := login;  
    PengPeny.Visible    := login;  
    BasisAturan1.Visible := login;  
    Pengguna1.Visible   := Login;  
    IdentifikasiPenyakit1.Visible:= not Login;  
    Laporan1.Visible    := Login;  
  end else  
  begin  
    Login1.Visible      := not login;  
    Pengguna1.Visible   := Login;  
    Logout1.Visible     := login;  
    PengPeny.Visible    := login;  
    BasisAturan1.Visible := login;  
    IdentifikasiPenyakit1.Visible:= not Login;  
    Laporan1.Visible    := Login;  
  end;  
end;
```

2. Buatlah form baru dengan nama frmLogin, seperti pada gambar dibawah ini :



Gambar 10.36. Tampilan desain form login

Daftar komponen yang terdapat pada form login sebagai berikut :

No.	Nama Komponen	Pengaturan Properti di Object Inspector
1.	Form1	Name : frmLogin Position : poDesktopCenter KeyPreview : true
2.	Panel1	Caption : Login Align : altop
3.	Label1	Caption : User Name :
4.	Label2	Caption : Password :
5.	Edit1 Edit2	Name : edUser Name : edPassword dan PasswordChar : *
6.	Button1 Button2	Name : BtnLogin, Caption : &Login Name : BtnBatal , Caption : &Batal

3. Pada event OnChange edPassword, isikan kode berikut ini :

```
procedure TfrmLogin.edPasswordChange(Sender: TObject);  
begin  
    CommandSQL(dm.Q1,'select * from pengguna where  
    userName="' + edUser.Text + '" and  
    userPassword="' + Enkripsi(edPassword.Text) + '" ',true);  
    if DM.Q1.IsEmpty then  
        BtnOperasi.Enabled:= False  
    else  
        begin  
            BtnOperasi.Enabled:= true;  
            BtnOperasi.SetFocus;  
        end;  
    end;
```

BAB XI

Materi :

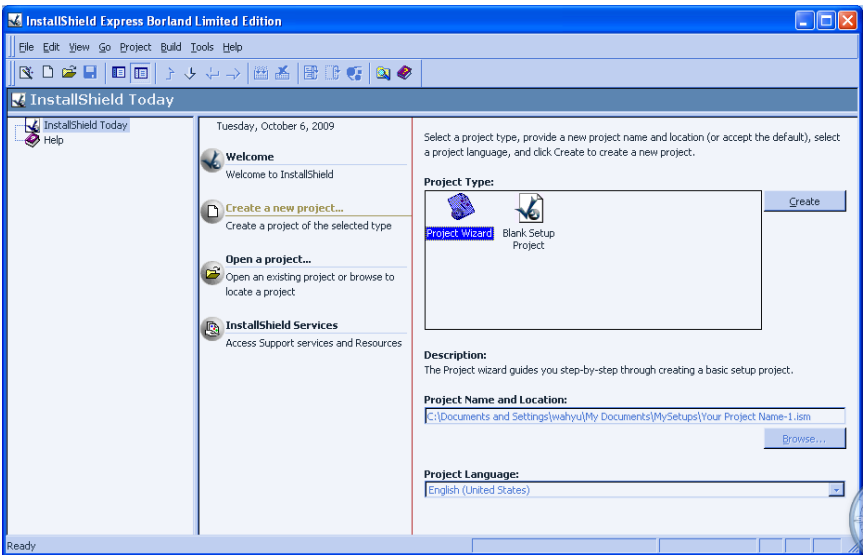
- 1. Membuat file setup

Maksud dan Tujuan :

- 1. Setelah bab ini berakhir, mahasiswa dapat membuat aplikasi yang telah di develop menjadi file setup (file instalasi).

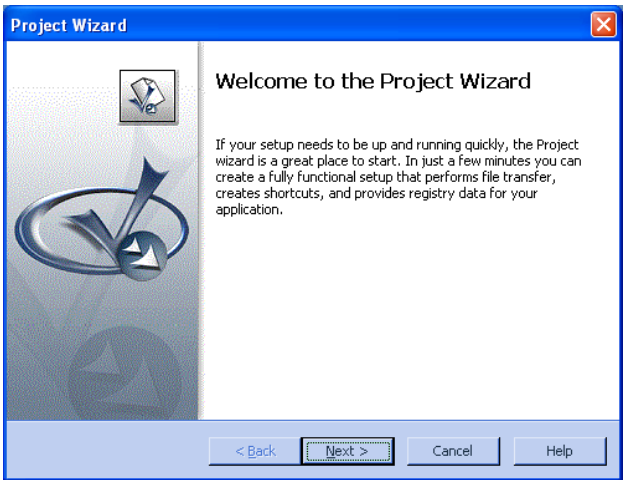
Program aplikasi siap pakai, tidak berbentuk excutable dari hasil kompilasi, namun berupa file setup yang nantinya akan diinstal pada system operasi. Pada bab ini, pembuatan file setup dengan menggunakan aplikasi bawaan delphi dengan nama *InstallShield*.

- 1. Klik *start -> Program -> InstallShield -> Express*. Sehingga muncul tampilan seperti dibawah ini :



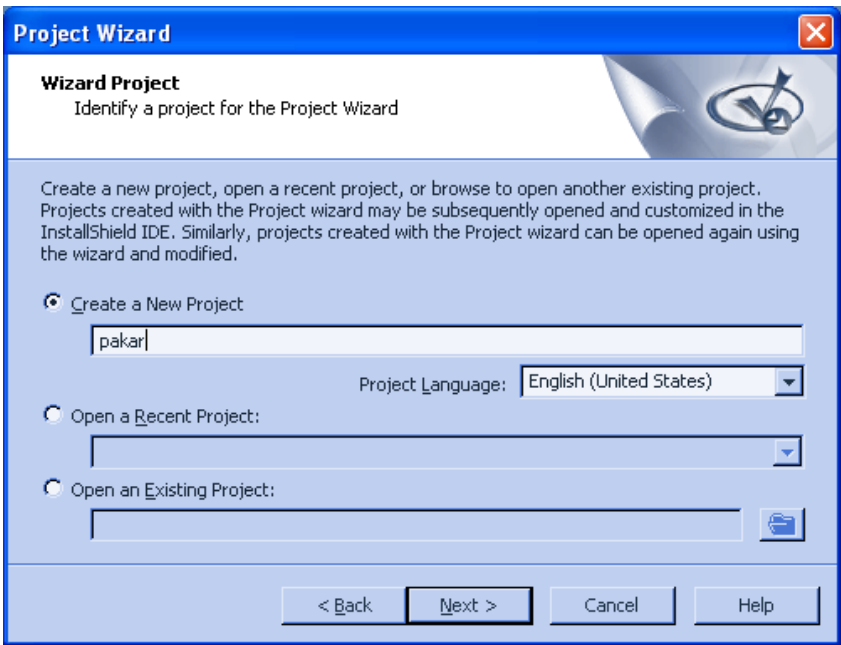
Gambar 11.36. Halaman depan installshield

- 2. Klik project wizard, pada project type, kemudian klik create sehingga muncul tampilan seperti dibawah ini.



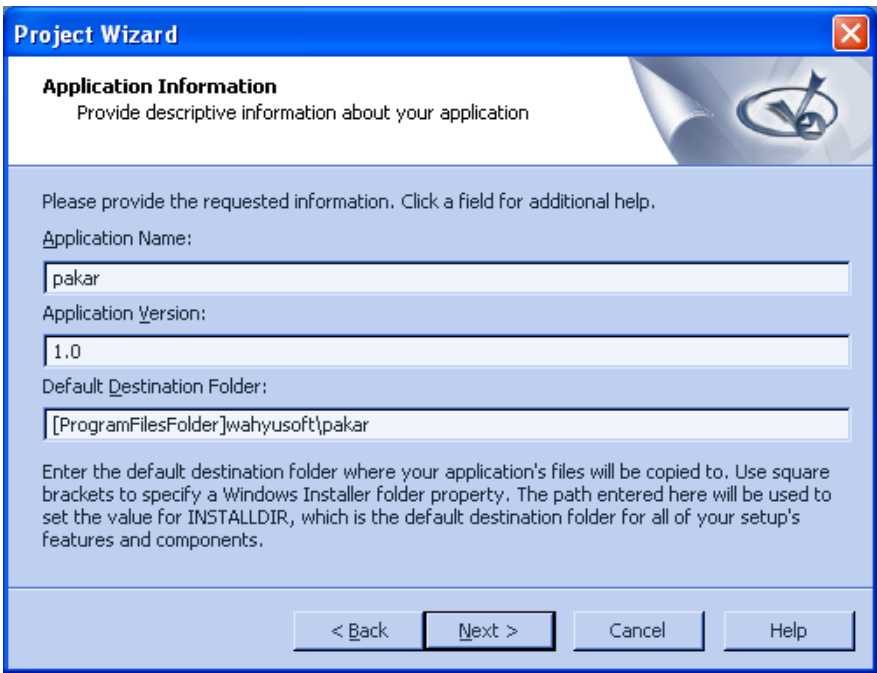
Gambar 11.37. Tampilan selamat datang

3. Klik next, isikan create a new project dengan nama project anda (misal : *pakar*).



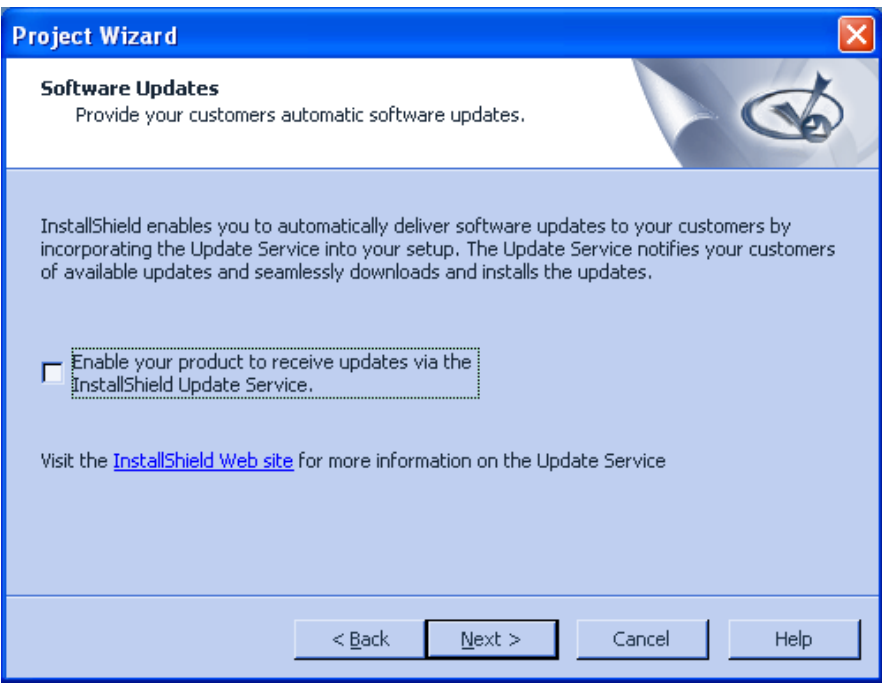
Gambar 11.38 Tampilan create new project

4. Klik next, isikan nama aplikasi, versi, dan default destination seperti pada gambar dibawah ini :



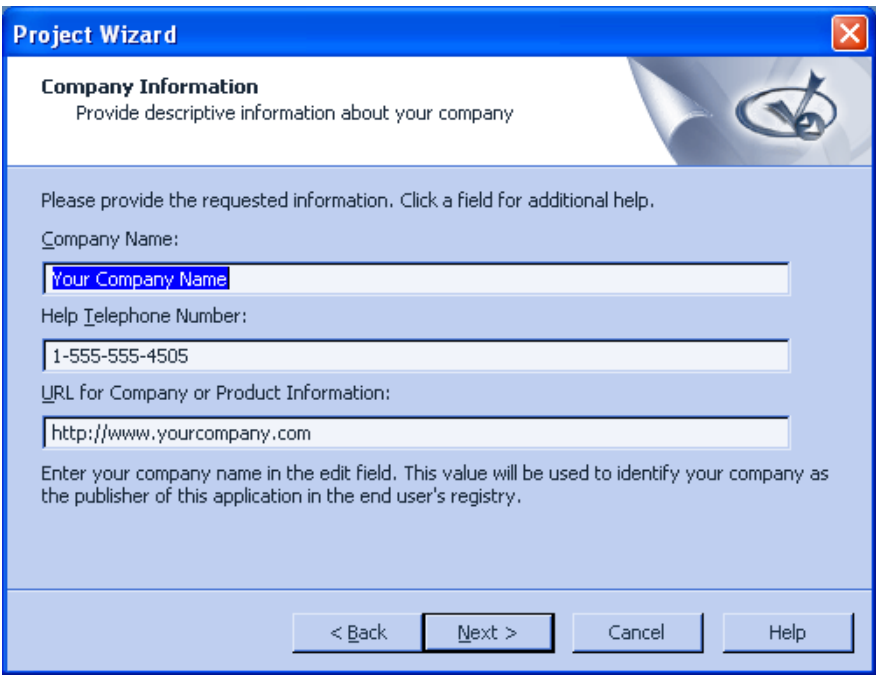
Gambar 11.39. Halaman setup informasi aplikasi anda

5. Klik next, kemudian *unchecked* pada automatic update.



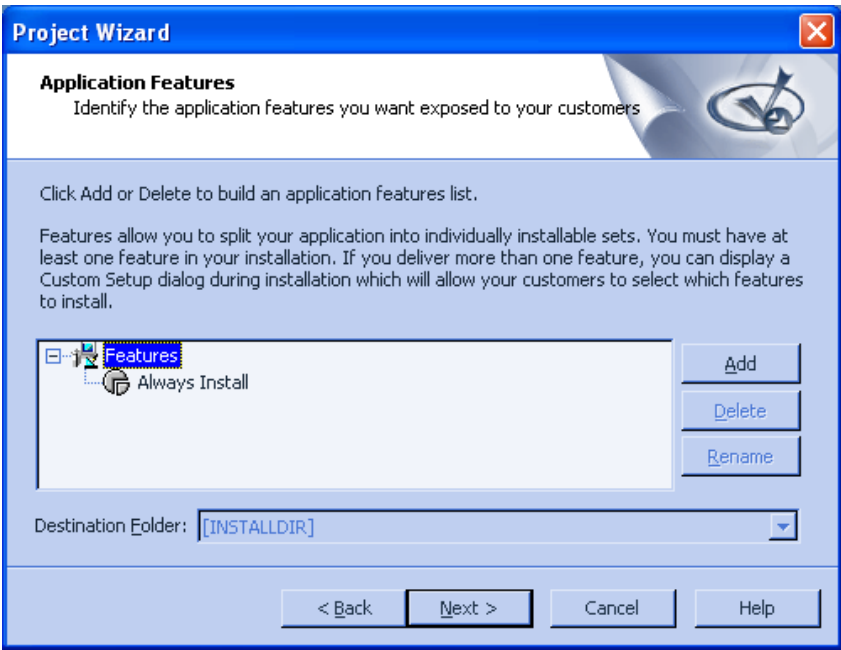
Gambar 11.40. konfirmasi auto update software installshield

6. Klik next, isikan nama perusahaan anda, nomor telephone dan website. Keterangan ini bersifat opsional, jika belum punya perusahaan, dapat anda lewati.



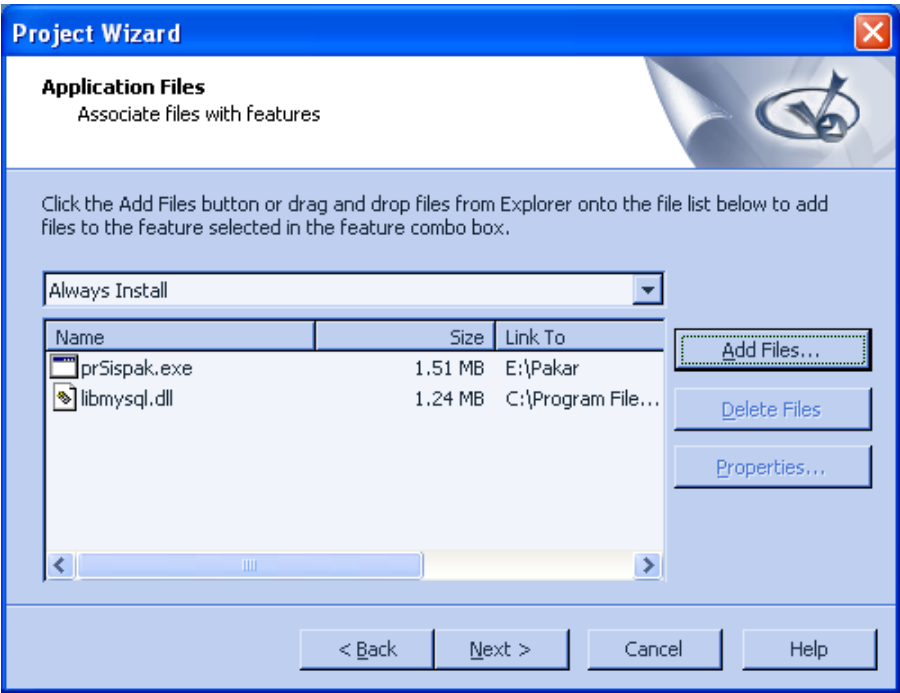
Gambar 11.41. Informasi perusahaan

7. Kemudian muncul tampilan seperti pada gambar dibawah ini, selanjutnya klik next :



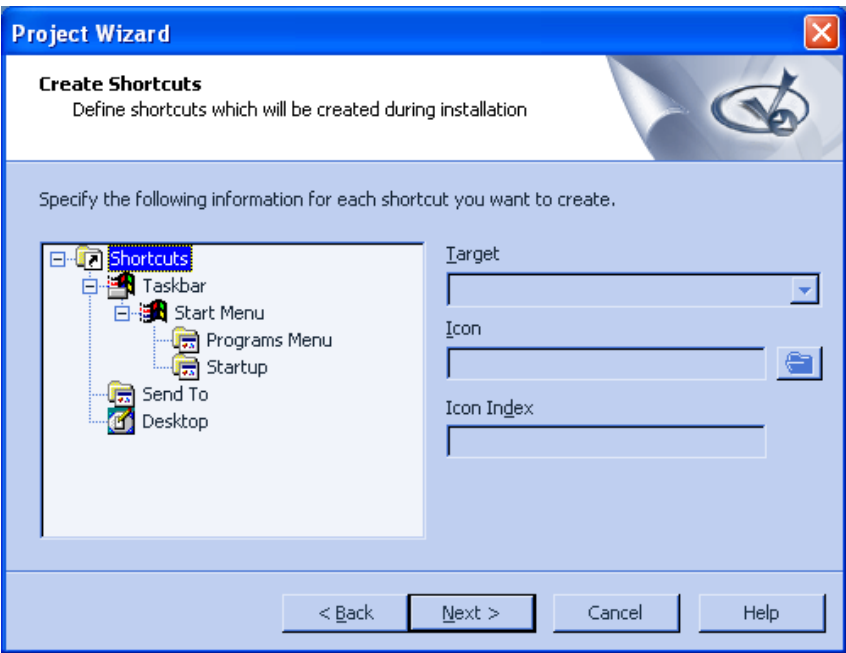
Gambar 11.42. Halaman fitur aplikasi.

8. Klik next, kemudian klik **add files** , kemudian buka file excutable project anda dan sertakan lib mysql.dll



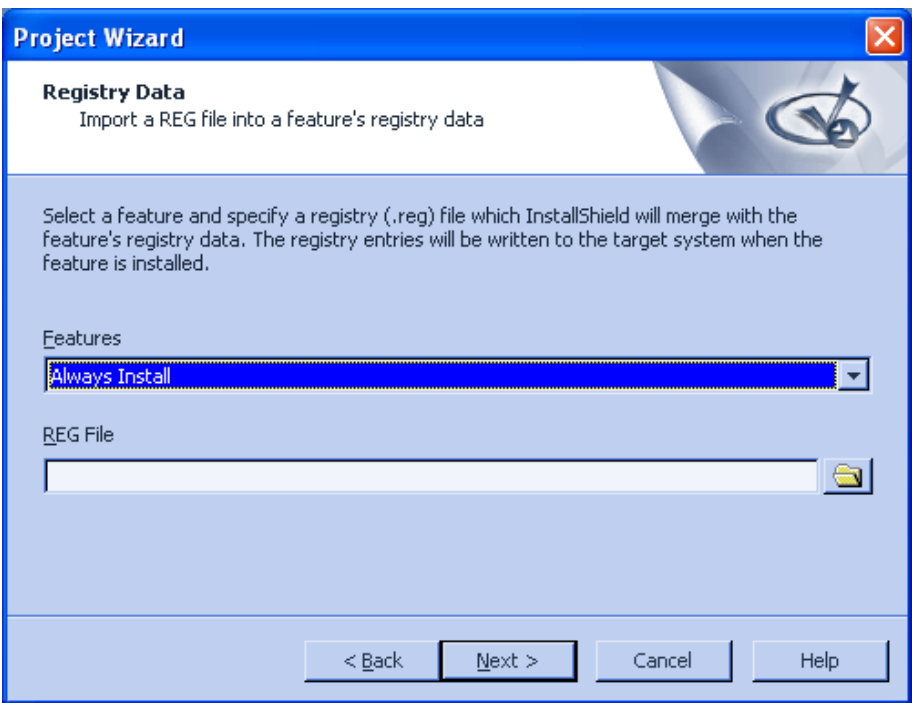
Gambar 11.43. Gambar file yang akan di built

9. Lanjutkan klik next, kemudian muncul tampilan dibawah ini :



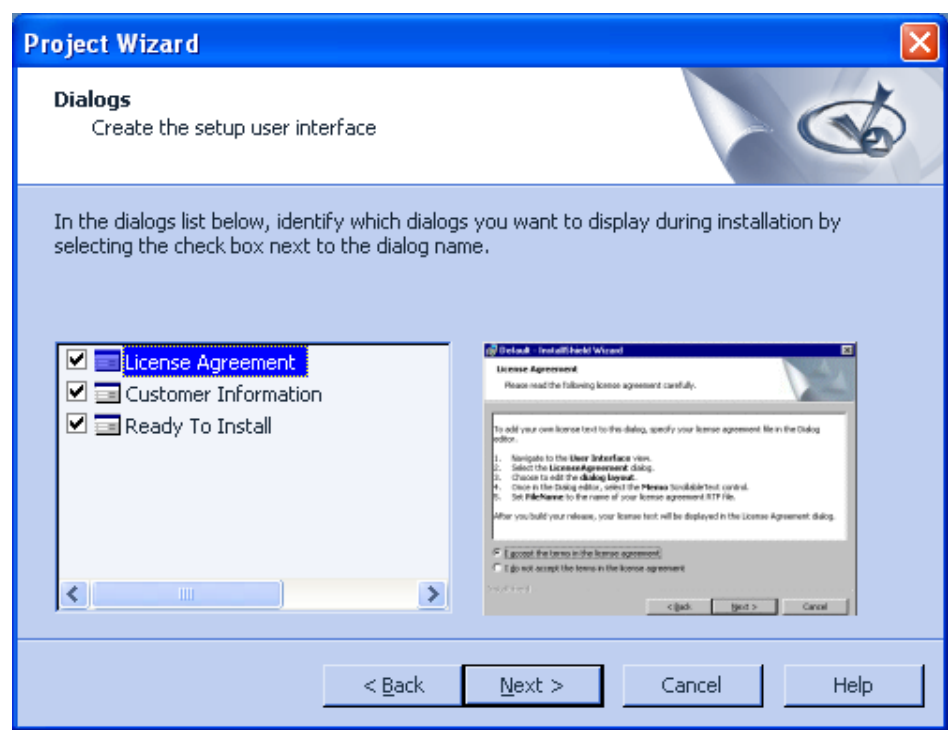
Gambar 11.44. Gambar setup shortcuts

10. Lanjutkan klik next, sehingga muncul tampilan dibawah ini :



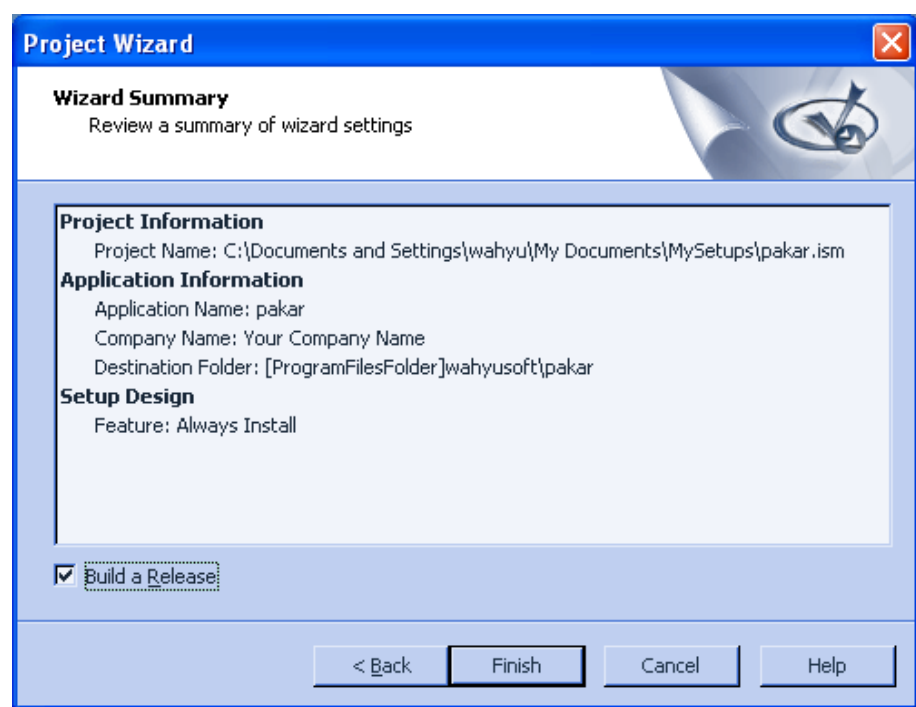
Gambar 11.45. Gambar registrasi data

11. Lanjutkan klik next, sehingga muncul tampilan dibawah ini :



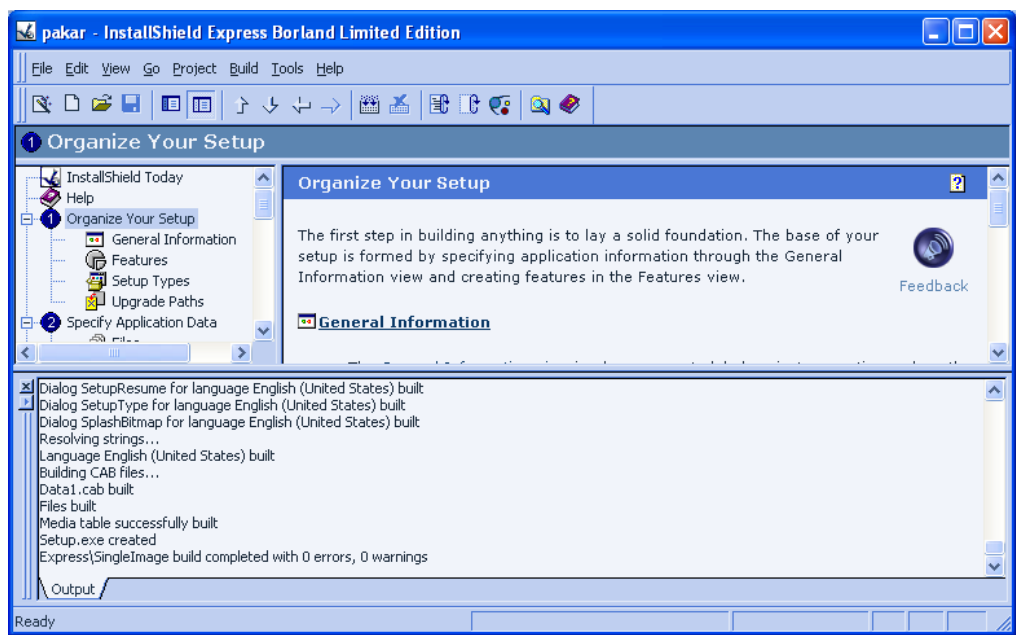
Gambar 11.45. Gambar dialog box instalasi

12. Lanjutkan klik next, sehingga muncul tampilan dibawah ini :



Gambar 11.46. Proses siap build

12. Klik finish, maka proses build setup.exe dijalankan. Setelah muncul konfirmasi sukses seperti pada gambar dibawa ini, maka proses telah selesai.

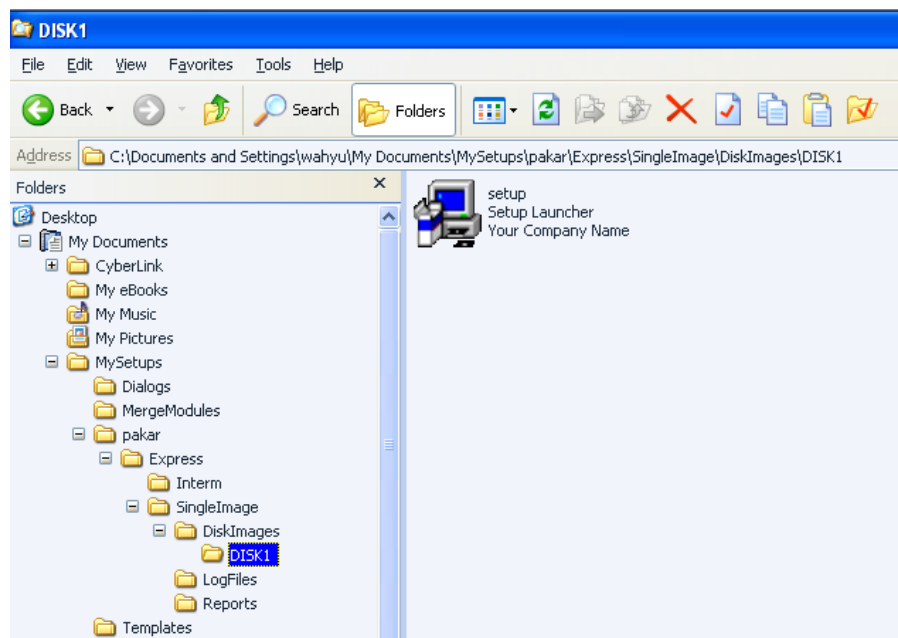


Gambar 11.47. Konfirmasi selesai

Mencoba Proses Instalasi

Setelah berhasil membuat instalasi, sekarang kita coba file setup yang telah anda buat.

1. Buka setup pada direktori Disk 1, pada destination instalasi yang anda buat seperti pada gambar dibawah ini :



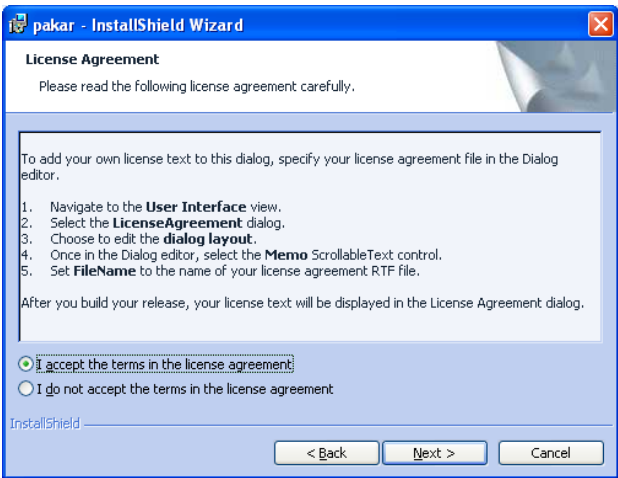
Gambar 11.48. Letak file setup

2. Klik next, jika berhasil akan muncul seperti pada gambar dibawah ini :



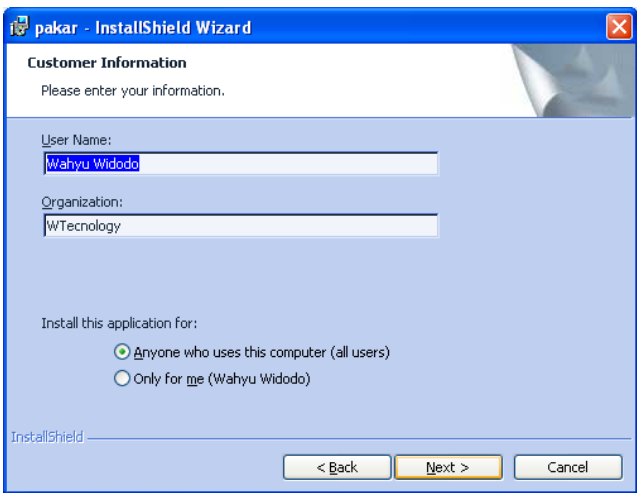
Gambar 11.49. Halaman awal instaler

3. Lanjutkan klik next, pilih accept pada license agreement. Seperti pada gambar dibawah ini :



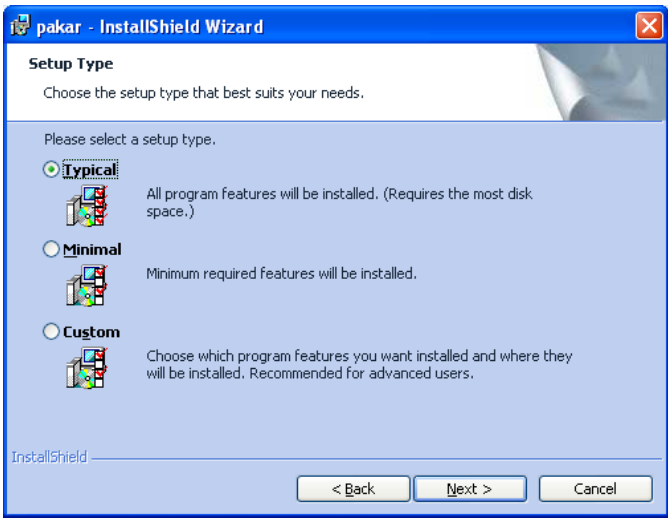
Gambar 11.50. Halaman persetujuan lisensi

4. Lanjutkan klik next sehingga muncul seperti gambar dibawah ini :



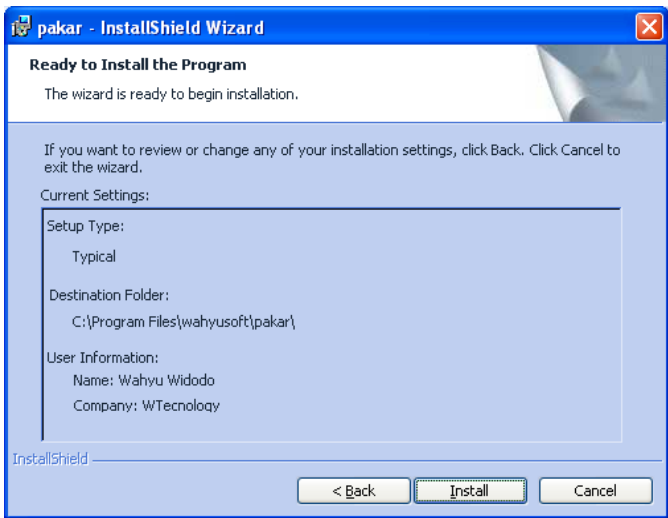
Gambar 11.51. Halaman customer information

5. Lanjutkan klik next, sehingga muncul seperti gambar dibawah ini :



Gambar 11.52. Halaman tipe setup

6. Lanjutkan klik next, sehingga muncul seperti gambar dibawah ini :



Gambar 11.53. Proses siap install

7. Klik install, tunggu proses dan klik finish. Selanjutnya cek hasil instalasi pada start->program files. Jika ada, berarti program telah berhasil dipasang.