

## MODUL III

### Konversi Data, Pengaturan Obyek dan Kotak Pesan

#### A. Fungsi konversi data

Dalam proses pengolahan data, terkadang melibatkan banyak jenis data, baik data jenis string, data numerik, ataupun jenis data yang lain. Khusus untuk proses perhitungan diperlukan cara khusus agar perhitungan dapat berjalan dengan baik dan menghasilkan keluaran yang benar. Cara yang dimaksudkan adalah proses untuk melakukan konversi data. Berikut ini akan diberikan beberapa fungsi untuk melakukan konversi data sebagai berikut:

<b>StrToInt</b>	: merubah data string ke integer
<b>IntToStr</b>	: merubah data integer ke string
<b>FloatToStr</b>	: merubah data real ke string
<b>StrToFloat</b>	: merubah data string ke float

#### B. Pengaturan Obyek

Secara umum pengaturan obyek atau komponen-komponen yang digunakan dilakukan dengan memanfaatkan object inspector dan merubah propertiesnya.

##### 1. Mengaktifkan dan Menonaktifkan Obyek

Untuk memberikan dampak tersebut pada objek yang digunakan, sebenarnya sudah disediakan pada properties (*object inspector*), pada bagian Enabled. Tetapi jika ditetapkan pada object inspector, maka sifatnya permanen selama aplikasi dijalankan. Adakalanya objek tertentu kondisinya berubah-ubah. Untuk itu diperlukan diatur dengan kode program.

Bentuk umum perintah :

Objek.Enabled:=true; jika ingin diaktifkan

Objek.Enabled:=false; jika ingin tidak diaktifkan

Contoh :

Edit1.Enabled:=false; berarti objek Edit1 di non aktifkan

Combobox1.Enabled:=true; berarti objek Combobox1 diaktifkan

##### 2. Menampilkan dan Tidak Menampilkan Obyek

Untuk memberikan dampak tersebut pada objek yang digunakan, sebenarnya sudah disediakan pada properties (*object inspector*), pada bagian Visible. Tetapi jika ditetapkan pada object inspector, maka sifatnya permanen selama aplikasi dijalankan. Adakalanya objek tertentu kondisinya berubah-ubah. Untuk itu diperlukan diatur dengan kode program.

Bentuk umum perintah :

Objek.Visible:=true; jika ingin ditampilkan

Objek.Visible:=false; jika ingin tidak ditampilkan

Contoh :

Edit1.Visible:=false; berarti objek Edit1 tidak ditampilkan

Combobox1.Visible:=true; berarti objek Combobox1 ditampilkan

GroupBox1.Visible:=false; berarti objek GroupBox1 tidak ditampilkan

### C. Kotak Pesan

Banyak jenis kotak pesan yang bisa diterapkan pada Borland Delphi, namaun pada bahasan ini hanya akan diberikan yang paling sering digunakan saja.

1. ShowMessage, menampilkan kotak pesan dengan tombol OK

Contoh 1:

**showmessage('Anda Coba ShowMessage');**

Hasil:



Gambar 3.1 Kotak Pesan Dengan Tombol OK Contoh 1

Contoh 2:

**showmessage('Data tidak ditemukan');**

Hasil:



Gambar 3.2 Kotak Pesan Dengan Tombol OK Contoh 2

2. ShowMessagePos, menampilkan kotak pesan pada posisi tertentu

Contoh:

**showmessagepos('Anda Mencoba ShowmessagePos',100,500);**

Hasil:

Tampilan sama seperti contoh di atas, hanya saja kotak pesan ditampilkan pada posisi kolom 100 dan baris 500.

3. MessageDlg, menampilkan kotak pesan di tengah layar  
Kotak pesan ini cukup kompleks, karena mempunyai lebih banyak parameter, yaitu tombol-tombol dan tipe kotak pesan. Selain itu juga menghasilkan nilai dari tombol yang dipilih.

Tabel 3.1 Tentang Tombol Yang Dihasilkan dari Perintah MessageDlg

Perintah	Tombol Yang Ditampilkan
[mbOK]	Tombol OK
[mbCancel]	Tombol Cancel
[mbYes]	Tombol Yes
[mbNo]	Tombol No
[mbRetry]	Tombol Retry
[mbAbort]	Tombol Abors
[mbIgnore]	Tombol Ignore

[mbOk, mbCancel]	Tombol OK dan tombol Cancel
[mbYes, mbNo]	Tombol Yes dan tombol No

Tabel 3.2 Tentang Icon Yang Dihasilkan dari Perintah MessageDlg

Tippe Kotak Pesan	Gambar/Bitmap
mtWarning	Tanda seru warna kuning
mtError	Tanda silang warna merah
mtInformation	Huruf i warna biru
mtConfirmation	Tanda tanya warna biru
mtCustom	Tdak ada

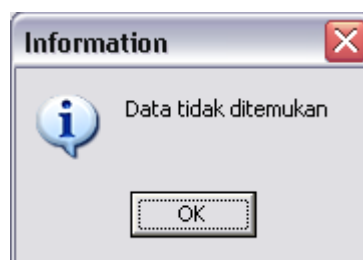
Tabel 3.3 Nilai Kembalian Saat Memilih Tombol

Tombol	Nilai Kembali
Tombol OK	[mrOK]
Tombol Cancel	[mrCancel]
Tombol Yes	[mrYes]
Tombol No	[mrNo]
Tombol Retry	[mrRetry]
Tombol Abors	[mrAbort]
Tombol Ignore	[mrIgnore]

Contoh 1:

```
messagedlg('Data tidak ditemukan',mtInformation,[mbOK],0);
```

Hasil :

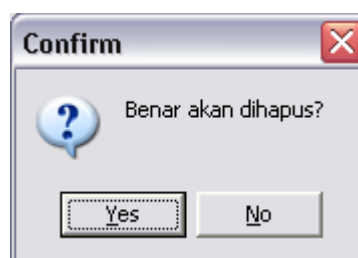


Gambar 3.3 Kotak Pesan Dengan Messagedlg Contoh 1

Contoh 2:

```
if messagedlg('Benar akan dihapus?',  
mtconfirmation,[mbyes,mbno],0)=mryes then  
Listbox1.DeleteSelected;
```

Hasil :



Gambar 3.4 Kotak Pesan Dengan MessageDlg Contoh 2

#### D. Praktek

##### 1. Implementasi Fungsi Konversi Data

###### a. Merancang Form

- ❖ Panggila Borland Delphi sampai disajikan sebuah Project baru dengan sebuah Form baru pula, maka rancanglah form seperti terlihat pada Gambar 3.5
- ❖ Komponen yang digunakan adalah 4 buah **Label**, 1 buah **Combobox**, dan 4 buah **Edit**. Atur properti Combobox dan Edit pada bagian Text isiannya dihapus saja supaya tulisan yang tampil pada masing-masing objek/komponen tersebut akan dihilangkan.
- ❖ Buat isi pilihan pada ComboBox dengan cara klik pada objek ComboBox dan atur properti Items, klik tombol titik-titik. Saat disajikan kotak String List Editor isikan pilihan berturut-turut untuk golongan berupa bilangan romawi I, II, III, dan IV, setelah selesai akhiri dengan mengklik tombol OK.



Gambar 3.5 Desain Form Fungsi Konversi Data

#### Aturan Bisnis:

Dari rancangan form di atas, diharapkan saat program dijalankan dan mengklik pilihan Golongan (Combobox1) maka Gaji Pokok (Edit1), Tunjangan Keluarga (Edit2), Tunjangan Kesehatan (Edit3), dan Jumlah Penghasilan (Edit4) akan tampil secara otomatis. Aturan-aturan yang berlaku adalah :

##### Untuk mengisi Gaji Pokok :

- Jika Golongan = 'I', maka Gaji Pokok = 100000
- Jika Golongan = 'II', maka Gaji Pokok = 200000
- Jika Golongan = 'III', maka Gaji Pokok = 300000
- Jika Golongan = 'IV', maka Gaji Pokok = 400000

##### Untuk mengisi Tunjangan Keluarga :

10% dari Gaji Pokok

##### Untuk mengisi Tunjangan Kesehatan :

5% dari Gaji Pokok

##### Untuk mengisi Jumlah Penghasilan

Gaji Pokok + Tunjangan Keluarga + Tunjangan Kesehatan

###### b. Menulis Kode Program

- ❖ Klik objek Combobox
- ❖ Klik Tab Event pada object inspector
- ❖ Arahkan mouse pada bagian OnClick, lalu klik dua kali.
- ❖ Saat masuk Code Editor, diantara prosedur dan Begin sisipkan baris kosong lalu ketik kode program sebagai berikut:

```
var a : integer;
```

```
    b,c,d : real;
```

- ❖ Diantara Begin dan End tulis perintah sebagai berikut :

```
if combobox1.Text='I' then
    edit1.text:='100000'
else if combobox1.Text='II' then
    edit1.text:='200000'
else if combobox1.Text='III' then
    edit1.text:='300000'
else if combobox1.Text='IV' then
    edit1.text:='400000';
a:=strtoint(edit1.Text);
b:=0.1*a;
c:=0.05*a;
d:=a+b+c;
edit2.Text:=floattostr(b);
edit3.Text:=floattostr(c);
edit4.text:=floattostr(d);
```

## 2. Implementasi Properties Enabled

Properti tersebut bermanfaat untuk mengaktifkan atau menon-aktifkan suatu objek. Jika nilainya adalah True, maka berfungsi untuk mengaktifkan objek, sedangkan jika nilainya False berfungsi untuk me-nonaktifkan.

- a. Setelah latihan pertama selesai, maka tambahkan Form baru ke project yang sedang Anda hadapi saat ini dan rancanglah form tersebut seperti Gambar 3.6.
- b. Komponen yang digunakan adalah 2 Label untuk Nama Mahasis dan Alamat. 2 Edit untuk isian nama dan isian alamat. 1 Listbox. 4 Button
- c. Atur properti Edit1 dan Edit1 pada bagian Text isian Edit1, dan Edit2 dihilangkan.
- d. Atur properti untuk Button khususnya pada bagian **Caption** dan **Name**. Untuk bagian **Caption** berturut-turut diganti dengan **Isi**, **Simpan**, **Hapus**, dan **Tutup**. Untuk bagian **Name** berturut-turut **TblIsi**, **TblSimpan**, **TblHapus**, dan **TblTutup**.

Gambar 3.6 Desain Form Fungsi Enabled

**Kasus:**

Dari rancangan form di atas, diharapkan bahwa :

- Saat form dijalankan, maka isian Nama Mahasiswa (Edit1) dan isian Alamat (Edit2), tombol Simpan, tombol Hapus dalam keadaan tidak aktif.
- Saat Tombol Isi dipilih/klik maka isian Nama, isian Alamat aktif. Tombol Isi tidak aktif, tombol Simpan aktif, lalu kursor langsung berada pada isian Nama.
- Saat tombol Simpan dipilih, maka data yang diketik pada isian Nama dan isian Alamat akan masuk ke objek ListBox, dengan susunan berjejer ke kanan.
- Saat ListBox diklik, maka tombol Hapus akan diaktifkan.
- Saat tombol Hapus dipilih, maka akan ada konfirmasi untuk penghapusan data. Jika dijawab Yes, maka data yang diseleksi akan dihapus dari Listbox.
- Saat tombol Tutup dipilih, maka akan keluar dari Form.

**Penyelesaian:**

- Pastikan objek yang aktif adalah Form (klik ditempat yang kosong pada Form). Klik tab Events pada Object Inspector, lalu arahkan mouse pada bagian **OnActive** dan klik dua kali. Tulis perintah sebagai berikut:

```
edit1.Enabled:=false;  
edit2.Enabled:=false;  
tblsimpan.Enabled:=false;  
tblhapus.Enabled:=false;
```

- Klik objek tombol Isi, lalu klik tab Events pada Object Inspector, arahkan mouse pada bagian **OnClick** dan klik dua kali. Tulis perintah sebagai berikut:

```
edit1.Enabled:=true;  
edit2.Enabled:=true;  
tblisi.Enabled:=false;  
tblsimpan.Enabled:=true;  
edit1.text:='';  
edit2.text:='';  
edit1.SetFocus;
```

- Klik objek tombol Simpan, lalu klik tab Events pada Object Inspector, arahkan mouse pada bagian **OnClick** dan klik dua kali. Tulis perintah sebagai berikut:

```
Listbox1.Items.Add(Edit1.Text+' '+edit2.text);  
edit1.Enabled:=false;  
edit2.Enabled:=false;  
tblisi.Enabled:=true;  
tblsimpan.Enabled:=false;
```

- Klik objek ListBox, lalu klik tab Events pada Object Inspector, arahkan mouse pada bagian **OnClick** dan klik dua kali. Tulis perintah sebagai berikut:

```
tblhapus.Enabled:=true;
```

- Klik objek tombol Hapus, lalu klik tab Events pada Object Inspector, arahkan mouse pada bagian **OnClick** dan klik dua kali. Tulis perintah sebagai berikut:

```
if messagedlg('Benar akan dihapus?',  
mtconfirmation,[mbyes,mbno],0)=mryes then
```

### **Listbox1.DeleteSelected;**

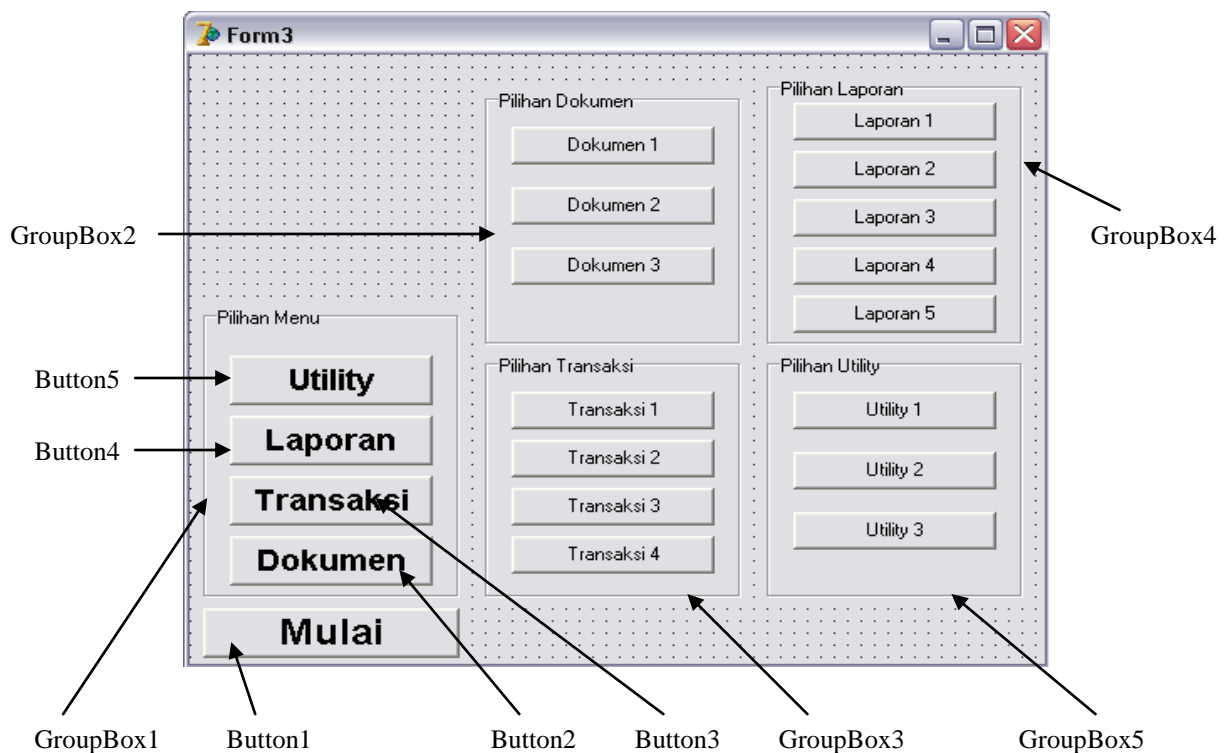
- Klik objek tombol Tutup, lalu klik tab Events pada Object Inspector, arahkan mouse pada bagian **OnClick** dan klik dua kali. Tulis perintah sebagai berikut:

### **application.Terminate;**

### 3. Implementasi Properties Visible

Properti tersebut bermanfaat untuk menampilkan atau tidak menampilkan suatu objek. Jika nilainya adalah True, maka berfungsi untuk menampilkan objek, sedangkan jika nilainya False berfungsi untuk menghilangkan/tdk menampilkan objek.

- Setelah latihan kedua selesai, maka tambahkan Form baru ke project yang sedang Anda hadapi saat ini dan rancanglah form tersebut seperti Gambar 3.7.
- Komponen yang digunakan adalah 5 GroupBox dan beberapa Button.
- Atur properti untuk masing-masing GroupBox dan Button kemudian rubah bagian Caption dan Name
- Caption untuk GroupBox berturut-turut : Pilihan Menu, Pilihan Dokumen, Pilihan Transaksi, Pilihan Laporan, dan Pilihan Utility.
- Name untuk masing-masing GroupBox berturut-turut : Kotakmenu, KotakDokumen, KotakTransaksi, KotakLaporan, dan KotakUtility
- Name untuk Button berturut-turut : TblMulai, TblDokumen, TblTransaksi, TblLaporan, TblUtility. Name untuk yang lain disesuaikan saja sesuai kebutuhan



Gambar 3.7 Racangan Form Menu Penggunaan Properti Visible

- Buat kotak dokumen, kotak transaksi, kotak laporan, dan kotak utility saling berhimpitan yang berada disebelah kanan kotak menu.

### **Kasus :**

- Saat Form dijalankan, maka semua kotak tidak ditampilkan.
- Saat memilih tombol Mulai, yang tampil hanya kotak menu, kotak yang lain tidak ditampilkan.

- Saat memilih tombol Dokumen, maka hanya kotak dokumen yang tampilkan, sedangkan kotak yang lain tidak ditampilkan.
- Saat memilih tombol Transaksi, maka hanya kotak transaksi yang tampilkan, sedangkan kotak yang lain tidak ditampilkan.
- Saat memilih tombol Laporan, maka hanya kotak laporan yang tampilkan, sedangkan kotak yang lain tidak ditampilkan.
- Saat memilih tombol Utility, maka hanya kotak utility yang tampilkan, sedangkan kotak yang lain tidak ditampilkan.

#### Penyelesaian :

- Pastikan objek yang aktif adalah Form (klik ditempat yang kosong pada Form). Klik tab Events pada Object Inspector, lalu arahkan mouse pada bagian **OnActive** dan klik dua kali. Tulis perintah sebagai berikut:

```
kotakmenu.Visible:=false;
kotakdokumen.Visible:=false;
kotaktransaksi.Visible:=false;
kotaklaporan.Visible:=false;
kotakutility.Visible:=false;
```

- Klik objek tombol **Mulai**, lalu klik tab Events pada Object Inspector, arahkan mouse pada bagian **OnClick** dan klik dua kali. Tulis perintah sebagai berikut:

```
kotakmenu.Visible:=true;
kotakdokumen.Visible:=false;
kotaktransaksi.Visible:=false;
kotaklaporan.Visible:=false;
kotakutility.Visible:=false;
```

- Klik objek tombol **Dokumen**, lalu klik tab Events pada Object Inspector, arahkan mouse pada bagian **OnClick** dan klik dua kali. Tulis perintah sebagai berikut:

```
kotakdokumen.Visible:=true;
kotaktransaksi.Visible:=false;
kotaklaporan.Visible:=false;
kotakutility.Visible:=false;
```

- Klik objek tombol **Transaksi**, lalu klik tab Events pada Object Inspector, arahkan mouse pada bagian **OnClick** dan klik dua kali. Tulis perintah sebagai berikut:

```
kotakdokumen.Visible:=false;
kotaktransaksi.Visible:=true;
kotaklaporan.Visible:=false;
kotakutility.Visible:=false;
```

- Klik objek tombol **Laporan**, lalu klik tab Events pada Object Inspector, arahkan mouse pada bagian **OnClick** dan klik dua kali. Tulis perintah sebagai berikut:

```
kotakdokumen.Visible:=false;
kotaktransaksi.Visible:=false;
kotaklaporan.Visible:=true;
kotakutility.Visible:=false;
```

- Klik objek tombol **Utility**, lalu klik tab Events pada Object Inspector, arahkan mouse pada bagian **OnClick** dan klik dua kali. Tulis perintah sebagai berikut:



```
kotakdokumen.Visible:=false;  
kotaktransaksi.Visible:=false;  
kotaklaporan.Visible:=false;  
kotakutility.Visible:=true;
```

#### **E. Soal-Soal**

1. Apa kelebihan pengaturan obyek ditetapkan dengan menulis kode program?
2. Apa yang terjadi jika dalam perhitungan tidak dilakukan proses konversi data? Apa pesan yang ditampilkan saat terjadi kesalahan?
3. Buatlah sebuah form yang dapat berfungsi seperti kalkulator!. Rancangan terserah Anda (dapat melihat fasilitas calculator yang ada di Accessories atau kalkulator asli).
4. Buatlah form untuk melakukan tampilan secara bergantian seperti lampu pengatur lalu lintas. Dapat juga Anda menggunakan beberapa gambar. Supaya program dapat berjalan terus menerus selama belum diakhiri, maka dapat memanfaatkan Timer.

## **MODUL IV**

### **Menu, ToolBar, Hint, Jam dan Tanggal**

#### **A. Menu**

Menu memberikan cara yang mudah untuk menjalankan kelompok perintah-perintah. Menu Designer memungkinkan Anda untuk menambahkan sebuah menu ke dalam form. Anda hanya perlu menambahkan sebuah komponen menu ke dalam form, membuka Menu Designer dan mengetikkan item-item menu pada jendela Menu Designer. Pada saat perancangan, Anda dapat menambah, menghapus item-item menu, atau drag dan drop untuk mengatur ulang.

Ada dua komponen menu, yaitu MainMenu dan PopUpMenu. Komponen MainMenu membuat menu yang menempel pada title bar dari form disebut juga dengan menu pull down. Sedangkan PopUpMenu membuat menu pada saat user mengklik kanan pada form atau pada sebuah komponen disebut juga menu melayang.

Untuk mengisi desain MainMenu, maka klik dua kali pada objek MainMenu, lalu isikan pilihan-pilihan menu yang dibutuhkan baik untuk menu utama maupun sub menu. Properti yang perlu dirubah adalah Caption dan Name untuk masing-masing pilihan. Saat mengisi Caption, maka Name juga sudah diberi secara otomatis. Jadi apabila ingin lebih praktis, maka Name cukup dimanfaatkan nama yang sudah diberikan secara otomatis tersebut.

#### **B. ToolBar**

Saat ini hampir setiap aplikasi Windows mempunyai toolbar atau speedbar. Toolbar digunakan untuk menggantikan beberapa item menu yang sering digunakan. Tujuannya agar pengguna bisa memilih item menu tersebut dengan lebih cepat.

#### **C. Hint**

Hint merupakan teks di dalam kotak kecil yang ditampilkan bila kursor mouse berada di atas objek (button, Edit, ComboBox, dll).

#### **D. Jam**

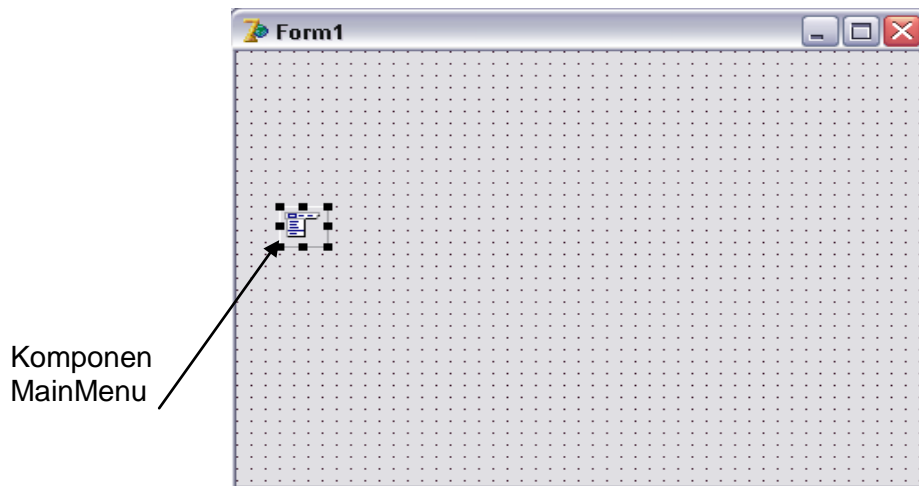
Untuk keperluan tertentu, terkadang dibutuhkan untuk menampilkan jam (jam sistem) pada form. Untuk keperluan tersebut dapat digunakan komponen Timer ditambah komponen lain misalnya Label untuk menampilkan data jam.

#### **E. Tanggal**

Selain menampilkan jam seperti contoh di atas, adakalanya dibutuhkan juga untuk menampilkan tanggal (tanggal sistem) pada form. Komponen yang digunakan untuk menampilkan tanggal dapat menggunakan komponen Label atau Edit.

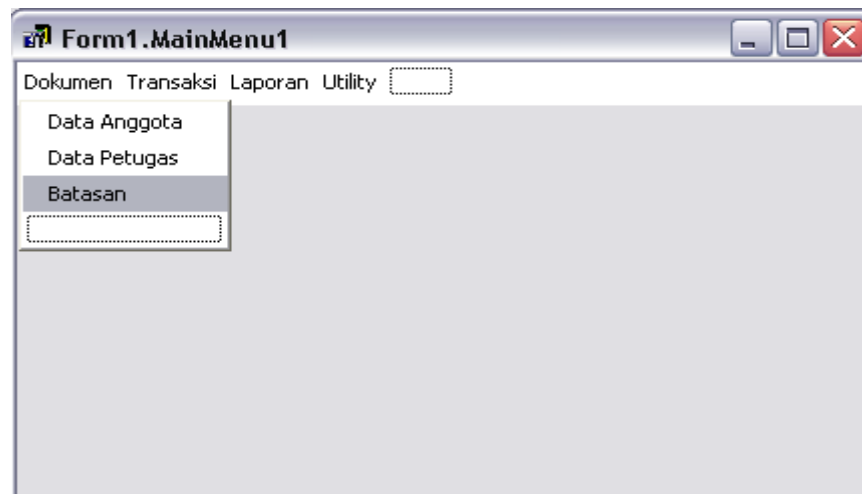
## F. Praktek

1. Membuat Main Menu (Menu *Pulldown*)
  - a. Buatlah Project baru, dan pastikan Anda berada pada sebuah form baru yang akan dijadikan menu utama (saat memanggil Borland Delphi kondisi ini sudah otomatis)
  - b. Atur properties yang sesuai supaya form tampil lebih bagus (anda bisa menambahkan beberapa label untuk membut tulisan yang akan ditampilkan pada menu).
  - c. Buat kode program untuk kejadian OnClose dan ketikkan perintah:  
**Application.Terminate;**
  - d. Ambil komponen MainMenu, letakkan pada form seperti terlihat pada Gambar 4.1.



Gambar 4.1 Racangan Form Menu Utama

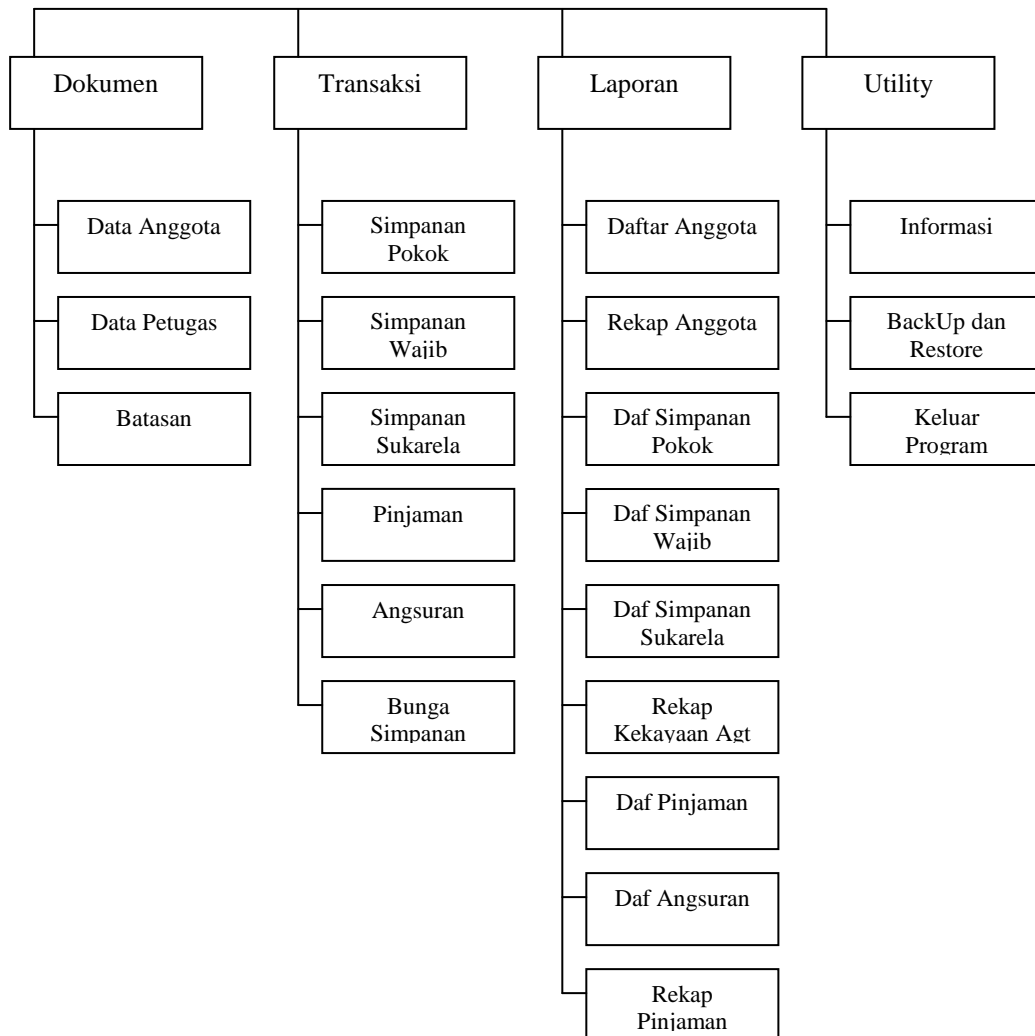
- e. Klik dua kali pada komponen MainMenu atau dari page properties object inspector, pilih Items, lalu klik ganda pada tanda (...), sehingga akan ditampilkan desain menu



Gambar 4.2 Tampilan Proses Pembuatan Desain Menu

- f. Isikan pilihan-pilihan menu yang diinginkan, baik untuk pilihan utama (secara horizontal/mendatar) maupun pilihan sub menu (secara vertical). Misalkan akan dibuat empat pilihan menu utama yaitu **Dokumen**, **Transaksi**, **Laporan**, dan **Utility**. Struktur menu selengkapnya dapat dilihat pada Gambar 4.3, dan buatlah pilihan menu dan sub menu sesuai gambar. Hal yang paling penting dilakukan adalah mengetik judul menu atau judul pilihan di Object Inspector pada bagian Caption. Selain Caption, sebenarnya perlu juga untuk menentukan Name dari setiap pilihan.

Tapi jika tidak ditentukan, maka secara otomatis akan disamakan dengan yang diketik pada bagian Caption dengan diberi indeks (agar cepat gunakan saja nama yang diberikan secara otomatis tersebut). Setelah selesai semua pilihan, klik tombol Close dari desain menu, untuk kembali ke form menu.



Gambar 4.3 Contoh Struktur Menu Utama

## 2. Berbagai Operasi Untuk Item Menu

### a. Menyisipkan Baris Item Menu

- Pastikan posisi berada pada desain menu. Letakkan kursor pada posisi dimana baris akan disisipkan.
- Klik kanan, lalu pilih Insert.
- Isikan baris yang ingin disisipkan

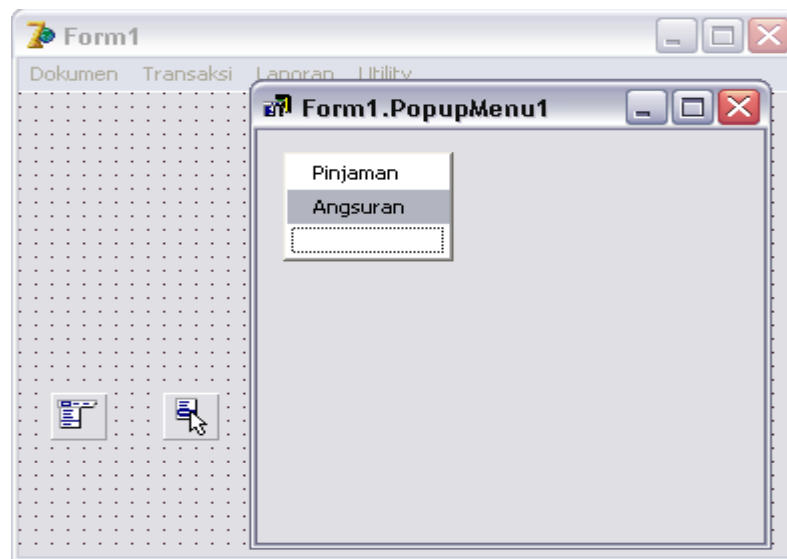
### b. Menghapus Baris Item Menu

- Pastikan posisi berada pada desain menu. Letakkan kursor pada posisi dimana baris akan dihapus.
- Klik kanan, lalu pilih Delete.

### c. Membuat Garis Pembatas Item Menu

- Letakkan kursor pada posisi dimana garis akan ditampilkan (jika pada baris tersebut sudah ada pilihan, maka lakukan penyisipan baris sampai pada baris tersebut merupakan baris kosong).

- Ketikkan karakter minus (-).
- d. Membuat Sub Menu
- Letakkan kursor pada posisi item menu yang akan dibuatkan sub menu.
  - Klik kanan, lalu pilih Create SubMenu.
  - Isikan baris yang ingin dijadikan sub menu
3. Membuat PopUp Menu (Menu Layang)
- a. Letakkan komponen PopupMenu pada form
  - b. Klik dua kali pada komponen PopupMenu, sehingga akan ditampilkan kotak dialog atau desain menu popup. Ketik pilihan yang akan dijadikan tampilan pada menu popup, misalnya : Pinjaman dan Angsuran



Gambar 4.4 Desain Menu PopUp

- c. Atur properti Form pada bagian PopupMenu dan isikan dengan nama Popupmenu yang telah dibuat misalnya PopupMenu1.
  - d. Silahkan jalankan form untuk melihat hasil
4. Menangani Kejadian Menu
- a. Apabila pada project baru memiliki satu form (form menu saja) maka tambahkan terlebih dahulu beberapa form ke dalam project untuk dijadikan ujicoba (misalkan tambahkan 3 form yaitu untuk pengolahan data anggota, form untuk pengolahan data petugas, dan pengolahan data batasan) untuk dapat menangani kejadian **OnClick** masing-masing pilihan.
  - b. Atur properti dan bentuk tampilan masing-masing form tersebut supaya jelas perbedaan antara satu dengan yang lain, misalnya dibedakan saja dengan judul (dibuat dengan label)
  - c. Aktifkan kembali form menu
  - d. Pilih item yang ingin Anda tangani (tidak harus masuk ke Form Designer). Delphi secara otomatis akan membuatkan penanganan kejadian OnClick untuk item menu yang Anda pilih.  
Jika kejadian OnClik akan dihubungkan dengan suatu form (misalnya digabungkan dengan Form2), maka kode program yang diberikan adalah sebagai berikut:

**Form2.Show;** atau  
**Form2.Showmodal;**

Sedangkan jika kejadian OnClick akan dihubungkan dengan suatu kotak pesan (misalnya untuk proses keluar dari program, maka kode program yang diberikan adalah sebagai berikut:

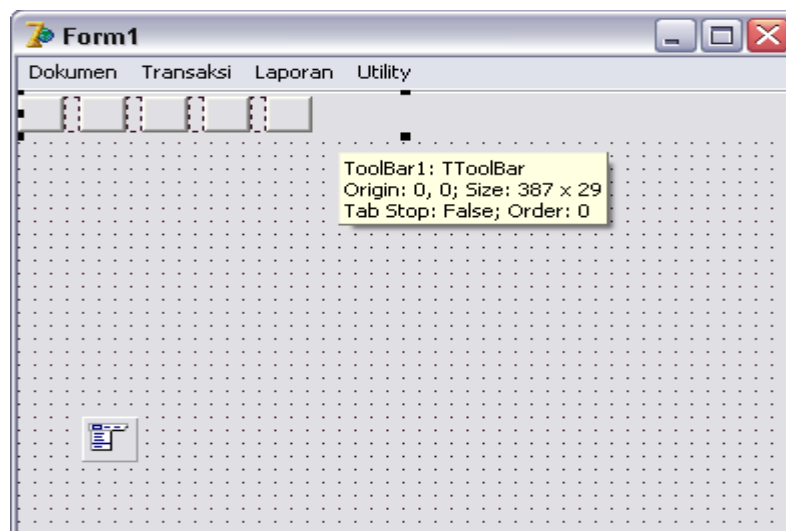
```
If messagedlg('Benar mau keluar program ini?',  
mtconfirmation,[mbytes,mbno],0)=mryes then  
application.Terminate;
```

- e. Apabila dalam proses menghubungkan menu dengan masing-masing form ada tampilan pesan error pada compile, itu tidak apa-apa. Silahkan dilanjutkan dan dicoba untuk menjalankan menu yang telah dihubungkan.

### G. Membuat ToolBar

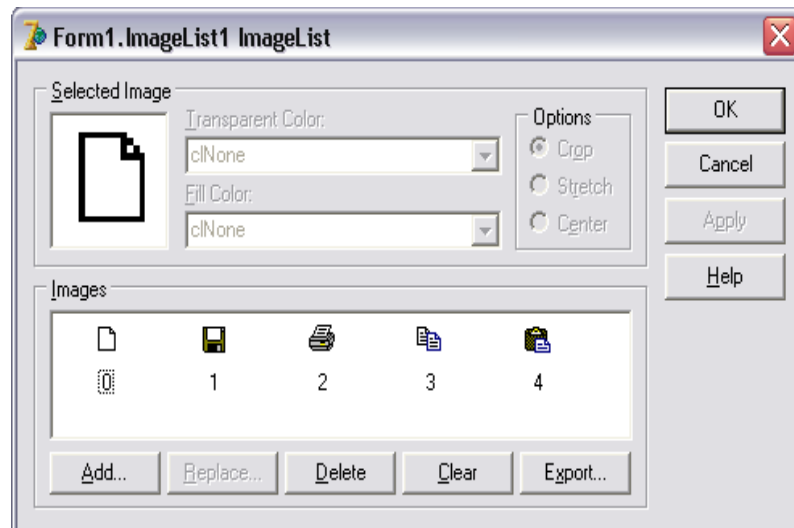
Saat ini hampir setiap aplikasi Windows mempunyai toolbar atau speedbar. Toolbar digunakan untuk menggantikan beberapa item menu yang sering digunakan. Tujuannya agar pengguna bisa memilih item menu tersebut dengan lebih cepat. Bila diinginkan pada toolbar akan ditampilkan icon (gambar kecil), sebaiknya pada folder dimana project disimpan sudah disediakan gambarkan-gambar yang dibutuhkan. Anda bisa mencari pada folder aplikasi-aplikasi yang ada, misalnya pada Borland atau Visual Studio.

1. Untuk membuat toolbar, pertama-tama letakkan komponen TToolBar pada form. Komponen ini dapat diambil dari tab Win32.
2. Langkah selanjutnya adalah meletakkan tombol-tombol di atas panel ini. Anda bisa menambahkan tombol dengan mengklik kanan dan memilih menu **New Button**. Untuk menambahkan pemisah tombol, klik kanan, pilih **New Separator**. Lakukan langkah ini sampai semua tombol yang dibutuhkan selesai. Secara lengkap proses pembuatan toolbar dapat dilihat pada Gambar 4.5.



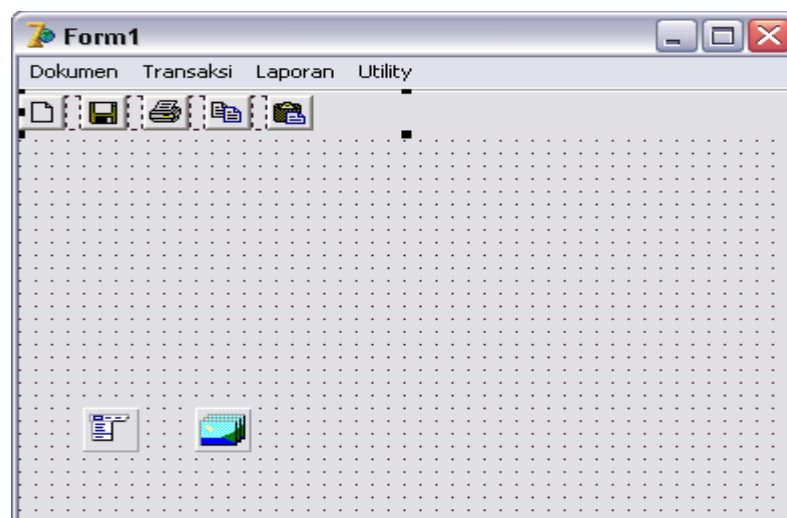
Gambar 4.5 Pembuatan ToolBar

3. Untuk menambahkan ikon di atas tombol ini, Anda memerlukan komponen **ImageList**. Letakkan komponen ImageList. Komponen ImageList dapat diambil pada tab Win32. Setelah komponen ImageList diletakkan pada form maka klik dua kali untuk mengisi gambar pada desain ImageList, seperti terlihat pada Gambar 4.6. Gambar yang diambil sesuaikan dengan fungsi toolbar.



Gambar 4.6 Kotak Dialog Menambah Image List

4. Hubungkan ToolBar dengan ImageList. Klik komponen ToolBar, atur properti Image dan isikan dengan nama ImageList1
5. Anda bisa juga menampilkan tombol dengan gaya datar (*flat*) seperti Internet Explorer. Caranya adalah dengan mengubah properti ToolBar pada bagian Flat menjadi true.



Gambar 4.7 Menu yang Sudah Jadi Dengan ToolBar

#### H. Menambahkan Hint

Hint merupakan teks di dalam kotak kecil yang ditampilkan bila kursor mouse berada di atas objek (button, Edit, ComboBox, ToolBar, dll). Karena saat ini bahasan kita adalah toolbar, maka pembuatan hint kita coba pada objek toolbar.

- Klik tombol yang akan dibuatkan Hint. Isi properti Hint dari masing-masing tombol dengan teks yang ingin Anda jadikan Hint.
- Ubah properti ShowHint dari ToolBar dimana tombol-tombol toolbar diletakkan menjadi true. Jalankan kembali program Anda, coba perhatikan dampak dari menambahkan Hint pada objek toolbar.

Jika diinginkan objek-objek yang ada pada form ditampilkan Hint-nya, maka properti Form pada bagian ShowHint dirubah menjadi True.

### I. Menampilkan Jam

- Ambil komponen Timer (pada tab System) dan letakkan di form
- Ambil komponen Label (pada tab Standard) dan letakkan di form di posisi yang diinginkan.
- Klik dua kali pada komponen Timer sehingga akan masuk ke editor program dan tuliskan perintah untuk menampilkan jam berikut (dianggap komponen Label yang digunakan adalah Label1):

**Label1.Caption:=formatdatetime('hh:mm:ss',now);**

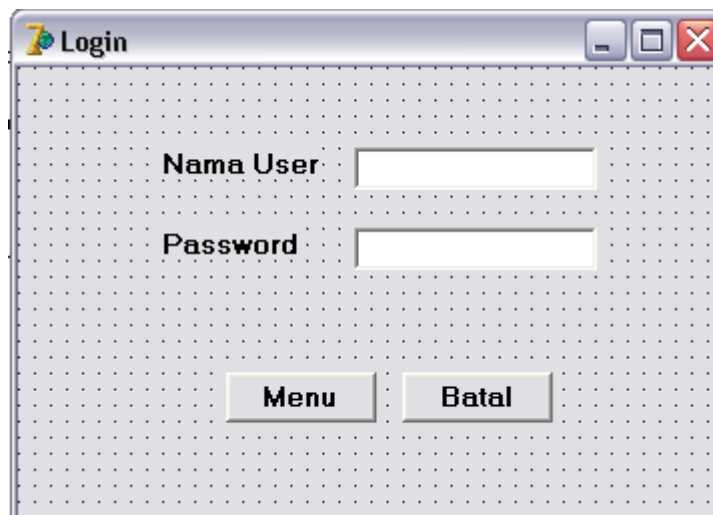
### J. Menampilkan Tanggal

- Ambil komponen Label dan letakkan pada form sebagai tempat untuk menampilkan tanggal
- Tulis kode program pada event Form OnCreat

**Label2.caption:=formatdatetime('dd-mm-yy',now);**

### K. Membuat Form Password

1. Tambahkan Form baru ke dalam Project (dianggap menjadi Form5)
2. Gunakan dua buah komponen Label tempatkan pada form dan gantilah Captionnya secara berturut-turut, Label1 diganti dengan Nama User, Label2 diganti dengan Password. Jika diperlukan ukuran font dapat diperbesar sesuai kebutuhan, dan jika perlu ditebalkan.
3. Tambahkan dua buah komponen Edit yang diletakkan di sebelah kanan Label. Hilangkan tulisan Edit1 dan Edit2 dengan menghapus pada properties di bagian Text.
4. Tambahkan dua buah komponen Button dan letakkan di bagian bawah. Aturlah properties dari masing-masing Button. Untuk Button1 gantilah Caption dengan tulisan Menu, lalu Name diganti dengan tblmenu. Untuk Button2 gantilah Caption dengan tulisan Batal, lalu Name diganti dengan tblbatal. Jika diperlukan ukuran font juga bisa diperbesar seperlunya.



Gambar 4.8 Form Password

5. Klik dua kali pada komponen Button1 (tblmenu) dan tulis kode program (yang diketik adalah yang tercetak tebal) sebagai berikut:



```

procedure TForm5.tbImenuClick(Sender: TObject);
var a:string;
    b:string;
begin
    a:=edit1.Text;
    b:=edit2.Text;
    if a='Admin' then
        if b='a' then
            begin
                form5.hide;
                form1.Show;
            end
        else
            messagedlg('Password salah...',mtinformation,[mbok],0)
        else
            messagedlg('User salah...',mtinformation,[mbok],0);
    end;

```

6. Klik dua kali pada komponen Button2 (tblkeluar) dan tulis kode program (yang diketik adalah yang tercetak tebal) sebagai berikut:

```

procedure TForm5.tblbatalClick(Sender: TObject);
begin
    form5.close;
end;

```

#### **L. Soal-Soal**

1. Apa beda insruksi Show dengan ShowModal yang digunakan untuk mengeksekusi form. Jelaskan secara rinci.
2. Bualah sebuah prototipe aplikasi, dimana pada aplikasi tersebut ada form untuk password, ada form menu, ada form-form untuk pengolahan data (buatlah form sebanyak pilihan yang ada pada menu walaupun hanya dibedakan dari judulnya saja, minimal form untuk pilihan menu Olah Data dan menu Layanan).
  - a. Olah Data
    - Data Pasien
    - Data Dokter
    - Data Obat
    - Data Pemakai
  - b. Layanan
    - Kunjungan Pasien
    - Pemberian Resep Obat
  - c. Laporan
    - Daftar Pasien
    - Daftar Dokter
    - Daftar Obat
    - Laporan Kunjungan Pasien
      - ❖ Daftar Kunjungan Pasien
      - ❖ Rekapitulasi Kunjungan Pasien
    - Laporan Resep Obat
      - ❖ Cetak Nota Resep
      - ❖ Daftar Pengeluaran Resep
  - d. Tambahan
    - Log Off
    - Informasi
    - BackUp dan Restore
    - Keluar Program

#### **Keterangan:**

- a. Antara pilihan Data Pasien dengan Data Dokter ada garis pembatas
- b. Antara pilihan Data Obat dengan Data Pemakai ada garis pembatas
- c. Pada pilihan Laporan Kunjungan Pasien dan pilihan Laporan Resep Obat ada sub menu
- d. Antara pilihan Daftar Obat dengan Laporan Kunjungan Pasien ada garis pembatas
- e. Antara pilihan Laporan Kunjungan Pasien dengan Laporan Resep Obat ada garis pembatas
- f. Antara pilihan Log Off dengan pilihan Informasi ada garis pembatas
- g. Antara pilihan BackUp dan Restore dengan Keluar Program ada garis pembatas
- h. Tulis kode program yang menghubungkan form password dengan form menu.
- i. Tulis juga kode program untuk menghubungkan form menu dengan form pengolahan data yang telah disediakan.

#### **Fasilitas ToolBar**

Untuk mempercantik tampilan menu buat juga ToolBar dan buat 6 (enam) tombol saja. Masing-masing tombol diberi tampilan gambar kecil (icon).

#### **Fasilitas Menu Layang**

Sebagai fasilitas pelengkap, perlu juga dibuat menu melayang dengan tiga pilihan yaitu : Data Pasien, Kunjungan Pasien, dan Pemberian Resep.

#### **Tampilan Jam dan Tampilan Tanggal**

Agar lebih lengkap, maka pada menu perlu ditampilkan Jam dan Tanggal. Jika perlu buat juga tampilan hari.

# MODUL VI

## Pemrograman Database Dengan BDE Dasar

### A. Pengantar

Aplikasi database menyediakan fasilitas bagi pengguna (user) untuk berinteraksi dengan informasi yang tersimpan dalam database. Database menyediakan struktur informasi yang dapat dipakai bersama oleh berbagai aplikasi (program).

Delphi mendukung pembuatan aplikasi menggunakan relational database. Relational database ini menyimpan informasi berupa tabel-tabel, yang berisi beberapa baris (records) dan kolom (fields), kemudian tabel-tabel ini dimanipulasi menggunakan operasi sederhana yang dikenal dengan relational calculus.

Ketika anda mendesain aplikasi database, anda harus mengerti betul bagaimana data tersebut diatur dan disimpan dalam database. Berdasarkan pengaturan tadi anda dapat mendesain tampilan pengguna (user interface) untuk menampilkan data ke pengguna dan menyediakan fasilitas untuk memasukan informasi baru atau memodifikasi data yang sudah ada, yang dikenal dengan entry data.

Pada dasarnya, Delphi menyediakan beberapa pendekatan untuk membangun aplikasi basis data (*database*), antara lain BDE (*Borland Database Engine*), ADO (*ActiveX Data Objects*), dan dbExpress. Pada modul ini hanya membahas BDE dan dilanjutkan pada modul VII tentang ADO karena fokus dari modul ini adalah pemrograman Windows. Komponen-komponen dbExpress digunakan untuk pemrograman basis data pada aplikasi CLX.

Baik BDE maupun ADO memiliki kelebihan dan kekurangan masing-masing. Secara umum, jika Anda ingin mengakses tabel Paradox atau dBase, gunakan BDE; jika Anda ingin mengakses tabel Access, gunakan ADO

### B. Koneksi ke Database

Ketika anda mendesain aplikasi database, anda harus mengerti betul bagaimana data tersebut diatur dan disimpan dalam database. Berdasarkan pengaturan tadi anda dapat mendesain tampilan pengguna (user interface) untuk menampilkan data ke pengguna dan menyediakan fasilitas untuk memasukan informasi baru atau memodifikasi data yang sudah ada, yang dikenal dengan entry data.

Setiap dataset yang terhubung ke server database menggunakan jenis komponen koneksinya masing-masing, yang di desain untuk bekerja pada sebuah mekanisme akses data (single data mechanism). Berikut ini daftar beberapa mekanisme akses data beserta komponen koneksinya

Database connection components

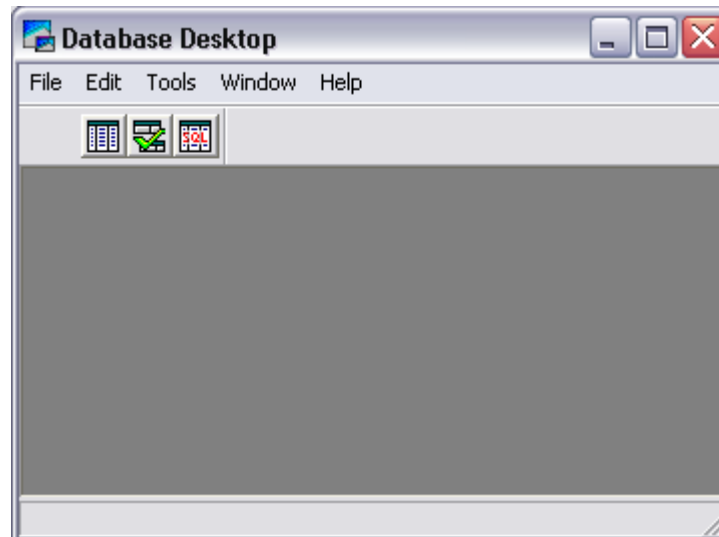
Data access mechanism	Connection
Borland Database Engine	TDatabase
ActiveX Data Objects (ADO)	TADOConnection
dbExpress	TSQLConnection
InterBase Express	TIBDatabase

### C. Praktek

#### 1. Membuat Struktur Data (Tabel)

Untuk membuat struktur data, khususnya database Desktop dengan Delphi dapat dilakukan dengan beberapa cara. Cara pertama dapat dipanggil langsung (klik menu Star, All Programs, Borland Delphi 7, Database Desktop). Cara kedua adalah dengan memanggil atau menjalankan Borland Delphi terlebih dahulu.

- Buka program Borland Delphi 7
- Pilih perintah menu Tools – Database Desktop, sehingga akan ditampilkan kotak dialog seperti Gambar 6.1



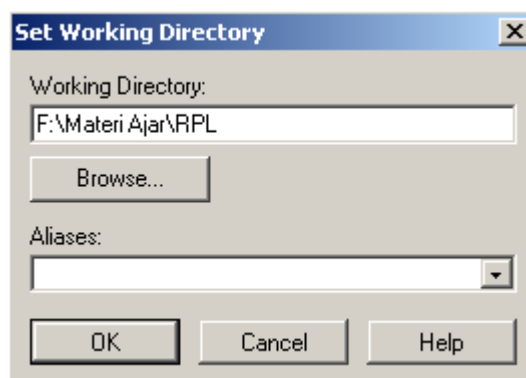
Gambar 6.1 Tampilan Database Desktop

Pada kotak dialog seperti Gambar 6.1, telah disediakan beberapa menu yang dapat digunakan untuk berbagai keperluan, diantaranya mengatur direktori kerja, membuat desain tabel dan lain sebagainya.

##### a. Direktori Kerja

Suatu direktori kerja *Database Desktop* adalah sebuah direktori yang digunakan sebagai direktori default apabila Anda menjalankan perintah membuka file (*open*) dan menyimpan file (*save*). Sebaiknya sebelum melakukan proses pengelolaan tabel, tahapan ini dilakukan terlebih dahulu.

- Pilih perintah menu File – Working Directory sehingga ditampilkan kotak dialog seperti Gambar 6.2



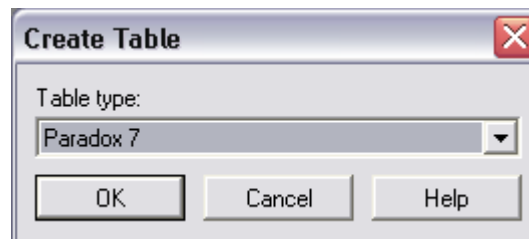
Gambar 6.2 Tampilan Directory Kerja

- Pada kotak dialog *Set Working Directory*, ketik urutan direktori yang dikehendaki, atau dapat juga menggunakan tombol Browse untuk memilih direktori yang diinginkan (sesuaikan dengan direktori Anda).
- Klik OK untuk mengakhiri perintah.

b. Mendesain Tabel

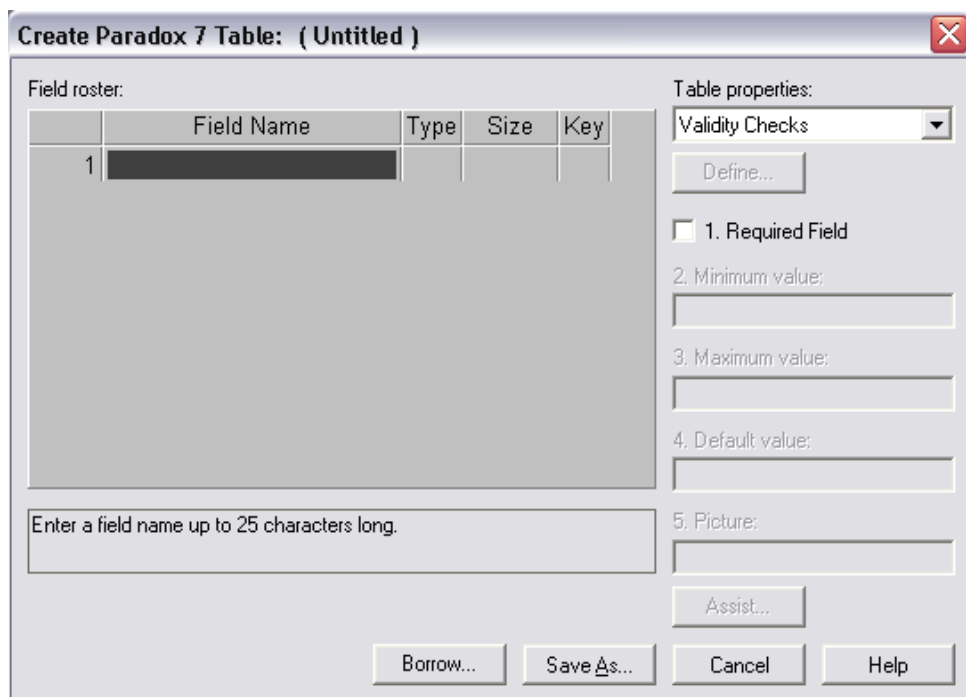
Untuk membuat sebuah tabel Paradox baru dari jendela Database Desktop lakukan langkah-langkah sebagai berikut:

- 1) Pilih perintah menu File – New – Table atau klik kanan tombol Open Table Toolbar, dan pilih New. Database Desktop akan membuka kotak dialog Create Table seperti terlihat pada Gambar 6.3.



Gambar 6.3 Tampilan Kotak Dialog Create Tabel

- 2) Jika Anda menginginkan tipe tabel yang lain selain Paradox 7.0, klik anak panah Table type dan pilih satu dari daftar drop-down yang diberikan, dan klik OK. Database Desktop akan menampilkan kotak dialog Create Paradox 7 Table, seperti terlihat pada Gambar 6.4. Dalam kondisi berada di kotak dialog ini Anda sudah dapat menentukan struktur tabel baru. Ada empat kolom yang bisa ditentukan yaitu: kolom untuk penentuan nama field, kolom penentuan jenis field, kolom penentuan ukuran field dan kolom menentukan apakah sebuah field dijadikan kunci atau tidak.



Gambar 6.4 Tampilan Create Paradox Table

- 3) Ketik nama field pertama dalam kolom *Field Name* pada bagian *Field Roster*. Buatlah nama field tidak terlalu panjang dan mudah diingat.
- 4) Pindah ke kolom type. Untuk berpindah antar kolom pada bagian *Field Roster* gunakan tombol Tab, Shift+Tab, atau Enter, atau dengan menggunakan tombol anak panah atau mouse. *Database Desktop* secara otomatis melompati beberapa kontrol secara bergantian.
- 5) Tekan SpaceBar atau klik kanan kontrol Type untuk menampilkan daftar tipe field. Ketik atau pilih simbol untuk tipe field yang diinginkan. Misalkan field akan diisi data karakter, maka pilihan yang tepat adalah Alpha (A). Apabila field akan diisi data angka, pilihan yang tepat adalah Numeric (N). Jika field akan diisi data jenis tanggal, pilihan yang tepat adalah Date (D). Setelah menentukan type field, maka beberapa isian yang terletak pada bagian kanan akan ditaktifkan. Anda dapat mengisi atau membiarkan saja.
- 6) Pindah ke kolom Size dan ketik suatu ukuran field yang sesuai. Apabila suatu field akan dijadikan kunci, maka pada kolom Key dapat ditandai dengan menekan tombol spasi sehingga akan diberi karakter tanda bintang (\*). Untuk menghilangkan tanda dapat dilakukan dengan menekan tombol spasi lagi.
- 7) Tekan anak panah bawah untuk menambahkan field yang lain. Ulangi langkah di atas sampai Anda menentukan beberapa field, seperti tampak pada Gambar 6.5.
- 8) Jika Anda menghendaki, tentukan suatu field untuk dibuat indeks (hanya dalam tabel Paradox) dan atur properti tabel. Caranya adalah klik combo pada Table properties dan pilih Secondary Indexes. Selanjutnya klik tombol Define. Klik field yang akan dijadikan index dan dilanjutkan menekan tombol panah supaya field yang dipilih berpindah ke bagian kanan (misalnya diindex berdasarkan NMDOKTER, lihat Gambar 6.6), akhiri dengan mengklik tombol OK. Berikutnya ketikkan nama file index dan akhiri dengan mengklik tombol OK (misalnya diberi nama file index dengan NAMA, lihat Gambar 6.7).

Restructure Paradox 7 Table: DOKTER.db

Field roster:

	Field Name	Type	Size	Key
1	KDDOKTER	A	5	*
2	NMDOKTER	A	40	
3	ALMDOKTER	A	50	
4	TELPDOKTER	A	15	
5	KATEGORI	A	10	

Enter a field name up to 25 characters long.

Table properties:

Validity Checks

Define...

☐ 1. Required Field

2. Minimum value:

3. Maximum value:

4. Default value:

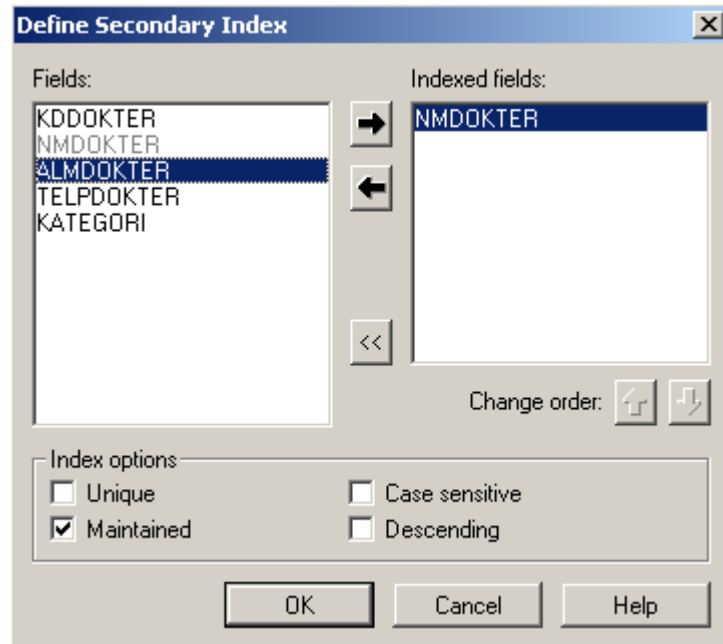
5. Picture:

☐ Pack Table

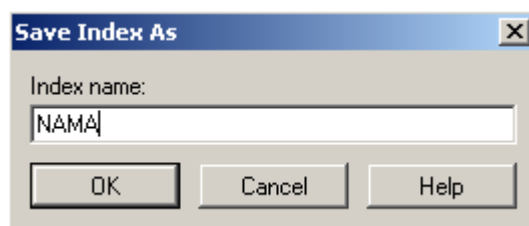
Assist...

Save Save As... Cancel Help

Gambar 6.5 Tampilan Proses Pembuatan Struktur Table Dokter

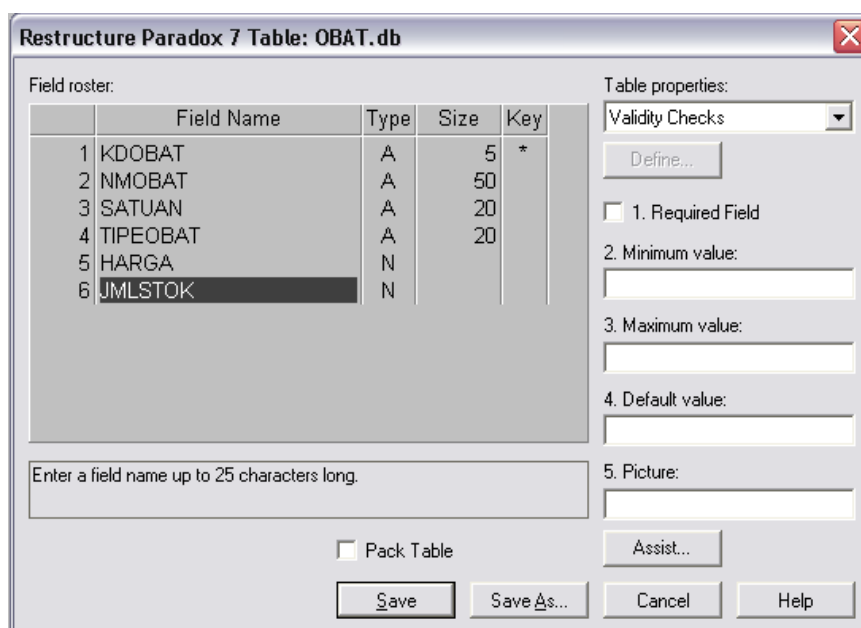


Gambar 6.6 Tampilan Proses Pembuatan Index



Gambar 6.7 Tampilan Pemberian Nama File Index

- 9) Ulangi langkah 1 sampai 8 untuk membuat tabel yang lain seperti Gambar 6.8.

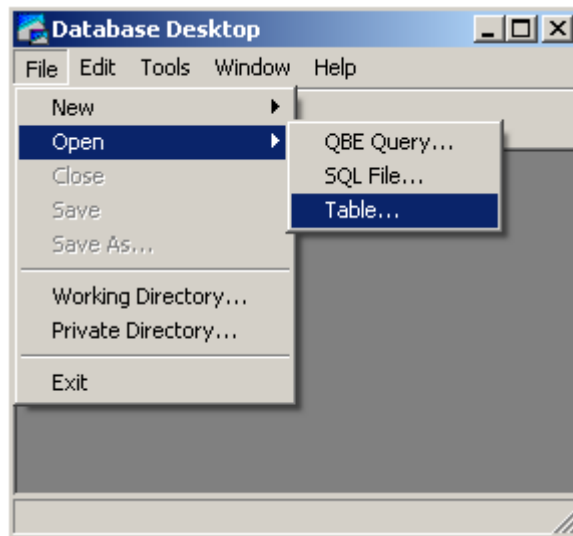


Gambar 6.8 Tampilan Proses Pembuatan Struktur Table Obat

c. Membuka Tabel Melalui Alias Manager

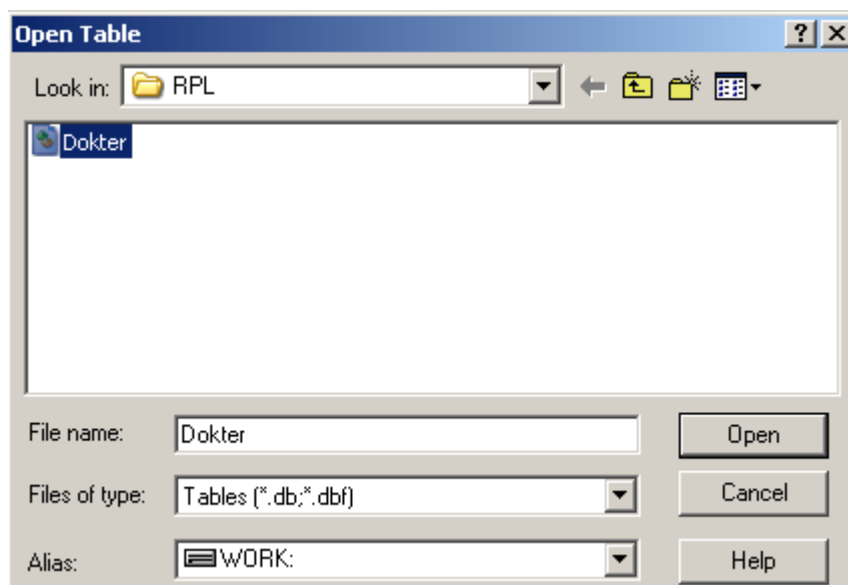
Dengan adanya Alias manager memberikan kemudahan kepada kita untuk melakukan pengelolaan terhadap tabel yang telah dibuat, misalnya membuka tabel. Langkah-langkah yang harus dilakukan untuk membuka tabel adalah sebagai berikut:

- Tampilkan menu File
- Pilih Open
- Pilih Table, seperti terlihat pada Gambar 6.9



Gambar 6.9 Tampilan Proses Membuka Table

Setelah mengklik menu Table, akan ditampilkan kotak dialog Open seperti terlihat pada Gambar 6.10.



Gambar 6.10 Tampilan Kotak Dialog Open

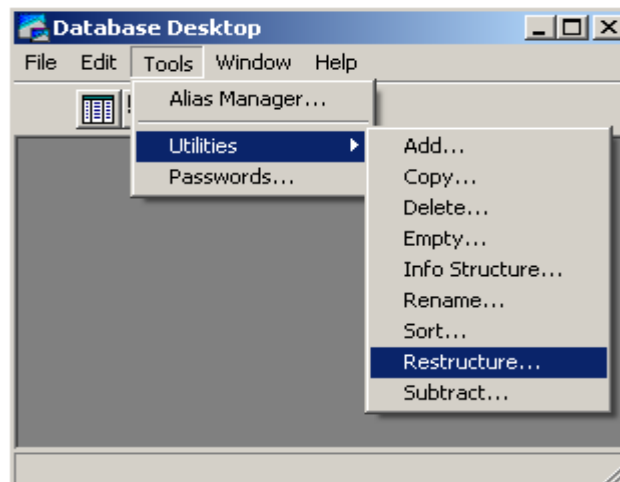


Pada saat ditampilkan kotak dialog Open, apabila directory kerja sudah benar, maka semua tabel yang pernah dibuat akan ditampilkan. Klik nama tabel yang akan dibuka, akhiri dengan mengklik tombol Open.

d. Mengubah Tabel

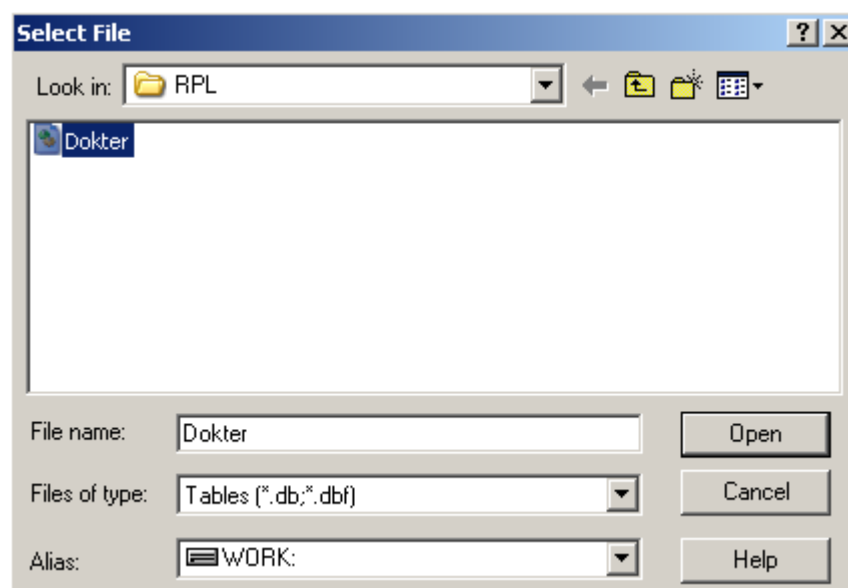
Suatu saat adakalanya tabel yang telah dibuat membutuhkan perbaikan. Misalkan perbaikan untuk merubah nama field, atau merubah tipe data atau ukuran field. Langkah-langkah untuk melakukan perubahan tabel adalah sebagai berikut:

- Tampilkan menu Tool
- Pilih menu Utilities
- Pilih Restructure, seperti Gambar 6.11.



Gambar 6.11 Tampilan Proses mengubah Tabel

Setelah mengklik menu Restructure, akan ditampilkan kotak dialog untuk memilih tabel yang akan diperbaiki. Klik nama tabel. Lanjutkan dengan mengklik tombol Open, lihat Gambar 6.12.

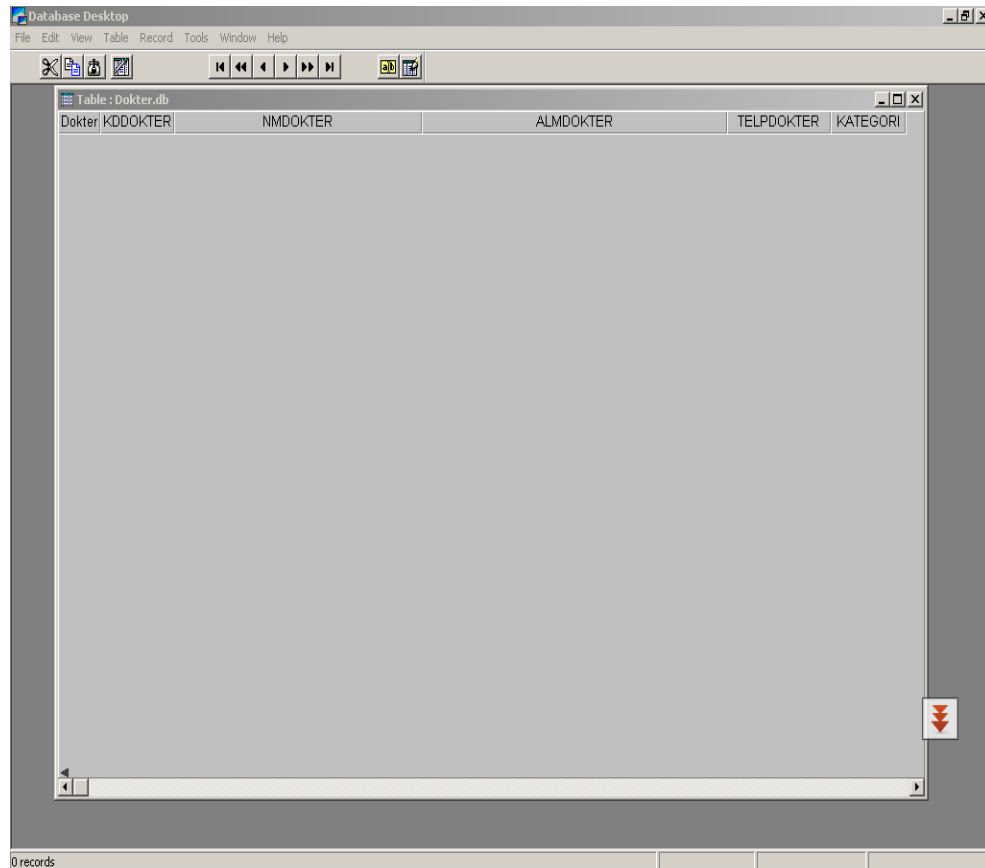


Gambar 6.12 Tampilan Kotak Dialog Memilih Tabel yang akan Diubah

e. Membuka dan Mengisi Tabel

Setelah tabel-tabel selesai dibuat, jika diperlukan tabel tersebut dapat diisi. Langkah-langkah untuk mengisi tabel adalah sebagai berikut:

- Tampilkan menu File
- Pilih Open
- Pilih Table
- Klik atau tentukan nama tabel yang akan dibuka, lanjutkan mengklik tombol Open



Gambar 6.13 Tampilan Kotak Dialog Open Tabel

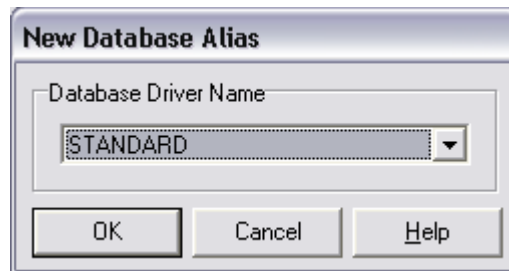
- Untuk mengisi data, tampilkan menu Table, klik menu Edit Data. Ketiklah data-data yang diinginkan. Untuk uji coba isilah minimal 5 buah data.

2. Membuat Alias Manager (BDE)

Alias manager berguna untuk memudahkan kita dalam menentukan letak suatu database atau sejumlah tabel-tabel yang sudah ada. Dengan adanya alias manager maka tabel-tabel tersebut dapat dikelompokkan menjadi satu tempat.

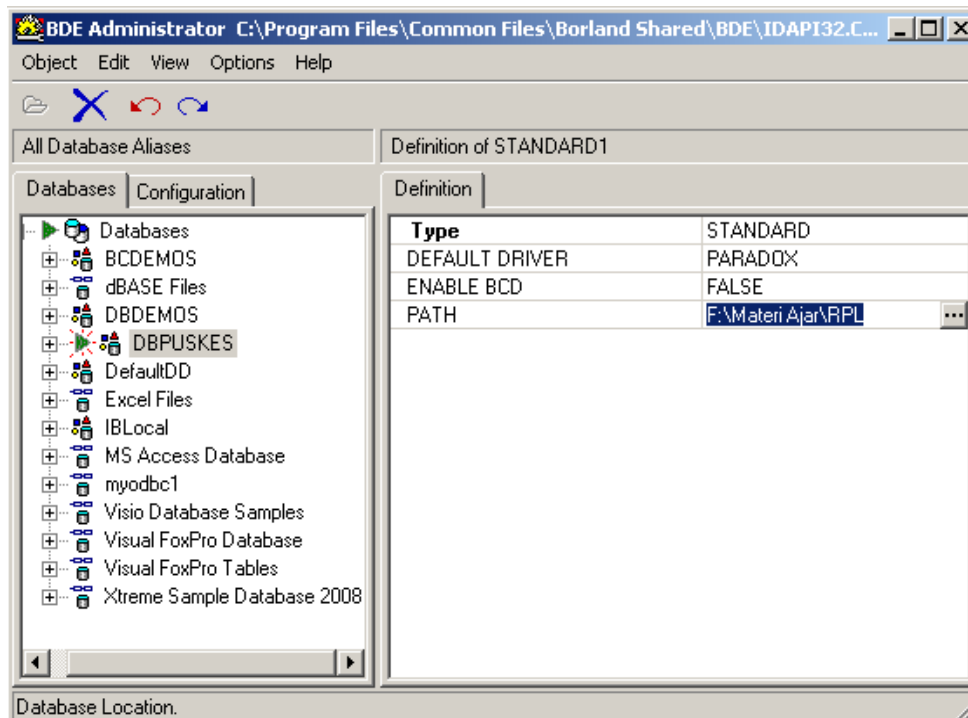
Adapun langkah-langkah membuat alias manager adalah sebagai berikut:

- Klik menu Start
- Pilih All Program
- Pilih Borland Delphi 7
- Pilih BDE Administrator
- Pilih menu Object : New. BDE Administrator akan menampilkan kotak dialog New Database Alias seperti Gambar 6.14.



Gambar 6.14 Tampilan Database Alias Manager BDE

- Klik Tombol OK, sehingga akan ditampilkan kotak dialog seperti Gambar 6.15



Gambar 6.15 Tampilan Proses Pembuatan Alias BDE

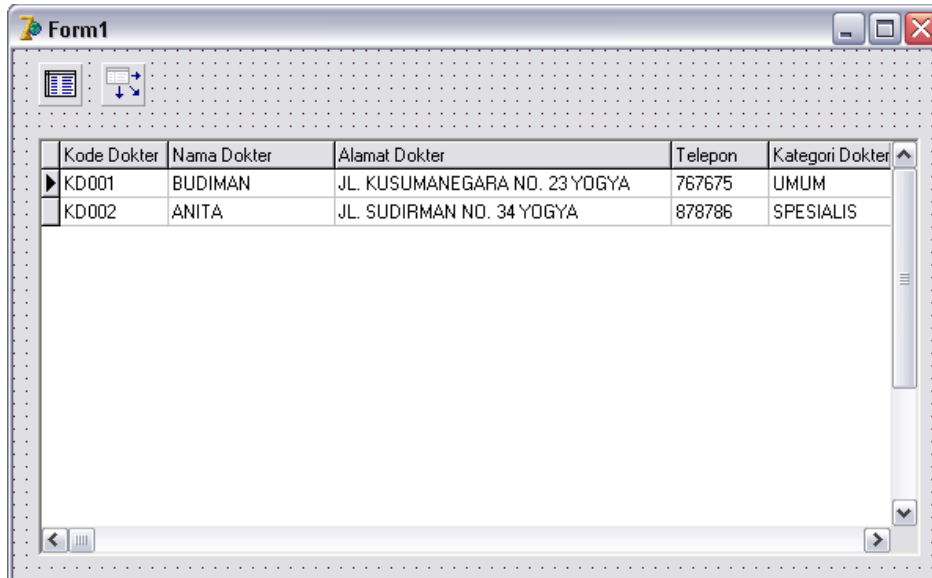
- Jika nama alias yang disajikan adalah Standard1 atau Standar2, dan seterusnya, maka gantilah dengan nama yang diinginkan, misalnya DBPUSKES atau nama lainnya yang Anda senangi.
- Langkah selanjutnya adalah menentukan direktori yang ditunjuk oleh alias tersebut. Caranya klik kotak edit disebelah PATH (lihat Gambar 6.15). Masukkan nama direktori pada kotak tersebut. Anda bisa juga memasukkan nama direktori dengan cara mengklik tombol '...' yang terletak pada bagian kanan. (Sesuaikan dimana data-data Anda akan diakses)
- Langkah terakhir pilih menu **Object – Apply** agar alias ini disimpan ke konfigurasi BDE.

### 3. Membuat Aplikasi Basis Data

- Form untuk menampilkan data dengan menggunakan Table, DataSource, dan DBGrid.
  - Buat proyek baru dan rancanglah form seperti Gambar 6.16
  - Letakkan komponen TTable pada Form (diambil pada tab page BDE). Isilah properti **DatabaseName** dengan nama direktori atau alias BDE. Isi properti **TableName**

dengan nama tabel (file dengan ekstensi .db atau .dbf). Kemudian ubahlah properti **Active** menjadi True.

- Letakkan komponen TdataSource pada Form (diambil pada tab page Data Access). Isilah properti **DataSet** dengan Table1.
- Letakkan komponen TDBGrid pada Form (diambil pada tab page Data Controls). Isilah properti **DataSource** dengan DataSource1.



The screenshot shows a Windows form titled "Form1" with a standard Windows XP-style title bar. The form contains a data grid with five columns: "Kode Dokter", "Nama Dokter", "Alamat Dokter", "Telepon", and "Kategori Dokter". The grid displays two rows of data. The first row has values: KD001, BUDIMAN, JL. KUSUMANEGARA NO. 23 YOGYA, 767675, and UMUM. The second row has values: KD002, ANITA, JL. SUDIRMAN NO. 34 YOGYA, 878786, and SPESIALIS. The grid has a vertical scrollbar on the right and a horizontal scrollbar at the bottom. The form's client area has a dotted grid background.

Kode Dokter	Nama Dokter	Alamat Dokter	Telepon	Kategori Dokter
KD001	BUDIMAN	JL. KUSUMANEGARA NO. 23 YOGYA	767675	UMUM
KD002	ANITA	JL. SUDIRMAN NO. 34 YOGYA	878786	SPESIALIS

Gambar 6.16 Pembuatan Form Dokter Untuk Menampilkan Data

- Jalankan program (Run).

b. Form untuk menampilkan data dengan menggunakan Table, DataSource, DBGrid dengan tambahan DBEDIT dan tombol Navigator untuk menjelajahi Tabel.

- Tambahkan form baru : File – New- Form. Rancanglah form seperti Gambar 6.17.
- Letakkan komponen TTable pada Form (diambil pada tab page BDE). Isilah properti **DatabaseName** dengan nama direktori atau alias BDE. Isi properti **TableName** dengan nama tabel (file dengan ekstensi .db atau .dbf). Kemudian ubahlah properti **Active** menjadi True.
- Letakkan komponen TdataSource pada Form (diambil pada tab page Data Access). Isilah properti **DataSet** dengan Table1.
- Letakkan komponen TDBGrid pada Form (diambil pada tab page Data Controls). Isilah properti **DataSource** dengan DataSource1.
- Letakkan komponen TDBNavigator pada form. Isilah properti **DataSource** dengan DataSource1.
- Letakkan lima komponen Label dan atur properti caption berturut-turut dari atas mulai dari Label1 sampai Label 5 : Kode Dokter, Nama Dokter, Alamat, Telp, dan Kategori.
- Letakkan lima komponen TDBEdit (diambil pada tab Data Controls) . Properti yang perlu dirubah pada komponen ini adalah properti DataSource diisi dengan DataSource1. Setelah itu rubah juga properti DataField diisikan sesuai dengan Field yang akan ditampilkan pada komponen tersebut.

The screenshot shows a Delphi form titled 'Form2' with a grid background. At the top, there is a set of navigation buttons (first, previous, next, last, insert, delete, undo, redo). Below these are five labels and their corresponding text boxes: 'Kode Dokter' (KD001), 'Nama Dokter' (BUDIMAN), 'Alamat' (JL. KUSUMANEGARA NO. 23 YOGYA), 'Telp' (767675), and 'Kategori' (UMUM). At the bottom of the form is a TDBGrid component displaying a table with five columns: 'Kode Dokter', 'Nama Dokter', 'Alamat Dokter', 'Telepon', and 'Kategori Dokter'. The table contains two rows of data.

Kode Dokter	Nama Dokter	Alamat Dokter	Telepon	Kategori Dokter
KD001	BUDIMAN	JL. KUSUMANEGARA NO. 23 YOGYA	767675	UMUM
KD002	ANITA	JL. SUDIRMAN NO. 34 YOGYA	878786	SPELIALIS

Gambar 6.17 Pembuatan Form Dokter Untuk Pengolahan Data

- Jalankan program (Run).

- c. Form untuk menampilkan data dengan menggunakan Table, DataSource, dan DBGrid dengan tambahan komponen TEdit, tombol buatan sendiri untuk menjelajahi Tabel.
- Tambahkan form baru : **File – New - Form**. Rancanglah form seperti Gambar 6.18
  - Letakkan komponen TTable pada Form. Isilah properti DatabaseName dengan nama direktori atau alias BDE. Isi properti TableName dengan nama tabel (file dengan ekstensi .db atau .dbf). Kemudian ubahlah properti Active menjadi True.
  - Letakkan komponen TdataSource pada Form. Isilah properti DataSet dengan Table1.
  - Letakkan komponen TDBGrid pada Form. Isilah properti DataSource dengan DataSource1.
  - Letakkan komponen empat komponen Button dan atur properti Caption berturut-turut mulai dari Button1 sampai Button2 yaitu : Pertama untuk Button1, Terakhir untuk Button2, Sebelum untuk Button3, dan Berikut untuk Button4.
  - Letakkan lima komponen Label dan atur properti caption berturut-turut dari atas mulai dari Label1 sampai Label 5 : Kode Dokter, Nama Dokter, Alamat, Telp, dan Kategori.
  - Letakkan lima komponen TEdit (diambil pada tab Standard) .

Kode Dokter	Nama Dokter	Alamat Dokter	Telepon	Kategori Dokter
KD001	BUDIMAN	JL. KUSUMANEGARA NO. 23 YOGYA	767675	Umum
KD002	ANITA	JL. SUDIRMAN NO. 34 YOGYA	878786	Spesialis
KD003	HERMAWAN	JL. KUSUMANEGARA NO. 32 YOGYA	534534	Umum

Gambar 6.18 Pembuatan Form Dokter Untuk Pengolahan Data

- Buatlah sebuah prosedur untuk menampilkan data. Jika saat ini Anda berada pada tampilan form, maka tekan tombol fungsi F12 untuk masuk ke Code Editor. Tempatkan kursor di antara **Implementation** dan **End**. pada baris kosong, ketik prosedur sebagai berikut:

```

procedure tampildata;
begin
form3.Edit1.text:=form3.Table1.Fields[0].value;
form3.Edit2.text:=form3.Table1.Fields[1].value;
form3.Edit3.text:=form3.Table1.Fields[2].value;
form3.Edit4.text:=form3.Table1.Fields[3].value;
form3.Edit5.text:=form3.Table1.Fields[4].value;
end;

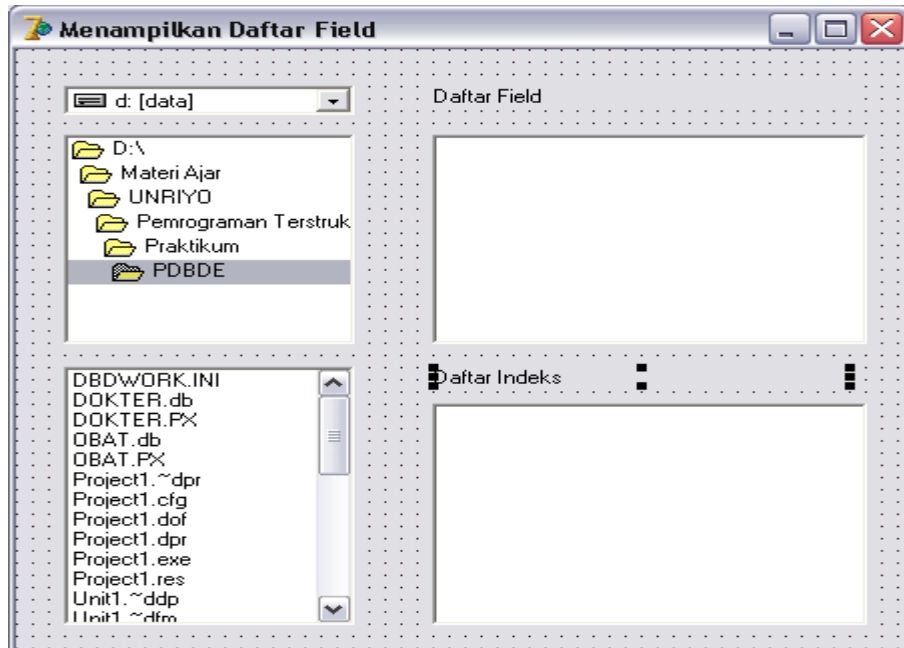
```

- Buat kode program pada kejadian tombol **Pertama** diklik. Jika saat ini Anda berada pada tampilan Code Editor, maka tekan tombol fungsi F12 untuk masuk ke tampilan form. Klik tombol Button1, lalu klik tab Events, klik dua kali pada baris OnClick dan tulis perintah sebagai berikut:  
`table1.first;  
tampildata;`
- Buat kode program pada kejadian tombol **Terakhir** diklik. Jika saat ini Anda berada pada tampilan Code Editor, maka tekan tombol fungsi F12 untuk masuk ke tampilan form. Klik tombol Button2, lalu klik tab Events, klik dua kali pada baris OnClick dan tulis perintah sebagai berikut:  
`table1.last;  
tampildata;`
- Buat kode program pada kejadian tombol **Sebelum** diklik. Jika saat ini Anda berada pada tampilan Code Editor, maka tekan tombol fungsi F12 untuk masuk ke tampilan form. Klik tombol Button3, lalu klik tab Events, klik dua kali pada baris OnClick dan tulis perintah sebagai berikut:  
`table1.prior;  
tampildata;`
- Buat kode program pada kejadian tombol **Berikut** diklik. Jika saat ini Anda berada pada tampilan Code Editor, maka tekan tombol fungsi F12 untuk masuk ke tampilan form. Klik tombol Button4, lalu klik tab Events, klik dua kali pada baris OnClick dan tulis perintah sebagai berikut:  
`table1.Next;  
tampildata;`
- Buat kode program untuk menangani kejadian saat form dipanggil. Jika saat ini Anda berada pada tampilan Code Editor, maka tekan tombol fungsi F12 untuk masuk ke tampilan form. Klik tempat kosong pada form, lalu klik tab Events, klik dua kali pada baris OnCreate dan tulis perintah sebagai berikut:

`tampildata;`

d. Form untuk menampilkan daftar field dan indeks yang terdapat pada sebuah tabel.

- Tambahkan form baru : File – New - Form. Rancanglah form seperti Gambar 6.19
- Ambil komponen DriveCombobox1, DirectoryListBox1, FileListBox1 (diambil pada tab Win 3.1)
- Ambil dua buah komponen label : Label1 dan Label2 dan dua buah komponen Listbox : ListBox1 dan ListBox2 (diambil pada tab Standard)
- Tambahkan komponen TTable
- Atur properti masing-masing komponen sebagai berikut :  
Caption Form diubah menjadi Menampilkan Daftar Field  
Caption Label1 diubah menjadi Daftar Field  
Caption Label2 diubah menjadi Daftar Indeks  
DirList DriveComboBox1 diubah menjadi DirectoryListBox1  
FileList dari DirectoryListBox1 diubah menjadi FileListBox1  
Mask dari FileListBox1 diubah menjadi \*.DB;\*.DBF



Gambar 6.19 Pembuatan Form Menampilkan Daftar Field dan Index

- Buat kode program untuk kejadian Form Create sebagai berikut :

-

```
DriveComboBox1.DirList:=DirectoryListBox1;  
DirectoryListBox1.FileList:=FileListBox1;  
FileListBox1.Mask:= "*.DB;*.DBF";  
Table1:=nil;
```

- Buat kode di bawah pernyataan Private  
Table1: TTable;

- Buat kode program untuk kejadian FileListBox1 OnClick (yang awali tanda garis miring dua adalah keterangan program, tidak ditulis juga tidak apa-apa) sebagai berikut :

```
//mengalokasikan memori untuk tabel  
Table1:=TTable.Create(Self);  
// Inisialisasi tabel;  
Table1.TableName:=FileListBox1.FileName;  
// Membuka tabel  
Table1.Active:=true;  
// Menampilkan daftar field dan indeks  
Listbox1.Items.Clear;  
Listbox2.Items.Clear;  
Table1.GetFieldNames(Listbox1.Items);  
Table1.GetIndexNames(Listbox2.Items);  
// Menutup tabel  
Table1.Active:=False;  
// Mendefinisikan tabel  
Table1.Free;
```



#### D. Soal-Soal

1. Manfaat apa yang Anda dapatkan saat bekerja dengan database khususnya BDE?
2. Tambahkan lah sebuah tabel ke directory kerja yang telah Anda tetapkan pada pekerjaan di atas. Tabel yang perlu dibuat adalah tabel untuk data Pasien.  
Rincian field-fieldnya terdiri dari:
  - KDPASIEN, jenis data Alpha dengan lebar 7
  - NAMA, jenis data Alpha dengan lebar 40
  - KELAMIN, jenis data Alpha dengan lebar 10
  - ALAMAT, jenis data Alpha dengan lebar 50
  - TELP, jenis data Alpha dengan lebar 15Catatan:
  - KDPASIEN dijadikan primary key
  - Perlu juga dibuat sebuah index yang diurutkan berdasarkan NAMA
3. Isilah data pasien minimal 5 data
4. Buatlah form untuk menampilkan data pasien tersebut

## MODUL VII

### Pemrograman Database Dengan BDE Lanjut

#### A. Pengantar

Pada modul ini merupakan lanjutan dari proses pemrograman database dengan BDE. Pada modul sebelumnya masih terbatas pada proses pembuatan database, membuat tabel, merubah tabel, mengisi data pada tabel dan membuat form untuk menampilkan data.

Pada modul ini akan dilanjutkan dengan konsep pengolahan data. Maksudnya adalah data bisa dimasukkan melalui form dengan memanfaatkan komponen-komponen yang sesuai. Aspek yang dikerjakan pada pengolahan data yaitu mengisi data, menyimpan data, merubah data, dan menghapus data.

#### B. Praktek

Sebelum memulai praktek untuk modul ini, terlebih dahulu siapkan folder untuk menyimpan pekerjaan pada modul ini. Misalnya disiapkan folder dengan nama SESI-7. Selanjutnya lakukan proses pengcopyan terhadap database yang sudah dibuat pada SESI-6 (yang dicopy semua tabel).

1. Form untuk memanipulasi data secara sederhana.
  - Buat proyek baru dan rancanglah form seperti Gambar 7.1.
  - Letakkan komponen TTable pada Form (diambil pada tab page BDE). Isilah properti **DatabaseName** dengan nama direktori atau alias BDE. Isi properti **TableName** dengan nama tabel (file dengan ekstensi .db atau .dbf). Kemudian ubahlah properti **Active** menjadi True.
  - Letakkan komponen TdataSource pada Form (diambil pada tab page Data Access). Isilah properti **DataSet** dengan Table1.
  - Letakkan komponen TDBGrid pada Form (diambil pada tab page Data Controls). Isilah properti **DataSource** dengan DataSource1.

The screenshot shows a Delphi form titled 'Form1' with a grid background. On the left, there are five labeled text boxes: 'Kode Dokter', 'Nama Dokter', 'Alamat', 'Telepon', and 'Kategori' (with a dropdown arrow). Below these are five buttons: 'Tambah', 'Batal', 'Edit', 'Hapus', and a larger 'Simpan' button. On the right, there is a TDBGrid component displaying a table with three columns: 'Kode Dokter', 'Nama Dokter', and 'Alamat Do'. The table contains three rows of data.

Kode Dokter	Nama Dokter	Alamat Do
KD001	BUDIMAN	JL. KUSUM
KD002	ANITA	JL. SUDIR
KD003	HERMAWAN	JL. KUSUM

Gambar 7.1 Rancangan Form Pengolahan Data Dokter

- Letakkan 5 komponen Label pada Form (diambil pada tab page Standard). Ubahlah **Caption** masing-masingnya seperti terlihat pada gambar.
- Letakkan 4 komponen Edit, dan 1 komponen Combobox.
- Klik komponen Combobox1, lalu atur properti **Items** dan isilah ItemString untuk pilihan yaitu:

Umum  
Spesialis

- Letakkan 5 (lima) komponen Button pada Form. Ubahlah **Caption** masing-masing Button tersebut. Button1 menjadi &Tambah, Button2 menjadi &Batal, Button3 menjadi &Edit, Button4 menjadi &Hapus, dan Button5 menjadi &Simpan. Ubahlah **Name** masing-masing Button tersebut yaitu cmdtambah, cmdbatal, cmdedit, cmdhapus dan cmdsimpan.
- Buatlah beberapa prosedur yang diletakkan diantara **Implementation** dengan **End.**, (jika Anda berada pada tampilan form, untuk masuk ke Editor Code, tekan tombol fungsi F12).

```
procedure nilaiawal;
begin
  form1.Edit1.text:=' ';
  form1.Edit2.text:=' ';
  form1.Edit3.text:=' ';
  form1.Edit4.text:=' ';
  form1.Combobox1.text:=' ';
end;
```

```
procedure aturisian(kondisi:boolean);
begin
  form1.Edit1.enabled:=kondisi;
  form1.Edit2.enabled:=kondisi;
  form1.Edit3.enabled:=kondisi;
  form1.Edit4.enabled:=kondisi;
  form1.combobox1.enabled:=kondisi;
end;
```

```
procedure aturtombol(k1,k2,k3,k4,k5:boolean);
begin
  form1.cmdtambah.Enabled:=k1;
  form1.cmdbatal.Enabled:=k2;
  form1.cmdedit.Enabled:=k3;
  form1.cmdhapus.Enabled:=k4;
  form1.cmdsimpan.Enabled:=k5;
end;
```

```
procedure tampildata;
begin
  form1.Edit1.text:=form1.Table1.Fields[0].Value;
  form1.Edit2.text:=form1.Table1.Fields[1].Value;
  form1.Edit3.text:=form1.Table1.Fields[2].Value;
  form1.Edit4.text:=form1.Table1.Fields[3].Value;
  form1.combobox1.text:=form1.Table1.Fields[4].Value;
end;
```

- Buatlah kode program untuk masing-masing Button pada kejadian OnClick. Caranya, klik komponen Button yang bersangkutan, lalu masuk ke properties dan

aktifkan tab **Events** dan arahkan mouse pada baris **OnClick**, dan lakukan klik dua kali. Masing-masing kode adalah sebagai berikut:

\* Saat cmdtambah OnClick

```
nilaiawal;  
aturisian(true);  
aturtombol(false,true,false,false,true);  
form1.Edit1.SetFocus;
```

\* Saat cmdbatal OnClick

```
nilaiawal;  
aturtombol(true,false,false,false,false);  
aturisian(false);  
if form1.Table1.RecordCount>0 then  
  tampildata;  
  aturtombol(true,true,true,true,false);
```

\* Saat cmdedit OnClick

```
aturisian(true);  
aturtombol(false,true,false,false,true);  
form1.Edit1.SetFocus;
```

\* Saat cmdhapus OnClick

```
if messagedlg('Benar akan dihapus?',  
  mtconfirmation,[mbytes,mbno],0)=mryes then  
  table1.Delete;  
  nilaiawal;  
  aturtombol(true,false,false,false,false);  
  aturisian(false);  
  if form1.Table1.RecordCount>0 then  
    tampildata;  
    aturtombol(true,true,true,true,false);
```

\* Saat cmdsimpan OnClick

```
table1.Append;  
table1.FieldName('KDDOKTER').AsString:=Edit1.Text;  
table1.FieldName('NMDOKTER').AsString:=Edit2.Text;  
table1.FieldName('ALMDOKTER').AsString:=Edit3.Text;  
table1.FieldName('TELPDOKTER').AsString:=Edit4.Text;  
table1.FieldName('KATEGORI').AsString:=ComboBox1.Text;  
table1.Post;  
aturisian(false);  
aturtombol(true,false,true,true,false);
```

\* Saat DBGrid OnCellClick

```
tampildata;
```

\* Saat Form OnCreate

```
nilaiawal;
```

```

aturtombol(true,false,false,false,false);
aturisian(false);
if form1.Table1.RecordCount>0 then
tampildata;
aturtombol(true,false,true,true,false);

```

- Jalankan program (Run).

2. Form untuk memanipulasi data dilengkapi dengan proses pencarian data.

- Copy semua komponen yang ada pada form1 (Ctrl+A)
- Tambahkan form baru : File – New- Form
- Saat form baru telah ditampilkan, klik icon Paste atau Ctrl+V
- Tambahkan satu komponen Edit dan satu komponen Button dan letakkan di bawah DBGrid, sehingga rancangan form seperti Gambar 7.2.

Kode Dokter	Nama Dokter	Alamat Do
KD001	BUDIMAN	JL. KUSUM
KD002	ANITA	JL. SUDIR
KD003	HERMAWAN	JL. KUSUM

Gambar 7.2 Rancangan Form Pengolahan Data Dokter Dengan Tambahan

- Atur properti untuk Button yang ditambahkan, dimana **Caption** diganti menjadi Cek, dan Name diganti dengan cmdcari.
- Buatlah beberapa prosedur yang diletakkan diantara Implementation dengan End., (jika Anda berada pada tampilan form, untuk masuk ke Editor Code, tekan tombol fungsi F12). Karena Form2 ini secara umum sama seperti Form1, maka copy saja semua prosedur pada Form1 agar lebih cepat. Tapi perlu diingat, cara peng-copy harus hati-hati. Untuk prosedur umum seperti nilaiawal, aturtombol, aturisian, tampildata dapat dicopy sekaligus dengan merubah tulisan Form1 menjadi Form2. Sedangkan prosedur untuk setiap event masing-masing tombol dicopy satu persatu berdasarkan perintah yang ada diantara Begin dan End.
- Tulis kode program untuk proses pencarian. Klik pada cmdcek, lalu klik dua kali dan tulis perintah seperti kode di bawah ini.

```

table1.IndexName:='urutnama';
if (not table1.FindKey([edit5.text])) then
    messagedlg('Data yang dicari tidak ada..!',
        mtinformation,[mbok],0)
else
    tampildata;
end;

```

- Jalankan program (Run).

Catatan :

Pastikan tabel yang Anda pakai telah diindeks berdasarkan urutan tertentu, misalnya NMDOKTER dengan nama secondary indexes adalah urutnama. Jika belum lakukan dengan cara masuk ke Database Desktop, buka tabel yang akan dibuatkan urutan datanya. Lakukan modifikasi terhadap struktur tabel yang bersangkutan. Klik pada combo TableProperti, ganti dengan pilihan Secondary Indexes. Klik tombol Define. Pilih field apa yang akan diurutkan, lalu klik tombol panah sebagai langkah untuk memastikan field yang akan diurutkan. Akhiri dengan mengklik tombol OK. Isikan nama untuk menyimpan Secondary Indexes, misalnya URUTNAMA.

### 3. Form untuk memfilter data.

- Tambahkan Form baru ke proyek yang sedang dikerjakan saat ini dan rancanglah form seperti Gambar 7.3
- Letakkan komponen TTable pada Form. Isilah properti DatabaseName dengan nama direktori atau alias BDE. Isi properti TableName dengan nama tabel (file dengan ekstensi .db atau .dbf). Kemudian ubahlah properti Active menjadi True.
- Letakkan komponen TdataSource pada Form. Isilah properti DataSet dengan Table1.
- Letakkan komponen TDBGrid pada Form. Isilah properti DataSource dengan DataSource1.

The screenshot shows a Delphi form titled 'Form3'. Inside the form, there is a TTable component displaying a grid of data. The grid has five columns: 'Kode Dokter', 'Nama Dokter', 'Alamat Dokter', 'Telepon', and 'Katego'. The data rows are as follows:

Kode Dokter	Nama Dokter	Alamat Dokter	Telepon	Katego
KD001	BUDIMAN	JL. KUSUMANEGARA NO. 23 YOGYA	767675	Umum
KD002	ANITA	JL. SUDIRMAN NO. 34 YOGYA	878786	Spesiali
KD003	HERMAWAN	JL. KUSUMANEGARA NO. 32 YOGYA	534534	Umum

Gambar 7.3 Rancangan Form Penyaringan Data Dokter

- Klik komponen TTable, lalu tampilkan tab Events, cari OnFilterRecord dan klik dua kali.

- Saat sudah berada pada Code Editor, tulis perintah sebagai berikut:

**accept:=table1.FieldByName('KATEGORI').AsString='Umum';**

Filter di atas menampilkan semua dokter dengan kategori Umum. Yang perlu diingat, data hanya akan difilter setelah properti Filtered diubah menjadi true.

Untuk menampilkan keseluruhan data, ubahlah kembali Filtered menjadi false.

- Jalankan program (**Run**).

### **C. Soal-Soal**

1. Apa pendapat Anda setelah melakukan pengolahan data dengan database khususnya BDE?
2. Buatlah sebuah database dengan menggunakan BDE, kemudian ciptakan tabel-tabelnya. Misalnya database untuk sistem informasi perpustakaan. Tabel-tabel induk yang perlu diadakan diantaranya: tabel anggota, tabel buku, tabel penerbit, tabel pengarang. Tabel transaksi seperti peminjaman dan pengembalian tidak perlu dibuat.
3. Buatlah form untuk mengolahan data untuk setiap tabel yang Anda ciptakan.

## MODUL VIII

### Pemrograman Database Dengan BDE Lanjut

#### D. Pengantar

Pada modul ini masih lanjutan dari modul sebelumnya yaitu melengkapi pengolahan data yang lain. Jika pada modul sebelumnya masih berfokus pada pengolahan untuk data Dokter, maka pada modul ini akan dikerjakan untuk pengolahan data yang lain yaitu data obat, data pasien, dan data pemeriksaan.

#### E. Praktek

##### a. Persiapan

- i. Sebelum memulai praktek untuk modul ini, terlebih dahulu siapkan folder untuk menyimpan pekerjaan pada modul ini. Misalnya disiapkan folder dengan nama SESI-8.
- ii. Selanjutnya lakukan proses pengcopyan terhadap database yang sudah dibuat pada SESI-7 (yang dicopy semua tabel).
- iii. Masuk atau panggil Database Desktop.
- iv. Buat tambahan tabel baru untuk pengolahan data pasien, dengan field sebagai berikut:

Nama Field	Tipe Data	Lebar	Key
KDPASIEN	A	7	*
NAMA	A	30	
KELAMIN	A	1	
ALAMAT	A	50	
TELP	A	12	

- v. Field KDPASIEN dijadikan primary key
- vi. Simpan tabel dengan nama PASIEN
- vii. Isilah data pasien, minimal sebanyak 5 record (jika bingung contoh datanya lihat Gambar 8.3 pada grid bagian atas)
- viii. Buat satu tabel baru lagi untuk pengolahan data pemeriksaan, dengan field sebagai berikut:

Nama Field	Tipe Data	Lebar	Key
NOPERIKSA	A	10	*
TGLPERIKSA	D		
KDPASIEN	A	7	
KELUHAN	A	50	

- ix. Field NOPERIKSA dijadikan primary key
- x. Simpan tabel dengan nama PERIKSA
- xi. Isilah data pemeriksaan, minimal sebanyak 5 record (jika bingung contoh datanya lihat Gambar 8.3 pada grid bagian bawah)



**b. Form untuk mengolah data obat**

- Buat proyek baru dan rancanglah form seperti Gambar 8.1
- Letakkan komponen TTable pada Form (diambil pada tab page BDE). Isilah properti **DatabaseName** dengan nama direktori atau alias BDE. Isi properti **TableName** dengan nama tabel (file dengan ekstensi .db atau .dbf). Kemudian ubahlah properti **Active** menjadi True.
- Letakkan komponen TdataSource pada Form (diambil pada tab page Data Access). Isilah properti **DataSet** dengan Table1.
- Letakkan komponen TDBGrid pada Form (diambil pada tab page Data Controls). Isilah properti **DataSource** dengan DataSource1.

The image shows a Delphi form titled 'Form1'. It features a data grid on the right with two columns: 'KDOBAT' and 'NMOBAT'. On the left side of the form, there are several labels and corresponding input fields: 'Kode Obat' (text box), 'Nama Obat' (text box), 'Satuan' (dropdown menu), 'Tipe Obat' (dropdown menu), 'Harga' (text box), and 'Jumlah Stok' (text box). Below these input fields are five buttons: 'Tambah', 'Batal', 'Edit', 'Hapus', and 'Simpan'. At the bottom right, there is a 'Cek' button. The form is designed with a dotted background and standard Windows-style window controls at the top.

Gambar 8.1 Rancangan Form Pengolahan Data Obat

- Letakkan 6 komponen **Label** pada Form. Ubahlah **Caption** masing-masingnya seperti terlihat pada Gambar 8.1.
- Letakkan 4 komponen **Edit** (untuk kode obat, nama obat, harga dan jumlah stok), dan 2 komponen **Combobox** (untuk satuan dan tipe obat).
- Klik komponen Combobox1, lalu atur properti **Items** dan isilah ItemString untuk pilihan yaitu:  
**Botol**  
**Kaplet**  
**Butir**  
**Kapsul**  
**Pak**
- Klik komponen Combobox2, lalu atur properti **Items** dan isilah ItemString untuk pilihan yaitu:  
**Generik**  
**Non Generik**
- Letakkan 5 (lima) komponen Button pada Form 5. Ubahlah **Caption** masing-masing Button terbut. **Button1** menjadi **&Tambah**, **Button2** menjadi **&Batal**, **Button3** menjadi **&Edit**, **Button4** menjadi **&Hapus**, dan **Button5** menjadi **&Simpan**. Ubah juga Name masing-masing Button tersebut. **Button1** menjadi **cmdtambah**, **Button2**

menjadi **cmdbatal**, **Button3** menjadi **cmdedit**, **Button4** menjadi **cmdhapus**, dan **Button5** menjadi **cmdsimpan**

- Tambahkan satu komponen Edit dan satu komponen Button dan letakkan di bawah DBGrid, sehingga rancangan form seperti Gambar 8.1
- Atur properti untuk Button yang ditambahkan, dimana **Caption** diganti menjadi &Cek, dan **Name** diganti dengan cmdcari
- Buatlah beberapa prosedur yang diletakkan diantara **Implementation** dengan **End.**, (jika Anda berada pada tampilan form, untuk masuk ke Editor Code, tekan tombol fungsi F12).

```
procedure nilaiawal;
begin
  form1.Edit1.text:=' ';
  form1.Edit2.text:=' ';
  form1.Combobox1.text:=' ';
  form1.Combobox2.text:=' ';
  form1.Edit3.text:=' ';
  form1.Edit4.text:=' ';
end;
```

```
procedure aturisian(kondisi:boolean);
begin
  form1.Edit1.enabled:=kondisi;
  form1.Edit2.enabled:=kondisi;
  form1.combobox1.enabled:=kondisi;
  form1.combobox2.enabled:=kondisi;
  form1.Edit3.enabled:=kondisi;
  form1.Edit4.enabled:=kondisi;
end;
```

```
procedure aturtombol(k1,k2,k3,k4,k5:boolean);
begin
  form1.cmdtambah.Enabled:=k1;
  form1.cmdbatal.Enabled:=k2;
  form1.cmdedit.Enabled:=k3;
  form1.cmdhapus.Enabled:=k4;
  form1.cmdsimpan.Enabled:=k5;
end;
```

```
procedure tampildata;
begin
  form1.Edit1.text:=form1.Table1.Fields[0].Value;
  form1.Edit2.text:=form1.Table1.Fields[1].Value;
  Form1.combobox1.text:=form1.Table1.Fields[2].Value;
  Form1.combobox2.text:=form1.Table1.Fields[3].Value;
  form1.Edit3.text:=form1.Table1.Fields[4].Value;
  form1.Edit4.text:=form1.Table1.Fields[5].Value;
end;
```

- Buatlah kode program untuk masing-masing Button pada kejadian OnClick. Caranya, klik komponen Button yang bersangkutan, lalu masuk ke object inspector/properties dan aktifkan tab Events dan arahkan mouse pada baris **OnClick**, dan lakukan klik dua kali. Masing-masing kode adalah sebagai berikut:

\* Saat cmdtambah OnClick

```
nilaiawal;  
aturisian(true);  
aturtombol(false,true,false,false,true);  
form1.Edit1.SetFocus;
```

\* Saat cmdbatal OnClick

```
nilaiawal;  
aturtombol(true,false,false,false,false);  
aturisian(false);  
if form1.Table1.RecordCount>0 then  
  tampildata;  
  aturtombol(true,true,true,true,false);
```

\* Saat cmdedit OnClick

```
aturisian(true);  
aturtombol(false,true,false,false,true);  
form1.Edit1.SetFocus;
```

\* Saat cmdhapus OnClick

```
if messagedlg('Benar akan dihapus?',  
  mtconfirmation,[mbyes,mbno],0)=mryes then  
  table1.Delete;  
  nilaiawal;  
  aturtombol(true,false,false,false,false);  
  aturisian(false);  
  if form1.Table1.RecordCount>0 then  
    tampildata;  
    aturtombol(true,true,true,true,false);
```

\* Saat cmdsimpan OnClick

```
table1.IndexName:="";  
if (not table1.FindKey([edit1.text])) then  
  begin  
    table1.Append;  
    table1.FieldName('KDOBAT').AsString:=Edit1.Text;  
    table1.FieldName('NMOBAT').AsString:=Edit2.Text;  
    table1.FieldName('SATUAN').AsString:=Combobox1.Text;  
    table1.FieldName('TIPEOBAT').AsString:=Combobox2.Text;  
    table1.FieldName('HARGA').AsString:=edit3.Text;  
    table1.FieldName('JMLSTOK').AsString:=edit4.Text;  
    table1.Post;  
  end  
else  
  if messagedlg('Data ini sudah ada, mau diperbarui?',  
    mtconfirmation,[mbyes,mbno],0)=mryes then  
    begin  
      table1.edit;  
      table1.FieldName('KDOBAT').AsString:=Edit1.Text;  
      table1.FieldName('NMOBAT').AsString:=Edit2.Text;  
      table1.FieldName('SATUAN').AsString:=Combobox1.Text;  
      table1.FieldName('TIPEOBAT').AsString:=Combobox2.Text;
```

```

table1.FieldName('HARGA').AsString:=edit3.Text;
table1.FieldName('JMLSTOK').AsString:=edit4.Text;
table1.Post;
end;

```

```

aturisian(false);
aturtombol(true,false,true,true,false);

```

\* Saat cmdcari OnClick

```

table1.IndexName:='urutnama';
if (not table1.FindKey([edit5.text])) then
  messagedlg('Data yang dicari tidak ada..!',
    mtinformation,[mbok],0)
else
  tampildata;
end;

```

\* Saat DBGrid OnCellClick

```

tampildata;

```

\* Saat Form OnCreate

```

nilaiawal;
aturtombol(true,false,false,false,false);
aturisian(false);
if form1.Table1.RecordCount>0 then
  tampildata;
aturtombol(true,false,true,true,false);

```

- Jalankan program (Run).

Catatan :

Pastikan tabel yang Anda pakai telah diindeks berdasarkan urutan tertentu, misalnya NMOBAT dengan nama secondary indexes adalah urutnama. Jika belum lakukan dengan cara masuk ke Database Desktop, buka tabel yang akan dibuatkan urutan datanya. Lakukan modifikasi terhadap struktur tabel yang bersangkutan. Klik pada combo TableProperti, ganti dengan pilihan Secondary Indexes. Klik tombol Define. Pilih field apa yang akan diurutkan, lalu klik tombol panah sebagai langkah untuk memastikan field yang akan diurutkan. Akhiri dengan mengklik tombol OK. Isikan nama untuk menyimpan Secondary Indexes, misalnya URUTNAMA.

### c. Form untuk mengolah data pasien.

Bentuk interface pengolahan data pengolahan pasien hampir sama dengan pengolahan data obat, hanya ada beberapa obyek yang berbeda. Untuk keperluan latihan praktikum yang kedua ini, Anda dapat melakukan copy obyek pengolahan data obat lalu lakukan penyesuaian. (Tapi jika takut keliru sebaiknya dirancang dari awal, dengan demikian akan memperlancar dalam penggunaan komponen dan pengisian properties). Jika Anda memilih cara pertama, maka ikuti langkah-langkah sebagai berikut:

- Copy semua komponen yang ada pada form1 atau form pengolahan data obat (dengan perintah : Ctrl+A)
- Tambahkan form baru : File – New- Form
- Saat form baru telah ditampilkan, sesuaikan dulu ukuran form kemudian klik icon Paste atau Ctrl+V

- Lakukan penyesuaian obyek-obyek yang dibutuhkan. Rubah properti Caption untuk masing-masing Label, dan hapus obyek yang tidak diperlukan, sehingga bentuk Form dapat terlihat seperti Gambar 8.2.
- Lakukan pengaturan properties untuk Ttable dan Tdatasource. Klik komponen Ttable lalu isilah properti **DatabaseName** dengan nama direktori atau alias BDE. Isi properti **TableName** dengan nama tabel (file dengan ekstensi .db atau .dbf). Kemudian ubahlah properti **Active** menjadi True. Untuk Tdatasource isilah properti **DataSet** dengan Table1

Gambar 8.2 Rancangan Form Pengolahan Data Pasien

- Buatlah beberapa prosedur yang diletakkan diantara Implementation dengan End., (jika Anda berada pada tampilan form, untuk masuk ke Editor Code, tekan tombol fungsi F12). Karena prinsip kerja pengolahan data pasien dengan pengolahan data obat hampir sama, maka Anda dapat juga melakukan peng-copy-an prosedur. Tapi harus hati-hati dalam proses peng-copy-an. Setelah dicopy jangan lupa untuk disesuaikan. Namun jika Anda takut keliru sebaiknya dilakukan proses pembuatan seperti pada tahap pembuatan prosedur pengolahan data obat.
  - Jika sudah selesai, jalankan program (Run). Jika masih terdapat kesalahan lakukan perbaikan sampai tidak ada error lagi. Pengolahan data pasien ini jadikan sebagai ajang uji coba, pemahaman Anda terhadap materi-materi sebelumnya.
- d. Form untuk menggabung tabel.
- Tambahkan form baru : File – New- Form
  - Letakkan 1 komponen Database pada Form
  - Letakkan 2 komponen TTable pada Form
  - Letakkan 2 komponen TdataSource pada Form
  - Letakkan 2 komponen TDBGrid pada Form
  - Rancanglah form seperti tampak pada Gambar 8.3

Kdpasien	Nama	Kelamin	Alamat
P000001	Cindy	P	Jl. Pemuda No. 23 Yogya
P000002	Santi	P	Jl. Pemuda No. 13 Yogya
P000003	Hadi	L	Jl. Pemuda No. 33 Yogya
P000004	Kiki	L	Jl. Pemuda No. 43 Yogya
P000005	Popy	P	Jl. Pemuda No. 53 Yogya

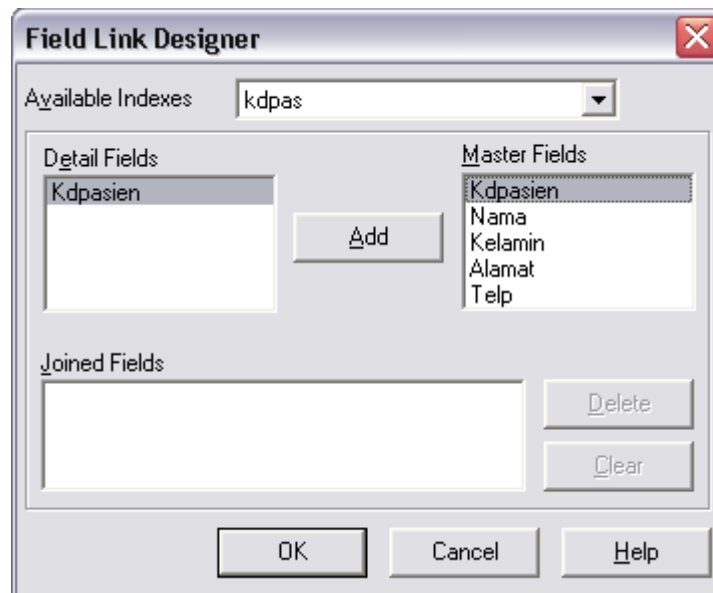
  

Noperiksa	Tglperiksa	Kdpasien	Keluhan
RM00000001	02/12/2009	P000001	
RM00000002	02/01/2010	P000001	
RM00000003	04/01/2010	P000001	

Gambar 8.3 Rancangan Form Penggabungan Tabel

Komponen	Properti	Nilai
Form	Caption	Menghubungkan Pasien dan Periksa
Database1	AliasName DatabaseName	STANDARD1 Databaseku
Table1	DatabaseName TableName	Databaseku Pasien.db
DataSource1	DataSet	Table1
DBGrid1	DataSource	
Table2	DatabaseName TableName MasterSource MasterFields	Databaseku Periksa.db DataSource1 Kdpasien
DataSource2	DataSet	Table2
DBGrid2	DataSource	DataSource

- Untuk mengisi properti Table2.MasterFields, klik Object Inspector, pilih MasterFields dan klik tombol '...', Delphi akan menampilkan kotak dialog Field Link Designer seperti Gambar 8.4.



Gambar 8.4 Kotak Dialog Field Link Designer

Pada kota dialog ini, pilih KdPas (sesuai nama secondary index) dari combo Available Indexes. Kemudian klik Kdpasien pada kotak daftar Detail Fields dan Master Fields.

Setelah kedua field KdPasien tersebut tersoroti, klik Add agar muncul teks 'Kdpasien-> Kdpasien' pada kotak daftar Joined Fields. Langkah ini memberitahukan Delphi bahwa field Kdpasien pada Periksa.db akan dihubungkan dengan field Kdpasien pada Pasien.db.

- Langkah terakhir dari penggabungan ini adalah mengaktifkan Table1 dan Table2.
- Setelah selesai, coba jalankan program (Run)

#### F. Soal-Soal

1. Apa yang Anda rasakan dan Anda pahami setelah melakukan praktikum sampai pada modul 8 ini?
2. Buatlah form untuk pengolahan data pemeriksaan!
3. Integrasikan semua pengolahan data yang sudah dibuat mulai dari modul 7 sampai modul 8 yang dikaitkan dalam sebuah menu!

# MODUL IX

## Pemrograman Database Dengan MS ACCESS

### A. Pengantar

Setelah menggunakan database BDE pada modul sebelumnya, pada modul ini akan menerapkan *database* MS Access untuk pengolahan *database* menggunakan Delphi. Microsoft Access adalah sebuah perangkat lunak (*software*) yang digunakan untuk manage *database* agar data dapat terorganisasi dengan baik. **Microsoft Access** adalah suatu program aplikasi basis data komputer relasional yang digunakan untuk merancang, membuat dan mengolah berbagai jenis data dengan kapasitas yang besar.

Ada empat objek penting pada Ms Access yaitu: Tabel, Query, Form, dan Report. Tabel menjadi objek utama dalam sebuah database yang digunakan untuk menyimpan sekumpulan data sejenis dalam sebuah objek. Query dapat digunakan untuk menampilkan data yang berasal dari satu tabel atau lebih dari satu tabel. Form dapat digunakan untuk mengolah dan menampilkan data. Report dapat digunakan untuk *output* dari pengolahan data.

Terdapat beberapa jenis data pada Ms Access, yaitu:

1. **Text** : tipe data untuk karakter, angka, atau gabungannya. Lebar data maksimal adalah 255 karakter. Tipe data ini cukup sering digunakan.
2. **Memo** : tipe data yang mirip dengan Text, tapi kemampuan menampung data lebih banyak. dapat menampung 64000 karakter untuk tiap fieldnya, tapi tidak bisa diurutkan/diindeks.
3. **Number** : tipe data ini digunakan untuk menampung data angka
4. **Date/Time** : tipe data ini untuk menampung data jenis tanggal atau jam
5. **Currency** : tipe data ini digunakan untuk data angka, tapi dengan adanya lambang mata uang.
6. **AutoNumber** : tipe ini untuk data angka terurut secara otomatis
7. **Yes/No** : tipe data ini merupakan jenis data logika yang hanya berisi Yes atau No.
8. **Ole Object** : tipe data ini untuk eksternal objek, seperti untuk gambar (Bitmap) atau file suara.
9. **Hyperlink** : tipe data ini merupakan dari teks dan bilangan
10. **Lookup Wizard** : tipe data ini merujuk ke suatu data. Jika menggunakan tipe data ini untuk sebuah field, maka bisa memilih sebuah nilai dari tabel lain atau dari sebuah daftar nilai yang ditampilkan dalam *combobox*

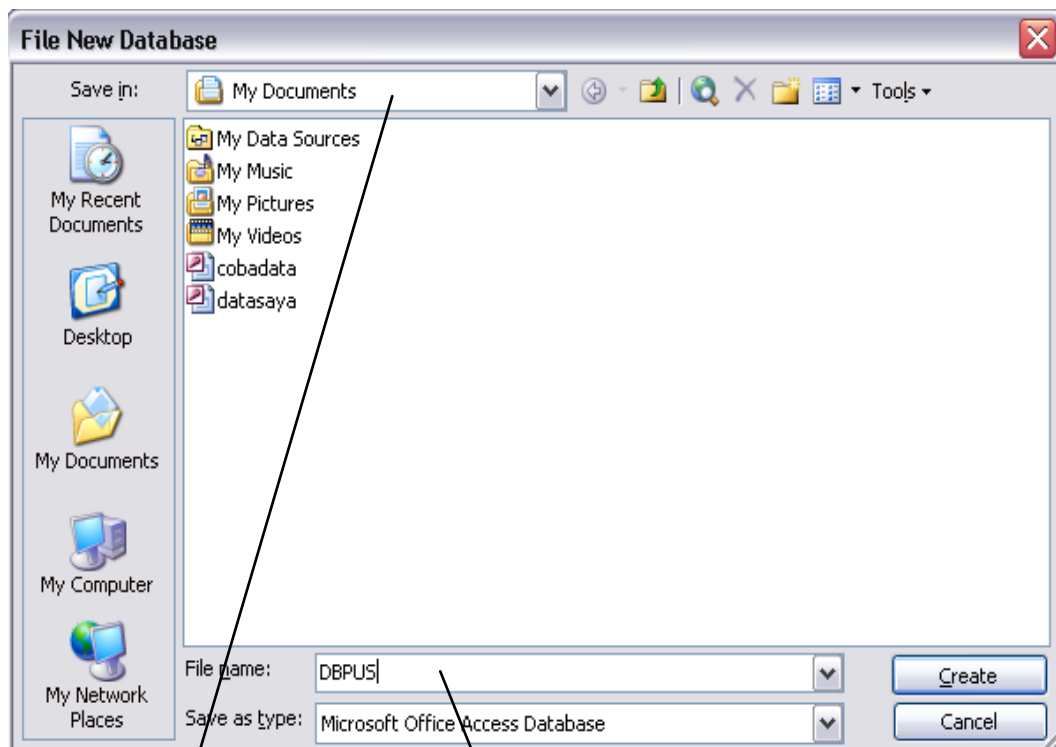
### B. Praktek Untuk Database

#### 1. Membuat *Database* Dengan Ms Access

Untuk membuat *database* dengan MS Access, lakukan langkah-langkah sebagai berikut:

- Klik menu Start
- Arahkan penunjuk ke All Program
- Cari pilihan Microsoft Office
- Arahkan penunjuk ke Microsoft Office Access, lalu klik
- Tunggu beberapa saat sampai masuk ke layar Microsoft Access
- Klik Icon New atau klik File, pilih New
- Klik pilihan **Blank Database** yang ada pada bagian sebelah kanan atas layar.
- Saat masuk ke kotak dialog "File New Database" seperti terlihat pada Gambar 9.1, ketikkan nama file database yang akan dibuat (misalkan diberi nama DBPUS), lanjutkan dengan mengklik tombol **Create**. Perhatikan bahwa sebelum mengklik tombol Create, pastikan directory kerja sudah benar yaitu dimana file database akan disimpan. Untuk kebutuhan praktikum modul ini semua pekerjaan disimpan pada folder **Sesi-8**.





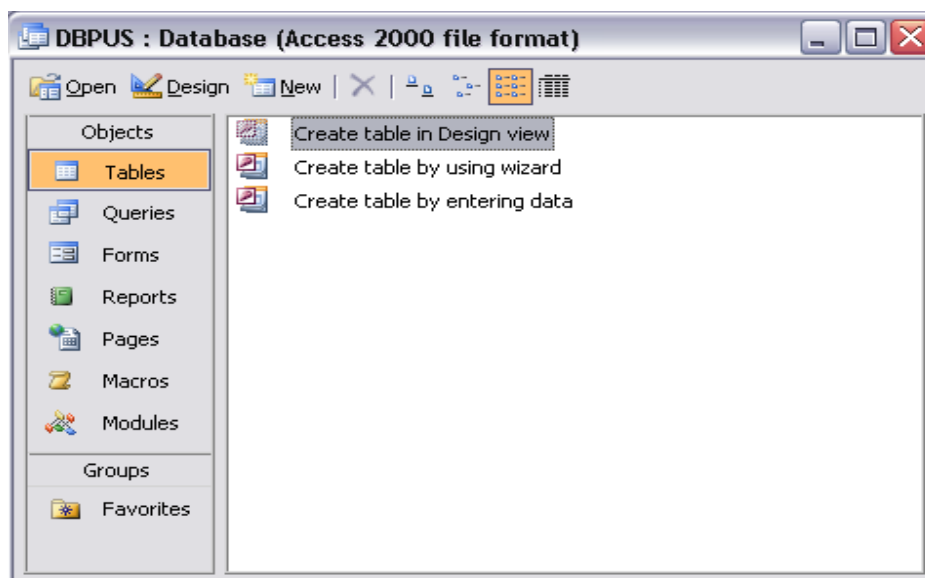
Directory kerja

#### Nama file database

Nama file database silahkan dibuat sesuai dengan nama yang Anda senangi, namun sebaiknya tidak terlalu panjang dan dapat menggambarkan isi datanya

Gambar 9.1 Kotak Dialog Penyimpanan Database

Setelah mengklik tombol Create, akan masuk ke kotak dialog *database* manajer seperti Gambar 9.2. Pada kotak dialog ini telah disediakan tombol perintah untuk aksi yang ingin dilakukan yang ada di bagian kiri atas, dan pilihan objek-objek (Tables, Queries, Forms, Reports, dll) yang ada dibawahnya.



Gambar 9.2 Database Manager Ms Access

## 2. Membuat Tabel

Tabel digunakan untuk menyimpan data. Buatlah tabel-tabel berikut ini sesuai petunjuk dan isilah data minimal 5 buah data. Untuk memulai membuat tabel,

### a. Tabel Dokter

Pastikan posisi pada Database Manager dan obyek yang ditunjuk adalah **Tables**, lalu klik tombol **New**, pilih **Design View**, klik **OK**. Saat berada pada layar editor tabel, isikanlah field-field yang diinginkan seperti terlihat pada Gambar 9.3

Field Name	Data Type	Description
kddokter	Text	
nmdokter	Text	
almdokter	Text	
telpdokter	Text	
kategori	Text	
tarif	Number	

Field Properties	
General    Lookup	
Field Size	5
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

Gambar 9.3 Rancangan Tabel Dokter

Ketiklah field-field seperti gambar di atas. Masing-masing field yang berjenis Text perlu ditetapkan lebarnya pada bagian Field Size. Misalnya field kddokter lebarnya adalah 5, nmdokter lebarnya 40, almdokter lebarnya 50, telpdokter lebarnya 15, kategori lebarnya 10.

Setelah selesai semua field diketikan, arahkan mouse pada field kddokter, lalu klik icon *primary key* (tanda kunci) untuk menjadikan kddokter sebagai kunci utama. Klik icon Save untuk menyimpan tabel, misalkan diberi nama Dokter, lalu klik OK

Save As

Table Name:  
Dokter

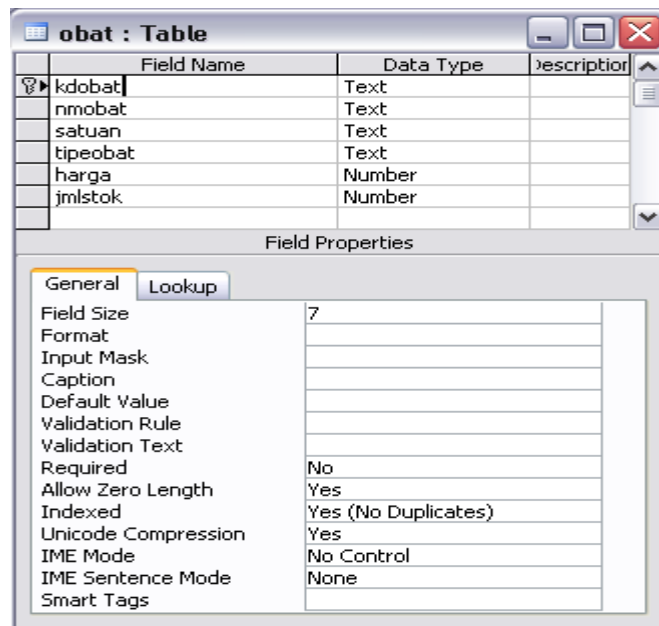
OK Cancel

Gambar 9.4 Kotak Dialog Penyimpanan Tabel

Setelah itu klik icon View (icon paling kiri atau berada disebelah kiri icon Save) untuk mengisi data. Apabila data sudah dirasa cukup, tutup tabel sehingga akan kembali ke database manager. Proses untuk membuat tabel dapat diteruskan untuk tabel-tabel yang lain.

b. Tabel Obat

Pastikan posisi pada Database Manager dan obyek yang ditunjuk adalah **Tables**, lalu klik tombol **New**, pilih **Design View**, klik **OK**. Saat berada pada layar editor tabel, isikanlah field-field yang diinginkan seperti terlihat pada Gambar 9.5

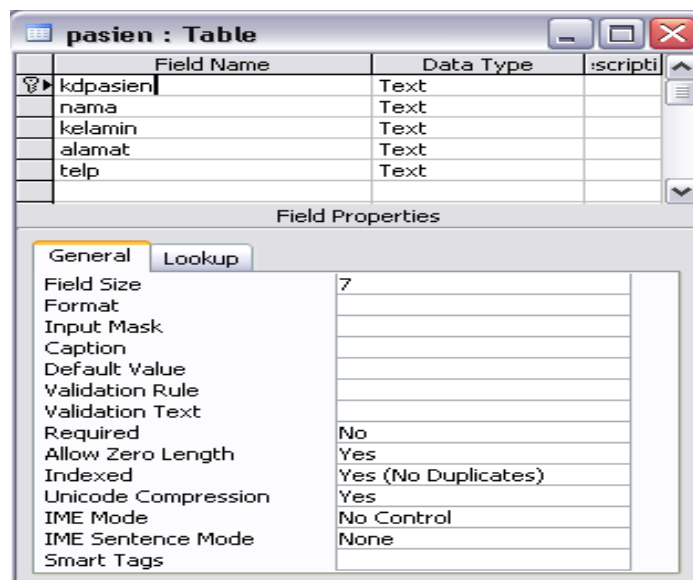


Gambar 9.5 Rancangan Tabel Obat

Field kdobat lebarnya 7, field nmobat lebarnya 50, field satuan lebarnya 20, field tipeobat lebarnya 20. Field kdobat dijadikan sebagai primary key.

c. Tabel Pasien

Pastikan posisi pada Database Manager dan obyek yang ditunjuk adalah **Tables**, lalu klik tombol **New**, pilih **Design View**, klik **OK**. Saat berada pada layar editor tabel, isikanlah field-field yang diinginkan seperti terlihat pada Gambar 9.6

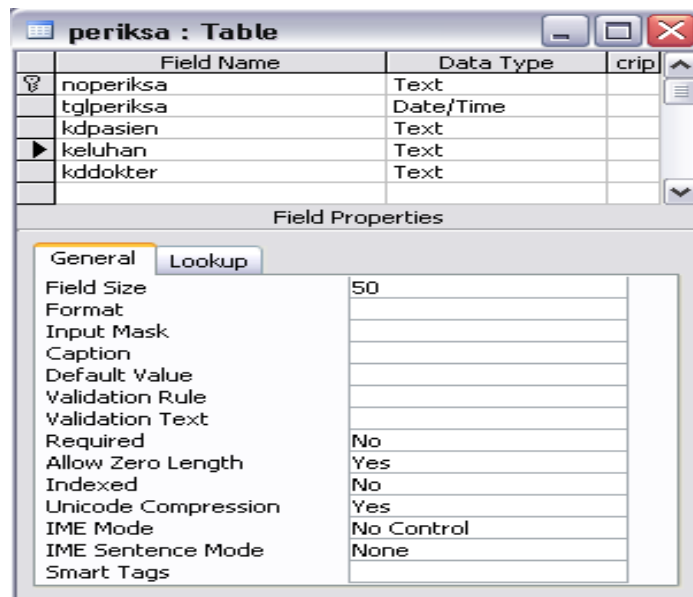


Gambar 9.6 Rancangan Tabel Pasien

Filed kdpasien lebarnya 7, filed nama lebarnya 40, filed kemain lebarnya 1, field alamat lebarnya 50, filed telp lebarnya 15. Field kdpasien dijadikan primary key

d. Tabel Periksa

Pastikan posisi pada Database Manager dan obyek yang ditunjuk adalah **Tables**, lalu klik tombol **New**, pilih **Design View**, klik **OK**. Saat berada pada layar editor tabel, isikanlah field-field yang diinginkan seperti terlihat pada Gambar 9.7

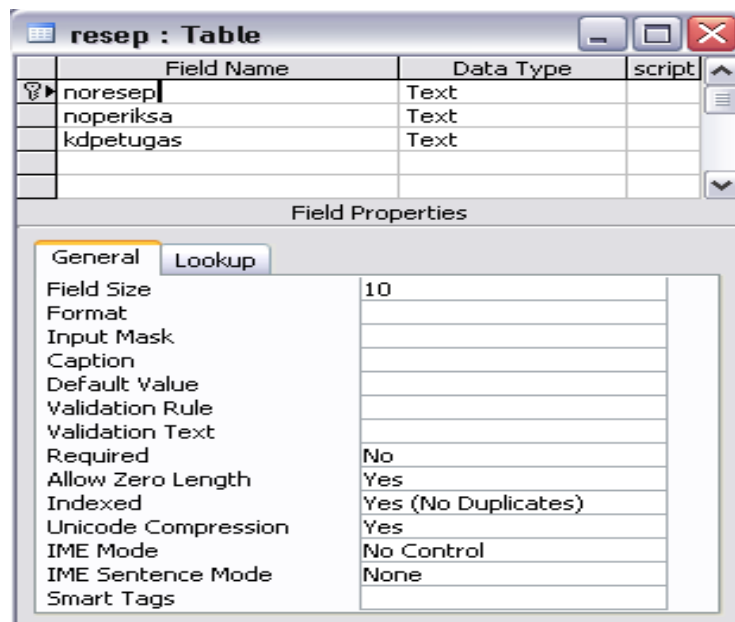


Gambar 9.7 Rancangan Tabel Periksa

Field noperiksa lebarnya 10, field kdpasien lebarnya 7, field keluhan lebarnya 50, field kddokter lebarnya 5. Field noperiksa dijadikan primary key

e. Tabel Resep

Pastikan posisi pada Database Manager dan obyek yang ditunjuk adalah **Tables**, lalu klik tombol **New**, pilih **Design View**, klik **OK**. Saat berada pada layar editor tabel, isikanlah field-field yang diinginkan seperti terlihat pada Gambar 9.8



Gambar 9.8 Rancangan Tabel Resep

Field noresep lebarnya 10, field noperiksa lebarnya 10, field kdpetugas lebarnya 5, Field noresep dijadikan primary key

f. Tabel DetResep

Pastikan posisi pada Database Manager dan obyek yang ditunjuk adalah **Tables**, lalu klik tombol **New**, pilih **Design View**, klik **OK**. Saat berada pada layar editor tabel, isikanlah field-field yang diinginkan seperti terlihat pada Gambar 9.9

Field Name	Data Type	scripti
noresep	Text	
kdobat	Text	
jml	Number	
hrgobat	Number	

Field Properties

General Lookup

Field Size: 10

Format:

Input Mask:

Caption:

Default Value:

Validation Rule:

Validation Text:

Required: No

Allow Zero Length: Yes

Indexed: No

Unicode Compression: Yes

IME Mode: No Control

IME Sentence Mode: None

Smart Tags:

Gambar 9.9 Rancangan Tabel Detail Resep

Field noresep lebarnya 10, field kdobat lebarnya 7. Pada tabel ini tidak ada primary key.

g. Tabel Petugas

Pastikan posisi pada Database Manager dan obyek yang ditunjuk adalah **Tables**, lalu klik tombol **New**, pilih **Design View**, klik **OK**. Saat berada pada layar editor tabel, isikanlah field-field yang diinginkan seperti terlihat pada Gambar 9.5

Field Name	Data Type	Description
kdpetugas	Text	
nmpetugas	Text	
nmuser	Text	
passw	Text	
levelp	Text	

Field Properties

General Lookup

Field Size: 5

Format:

Input Mask:

Caption:

Default Value:

Validation Rule:

Validation Text:

Required: No

Allow Zero Length: Yes

Indexed: Yes (No Duplicates)

Unicode Compression: Yes

IME Mode: No Control

IME Sentence Mode: None

Smart Tags:

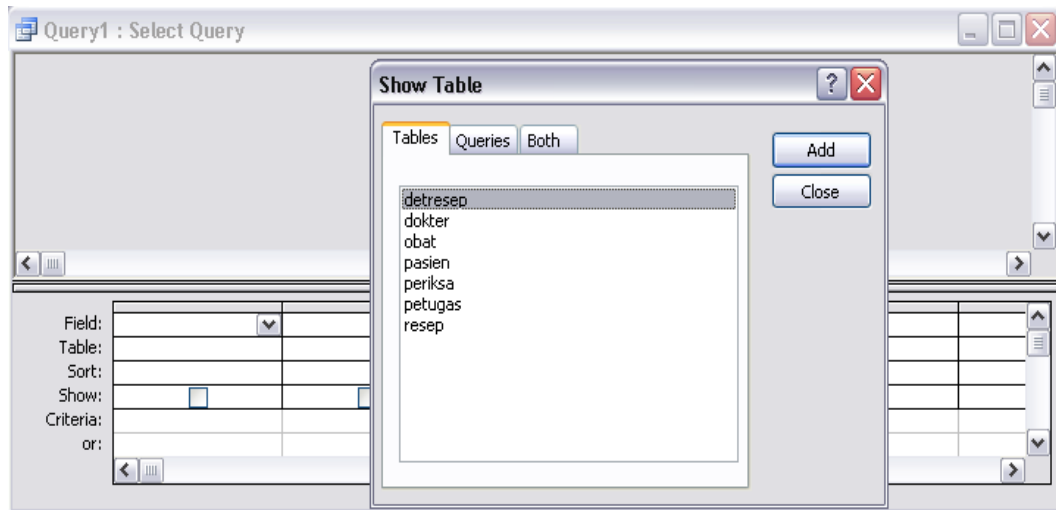
Gambar 9.10 Rancangan Tabel Petugas

Field kdpetugas lebarnya 5, nmpetugas lebarnya 30, nmuser lebarnya 10, passw lebarnya 10, levelp lebarnya 10. Field kdpetugas dijadikan primary key

### 3. Membuat Query

Query digunakan untuk menggabungkan beberapa tabel untuk mendapatkan informasi yang lebih lengkap dari tabel-tabel yang saling berelasi.

Untuk memulai membuat query, pastikan obyek yang ditunjuk adalah **Queries**, lalu klik tombol **New**, pilih **Design View**, klik **OK**. Saat ditampilkan layar seperti Gambar 9.11 di bawah ini, isikanlah nama tabel yang akan membentuk query.



Gambar 9.11 Kotak Dialog Query

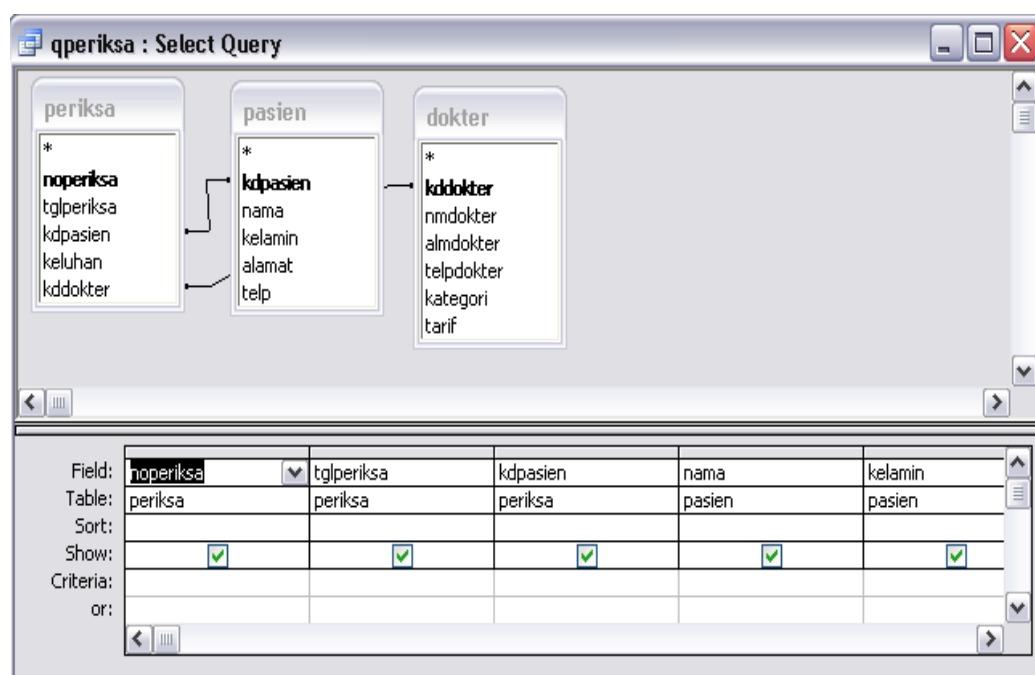
Untuk melanjutkan proses, baca penjelasan pembuatan query periksa di bawah ini:

#### a. Query Periksa

Untuk query periksa, tabel yang terlibat adalah tabel periksa, tabel pasien, dan tabel dokter.

Untuk memasukkan tabel-tabel tersebut, maka saat ditampilkan kotak dialog seperti gambar di atas, klik nama tabel lalu klik tombol Add. Lakukan hal yang sama untuk tabel-tabel yang lain.

Untuk memasukkan field ke kolom-kolom query lakukan dengan cara di drag. Ambillah field-field pada masing-masing tabel sesuai kebutuhan.



Gambar 9.12 Rancangan Query Periksa

Untuk contoh ini yang diambil adalah **semua field pada tabel periksa**, lalu ditambahkan dengan field **nama**, field **kelamin**, field **alamat**, dan field **telp** dari tabel **pasien** yang diletakkan berturut-turut setelah field kdpasien. Sedangkan dari tabel **dokter**, diambil field **nmdokter**, field **alamdokter**, field **telpdokter**, field **kategori**, dan field **tarif** yang diletakkan berturut-turut setelah field **kaddokter**. Pada kolom terakhir ambil field **keluhan** pada tabel **periksa**.

Jika field yang dimasukkan cukup banyak, maka gunakan tombol scroll untuk menggeser ke kanan atau ke kiri.

Jika ada kolom yang salah atau keliru, dapat dihapus dengan cara memblok pada kolom yang bersangkutan, lalu tampilkan menu Edit dan pilih **Delete Columns**.

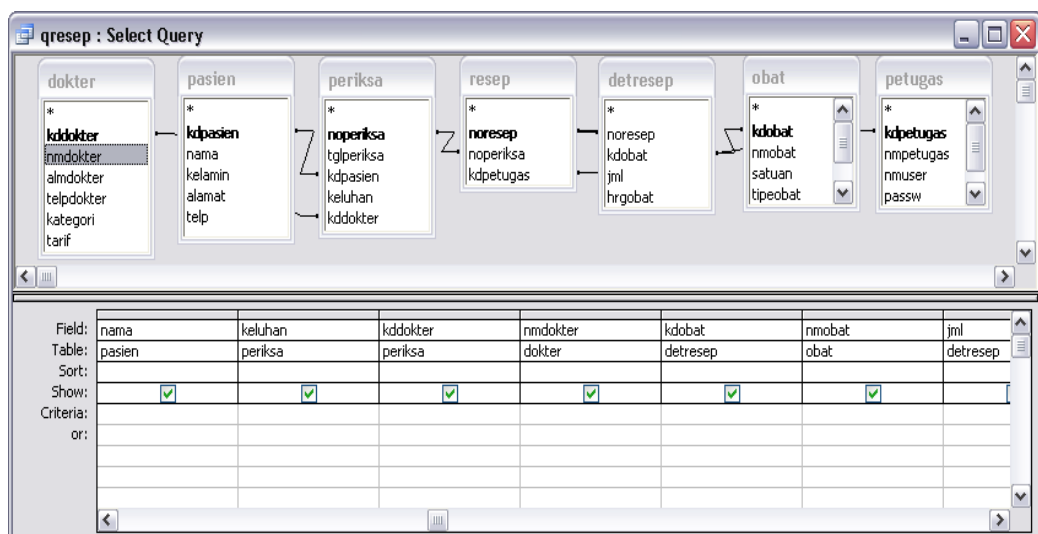
Jika ada penempatan field yang keliru, dapat dipindah dengan cara mendrag (klik mouse pada field tersebut lalu geser atau tarik ke posisi yang diinginkan).

Setelah selesai membuat query di atas, simpanlah dengan nama **qperiksa**. Tutuplah query ini untuk kembali ke *database* manajer agar dapat membuat query yang lain.

#### b. Query Resep

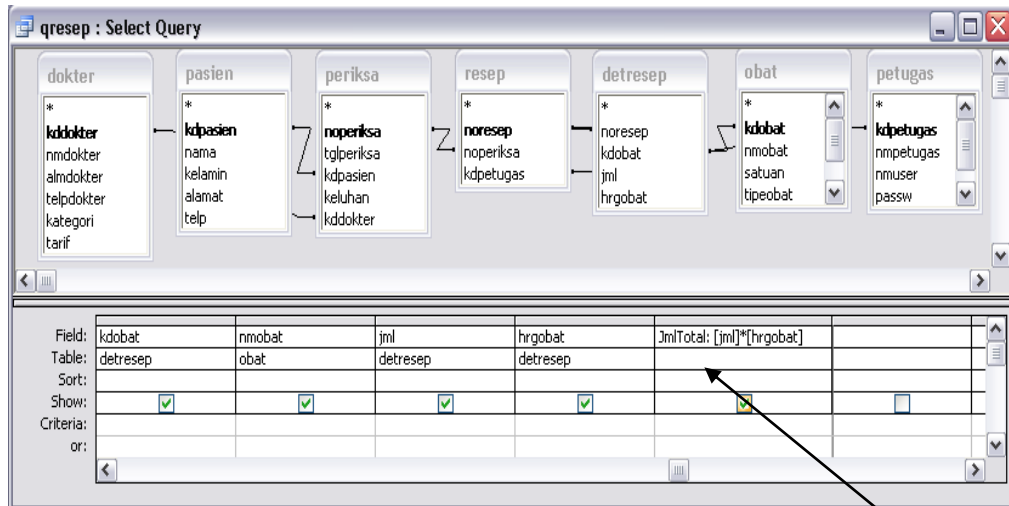
Untuk membuat query ini lakukan langkah-langkah sebagai berikut :

- Dari database manajer, klik tombol New
- Pilih Design View, lalu klik OK
- Masukkan tabel-tabel yang dibutuhkan berturut-turut yaitu : tabel **detresep**, tabel **resep**, tabel **periksa**, tabel **pasien**, tabel **dokter**, tabel **obat**, dan tabel **petugas**.



Gambar 9.13 Rancangan Query Resep

- Masukkan field-field yang dibutuhkan berturut-turut : field **noresep** dari tabel **resep**, field **noperiksa** dari tabel **resep**, field **tglperiksa** dari tabel **periksa**, field **kdpasien** dari tabel **periksa**, field **nama** dari tabel **pasien**, field **keluhan** dari tabel **periksa**, field **kddokter** dari tabel **periksa**, field **nmdokter** dari tabel **dokter**, field **kdobat** dari tabel **detresep**, field **nmobat** dari tabel **obat**, field **jml** dari tabel **detresep**, field **hrgobat** dari tabel **detresep**.
- Jika diinginkan ada field tambahan, Anda dapat membuat ekspresi sendiri pada kolom kosong misalnya untuk menampilkan **JmlHarga** yang merupakan perkalian antara **jml** dan **hrgobat** (lihat gambar 9.14).



Gambar 9.14 Pembuatan Field Pada Query Resep

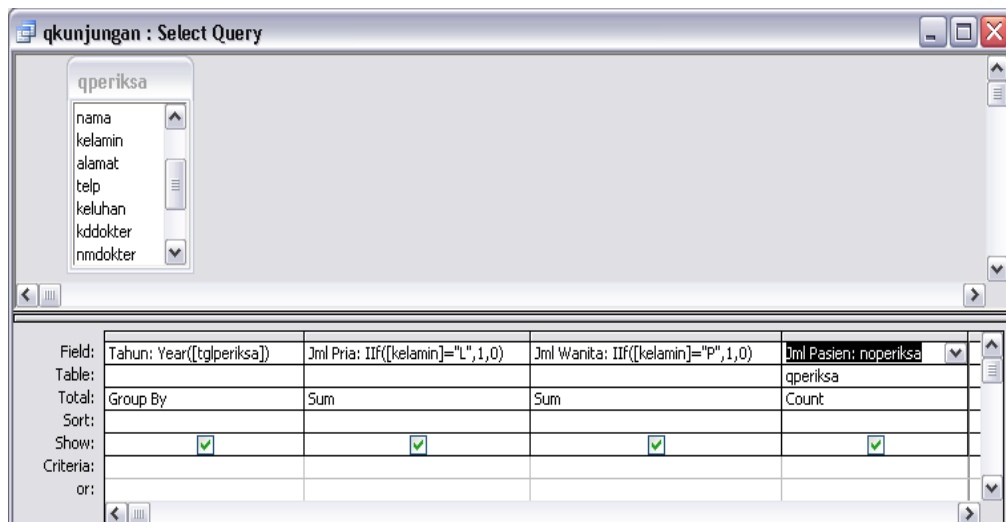
Ketik JmlTotal lalu titik dua, kemudian ketik jml\*hrgobat

#### c. Query Crosstab

Jenis query ini diperlukan untuk menghasilkan informasi yang berupa rekapitulasi data. Maksudnya adalah data-data yang sama dikelompokkan dalam satu baris data dan dilakukan manipulasi data.

Contoh berikut adalah untuk membuat rekapitulasi kunjungan pasien pada puskesmas. Data akan dilihat berdasarkan tahun dan informasi yang ditampilkan adalah berapa jumlah pasien laki-laki, pasien perempuan dan jumlah semua pasien. Untuk membuat query ini lakukan langkah-langkah sebagai berikut :

- Dari database manajer, klik tombol New
- Pilih Design View, lalu klik OK
- Ambil qperiksa sebagai sumber data
- Klik menu View lalu klik pada pilihan Totals



Gambar 9.15 Rancangan Query Crosstab

- Tulislah ekspresi-ekspresi seperti terlihat pada gambar di atas

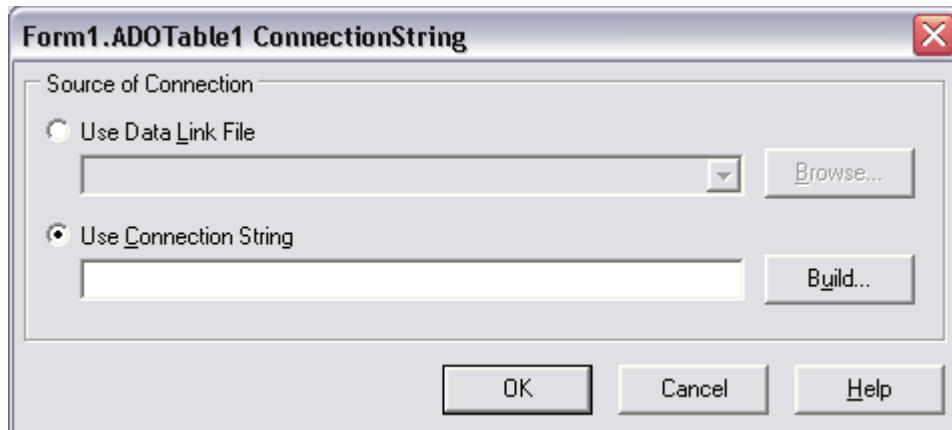


### C. Praktek Untuk Membuat Aplikasi ADO

1. Aplikasi untuk menampilkan data dengan ADOTable

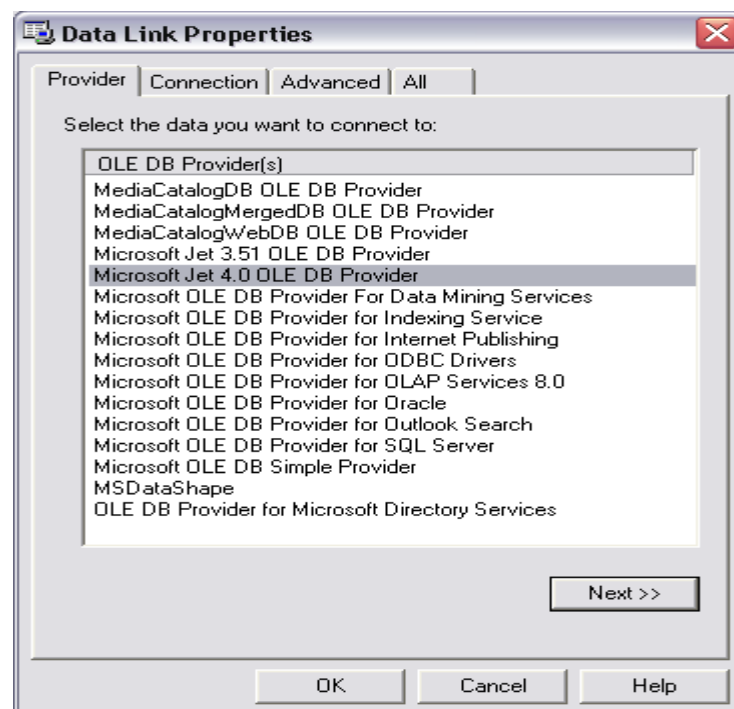
Untuk mencoba membuat aplikasi database dengan ADO, maka buatlah aplikasi baru, dan pada form lakukanlah hal-hal sebagai berikut:

- Ambil komponen ADOTable (diambil pada tab page **ADO**)
- Aktifkan Object Inspector dan klik ganda properti ConnectionString. Anda akan melihat kotak dialog seperti Gambar 9.16.



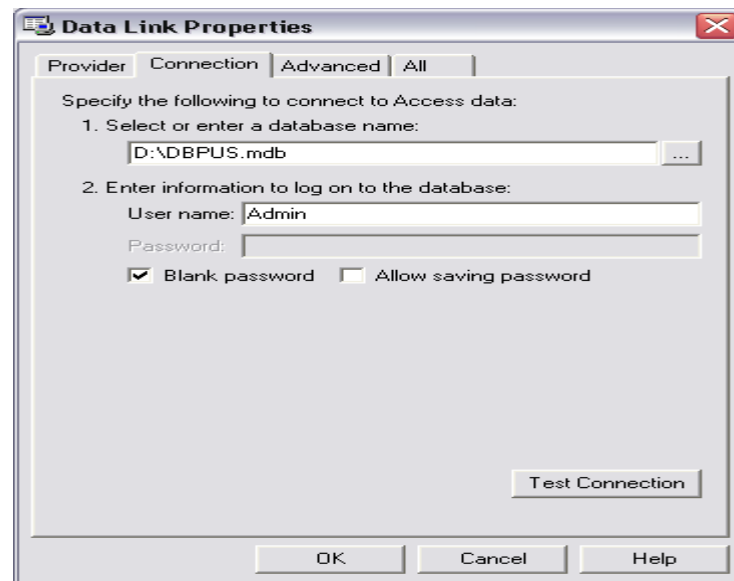
Gambar 9.16 Kotak Dialog ConnectionString

- Klik Build dan Anda akan melihat kotak dialog seperti Gambar 9.17.



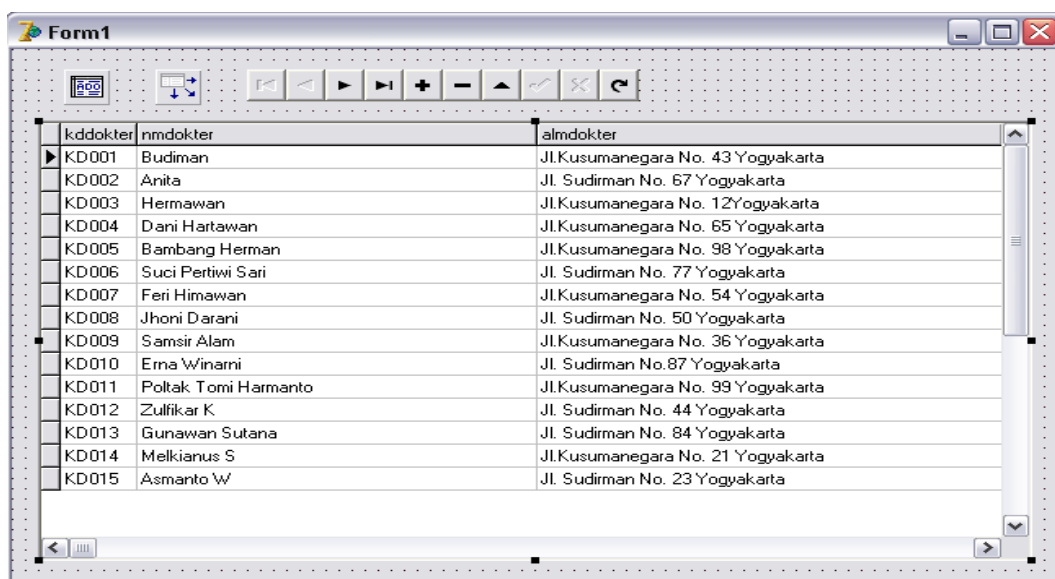
Gambar 9.17 Kotak Dialog Build

- Pilihlah Microsoft Jet 4.0 OLE DB Provider dan klik Next. Pada kotak dialog berikutnya (lihat Gambar 9.18), masukkan basis data DBPUS (klik tombol titik tiga untuk mencari database).



Gambar 9.18 Kotak Dialog Database

- Setelah Anda menentukanConnectionString dan menetapkan databasenya, masukkan nama tabel pada properti TableName. Sebagai contoh, pilihlah tabel DOKTER. Klik tombol OK sampai selesai.
- Ubahlah properti Active menjadi true
- Ambil komponen TdataSource diambil pada tab page **Data Access**, lalu atur properti **DataSet** dan isikan ADOTable1 (klik combo).
- Ambil komponen TDBNavigator pada tab page **Data Controls**, lalu atur properti **DataSource** dan isikan DataSource1.
- Ambil komponen TDBGrid pada tab **Data Controls**, lalu atur properti **DataSource** dan isikan DataSource1

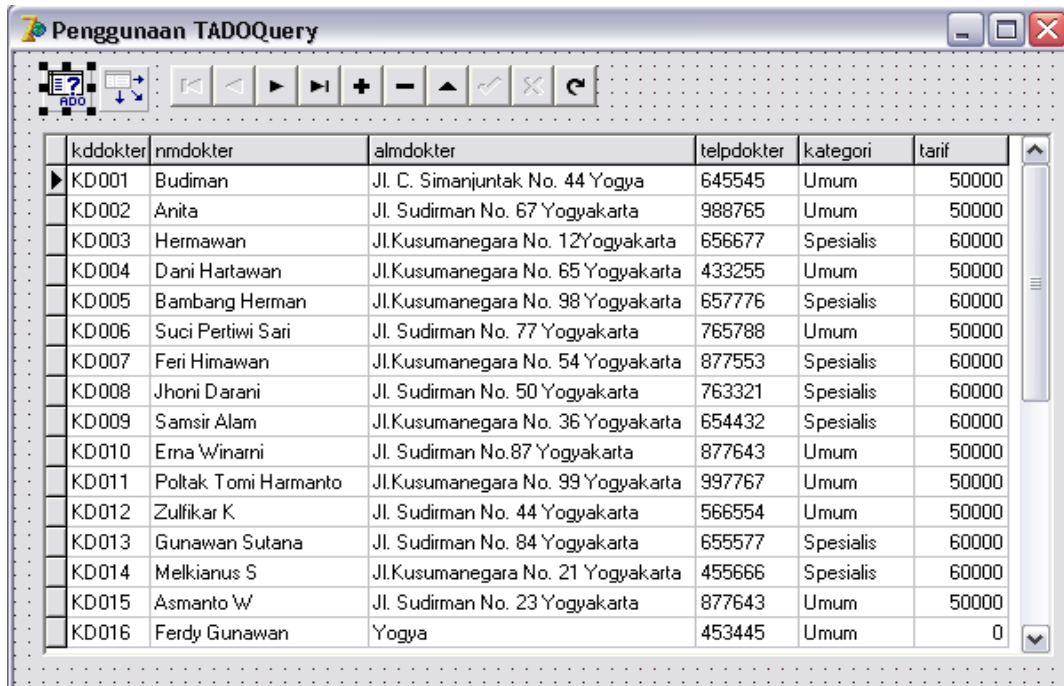


Gambar 9.19 Form Menampilkan Data Dengan ADO Table

- Jalankan aplikasi (Run)

2. Aplikasi menampilkan data dengan ADOQuery

- Tambahkan form baru
- Rancanglah form seperti terlihat pada Gambar 9.20.
- Gunakan komponen-komponen ADOQuery (diambil pada tab page **ADO**), TDataSource (diambil pada tab page **Data Access**), TDBNavigator (diambil pada tab page **Data Controls**), TDBGrid (diambil pada tab **Data Controls**).
- Aturlah properti masing-masing komponen seperti cara menampilkan data dengan ADOTable. Perbedaannya adalah saat mengisi nama tabel tabel ADOTable tapi pada ADOQuery yang diisi adalah SQL
- Isilah properti SQL dengan **SELECT \* FROM DOKTER**



kddokter	nmdokter	almdokter	telpdokter	kategori	tarif
KD001	Budiman	Jl. C. Simanjuntak No. 44 Yogya	645545	Umum	50000
KD002	Anita	Jl. Sudirman No. 67 Yogyakarta	988765	Umum	50000
KD003	Hermawan	Jl.Kusumanegara No. 12Yogyakarta	656677	Spesialis	60000
KD004	Dani Hartawan	Jl.Kusumanegara No. 65 Yogyakarta	433255	Umum	50000
KD005	Bambang Herman	Jl.Kusumanegara No. 98 Yogyakarta	657776	Spesialis	60000
KD006	Suci Pertiwi Sari	Jl. Sudirman No. 77 Yogyakarta	765788	Umum	50000
KD007	Feri Himawan	Jl.Kusumanegara No. 54 Yogyakarta	877553	Spesialis	60000
KD008	Jhoni Darani	Jl. Sudirman No. 50 Yogyakarta	763321	Spesialis	60000
KD009	Samsir Alam	Jl.Kusumanegara No. 36 Yogyakarta	654432	Spesialis	60000
KD010	Erna Winarni	Jl. Sudirman No.87 Yogyakarta	877643	Umum	50000
KD011	Poltak Tomi Harmanto	Jl.Kusumanegara No. 99 Yogyakarta	997767	Umum	50000
KD012	Zulfikar K	Jl. Sudirman No. 44 Yogyakarta	566554	Umum	50000
KD013	Gunawan Sutana	Jl. Sudirman No. 84 Yogyakarta	655577	Spesialis	60000
KD014	Melkianus S	Jl.Kusumanegara No. 21 Yogyakarta	455666	Spesialis	60000
KD015	Asmanto W	Jl. Sudirman No. 23 Yogyakarta	877643	Umum	50000
KD016	Ferdy Gunawan	Yogya	453445	Umum	0

Gambar 9.20 Form Menampilkan Data Dengan ADOQUERY

- Jalankan aplikasi (Run)

### 3. Aplikasi untuk pengolahan data

- Rancanglah form seperti Gambar 9.21 (dianggap form yang tercipta adalah Form3, jika berbeda, Anda tinggal menyesuaikan sedikit kode program)
- Letakkan komponen TADOTable pada Form (diambil pada tab page **ADO**). Atur properti **ConnectionString**. Isi properti **TableName** dengan nama tabel yang akan digunakan (tabel DOKTER). Kemudian ubahlah properti **Active** menjadi True.
- Letakkan komponen TDataSource pada Form (diambil pada tab page **Data Access**). Isilah properti **DataSet** dengan ADOTable1.
- Letakkan komponen TDBGrid pada Form (diambil pada tab **Data Controls**). Isilah properti **DataSource** dengan DataSource1.

Kode Dokter	Nama Dokter	Alamat Dokter
KD001	Budiman	Jl. C. Simanjuntak No. 4
KD002	Anita	Jl. Sudiman No. 67 Yogi
KD003	Hermawan	Jl. Kusumanegara No. 12
KD004	Dani Hartawan	Jl. Kusumanegara No. 65
KD005	Bambang Herman	Jl. Kusumanegara No. 98
KD006	Suci Pertiwi Sari	Jl. Sudiman No. 77 Yogi
KD007	Feri Himawan	Jl. Kusumanegara No. 54
KD008	Jhoni Darani	Jl. Sudiman No. 50 Yogi
KD009	Samsir Alam	Jl. Kusumanegara No. 36
KD010	Erna Winarni	Jl. Sudiman No. 87 Yogi
KD011	Poltak Tomi Harmanto	Jl. Kusumanegara No. 99
KD012	Zulfikar K	Jl. Sudiman No. 44 Yogi
KD013	Gunawan Sutana	Jl. Sudiman No. 84 Yogi
KD014	Melkianus S	Jl. Kusumanegara No. 21
KD015	Asmanto W	Jl. Sudiman No. 23 Yogi
KD016	Ferdy Gunawan	Yogya

Gambar 9.21 Form Pengolahan Data Dengan ADO

- Letakkan 5 komponen Label pada Form. Ubahlah **Caption** masing-masingnya seperti terlihat pada Gambar 9.21.
- Letakkan 4 komponen Edit, dan 2 komponen RadioButton. Ubahlah **Caption** dari masing-masing RadioButton menjadi : Umum dan Spesialis
- Letakkan 5 (lima) komponen Button pada Form. Ubahlah Caption masing-masing Button tersebut. **Button1** menjadi **&Tambah**, **Button2** menjadi **&Batal**, **Button3** menjadi **&Edit**, **Button4** menjadi **&Hapus**, dan **Button5** menjadi **&Simpan**. Ubah juga Name masing-masing Button tersebut. **Button1** menjadi **cmdtambah**, **Button2** menjadi **cmdbatal**, **Button3** menjadi **cmdedit**, **Button4** menjadi **cmdhapus**, dan **Button5** menjadi **cmdsimpan**
- Tambahkan satu komponen Edit dan satu komponen Button dan letakkan di bawah DBGrid, (lihat gambar)
- Atur properti untuk Button yang ditambahkan, dimana **Caption** diganti menjadi **&Cek**, dan **Name** diganti dengan **cmdcari**.
- Buatlah beberapa prosedur yang diletakkan diantara Implementation dengan End., (jika Anda berada pada tampilan form, untuk masuk ke Editor Code, tekan tombol fungsi F12). Jika sudah berada di Editor Code, maka letakkan kursor di antara Implementation dengan End, lalu ketiklah prosedur-prosedur yang dibutuhkan sebagai berikut:

```

procedure nilaiawal;
begin
  form3.Edit1.text:=' ';
  form3.Edit2.text:=' ';
  form3.Edit3.text:=' ';
  form3.Edit4.text:=' ';
  form3.radiobutton1.Checked:=false;
  form3.radiobutton2.Checked:=false;
end;

procedure buatkode;
var kode:integer;
begin
  if form3.adotable1.RecordCount=0 then
    form3.edit1.Text:='KD001'
  else
    begin
      form3.adotable1.Last;
      kode:=strtoint(copy(form3.adotable1 .fieldbyname('kddokter').AsString,4,5))+1;
      if kode<1000 then form3.edit1.Text:='KD'+inttostr(kode);
      if kode<100 then form3.edit1.Text:='KD0'+inttostr(kode);
      if kode<10 then form3.edit1.Text:='KD00'+inttostr(kode);
    end;
end;

procedure aturisian(kondisi:boolean);
begin
  form3.Edit1.enabled:=kondisi;
  form3.Edit2.enabled:=kondisi;
  form3.Edit3.enabled:=kondisi;
  form3.Edit4.enabled:=kondisi;
  form3.radiobutton1.enabled:=kondisi;
  form3.radiobutton2.enabled:=kondisi;
end;

procedure aturtombol(k1,k2,k3,k4,k5:boolean);
begin
  form3.cmdtambah.Enabled:=k1;
  form3.cmdbatal.Enabled:=k2;
  form3.cmdedit.Enabled:=k3;
  form3.cmdhapus.Enabled:=k4;
  form3.cmdsimpan.Enabled:=k5;
end;

procedure simpandata;
begin
  form3.adotable1.FieldByName('KDDOKTER').AsString:=form3.Edit1.Text;
  form3.adotable1.FieldByName('NMDOKTER').AsString:=form3.Edit2.Text;
  form3.adotable1.FieldByName('ALMDOKTER').AsString:=form3.Edit3.Text;
  form3.adotable1.FieldByName('TELPDOKTER').AsString:=form3.Edit4.Text;
  if form3.RadioButton1.Checked=true then
    form3.adotable1.FieldByName('KATEGORI').AsString:='Umum';
  if form3.RadioButton2.Checked=true then
    form3.adotable1.FieldByName('KATEGORI').AsString:='Spesialis';
end;

```

```

procedure tampildata;
begin
  form3.Edit1.text:=form3.adotable1.Fields[0].Value;
  form3.Edit2.text:=form3.adotable1.Fields[1].Value;
  form3.Edit3.text:=form3.adotable1.Fields[2].Value;
  form3.Edit4.text:=form3.adotable1.Fields[3].Value;
  if form3.adotable1.Fields[4].Value='Umum' then
  begin
    form3.RadioButton1.Checked
  end;
  if form3.adotable1.Fields[4].Value='Spesialis' then
    form3.RadioButton2.Checked;
  end;
end;

```

- Buatlah kode program untuk masing-masing Button pada kejadian OnClick. Caranya, klik komponen Button yang bersangkutan, lalu masuk ke properties dan aktifkan tab Events dan arahkan mouse pada baris OnClick, dan lakukan klik dua kali. Masing-masing kode adalah sebagai berikut:

\* Saat cmdtambah OnClick

```

nilaiawal;
buatkode;
aturisian(true);
aturtombol(false,true,false,false,true);
form3.Edit2.SetFocus;

```

\* Saat cmdbatal OnClick

```

nilaiawal;
aturtombol(true,false,false,false,false);
aturisian(false);
if form3.adotable1.RecordCount>0 then
  tampildata;
  aturtombol(true,true,true,true,false);

```

- Saat cmdedit OnClick

```

aturisian(true);
aturtombol(false,true,false,false,true);
form3.Edit2.SetFocus;

```

\* Saat cmdhapus OnClick

```

if messagedlg('Benar akan dihapus?',
  mtconfirmation,[mbytes,mbno],0)=mryes then
  adotable1.Delete;
  nilaiawal;
  aturtombol(true,false,false,false,false);
  aturisian(false);
  if form3.adotable1.RecordCount>0 then
    tampildata;
    aturtombol(true,true,true,true,false);

```

\* Saat cmdsimpan OnClick

```
if adotable1.Locate('kddokter',Edit1.Text,[loPartialKey])=true then
  If messagedlg('Data ini sudah ada, mau diperbarui?',
    mtconfirmation,[mbyes,mbno],0)=mryes then
    begin
      adotable1.edit;
      simpandata;
      adotable1.Post;
    end
  else
    begin
      tampildata;
    end
  else
    begin
      adotable1.Append;
      simpandata;
      adotable1.Post;
    end;
  aturisian(false);
  aturtombol(true,false,true,true,false);
  form3.DBGrid1.Refresh;
```

\* Saat cmdcari OnClick

```
if adotable1.Locate('nmdokter',Edit5.Text,[loPartialKey])=true then
  begin
    tampildata;
  end
  else
    ShowMessage('Nama Yang Anda Cari Tidak Ditemukan');
  end;
```

\* Saat DBGrid OnCellClick

```
tampildata;
```

\* Saat Form OnCreate

```
nilaiawal;
aturtombol(true,false,false,false,false);
aturisian(false);
if form3.adotable1.RecordCount>0 then
  tampildata;
  aturtombol(true,false,true,true,false);
```

- Jalankan program (run)

#### D. Soal-Soal

1. Apa pendapat Anda setelah menggunakan ADO dalam berinteraksi dengan database?
2. Jika dibandingkan dengan BDE, menurut Anda mana yang lebih praktis?
3. Buatlah form untuk melakukan pengolahan data yang lain dari tabel yang sudah diciptakan, seperti pengolahan data pasien, data obat dan lain-lain!
4. Integrasikan semua form yang dihasilkan dengan menggunakan menu!

# MODUL X

## Membuat Laporan

### E. Pengantar

Setelah melewati materi pada modul yang berkaitan dengan pemrograman database, tentu diharapkan sudah mengerti dan bisa melakukan pengolahan data. Namun tidak lengkap apabila hanya sampai bisa melakukan pengolahan data saja, jika tidak didukung dengan kemampuan pembuatan laporan.

Pada modul ini akan diberikan penjelasan dan tahap-tahap pembuatan laporan dengan memanfaatkan **QuickReport** pada Borland Delphi. Proses pembuatan laporan tetap menggunakan object form kemudian tambahkan komponen **TquickRep** dan tabel.

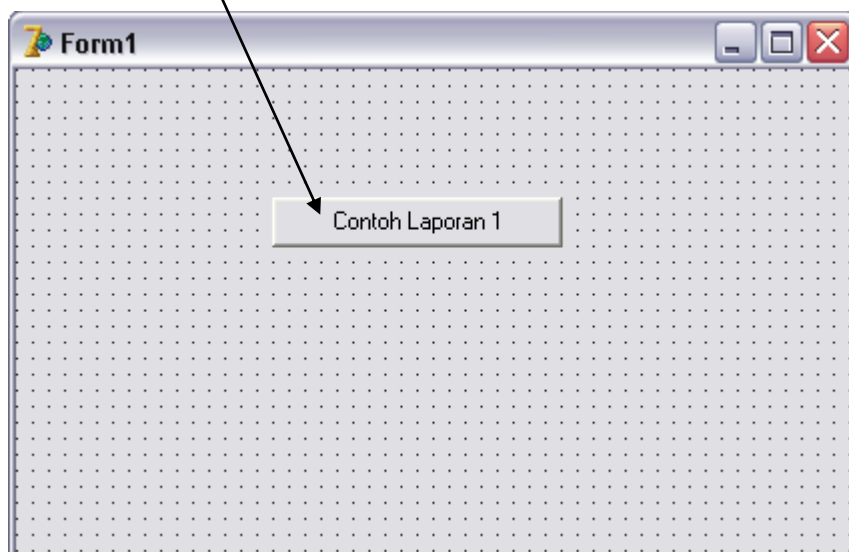
### F. Praktek

Sebelum memulai praktek untuk modul ini, terlebih dahulu siapkan folder untuk menyimpan pekerjaan pada modul ini. Misalnya disiapkan folder dengan nama SESI-10.

Selanjutnya lakukan proses pengcopyan terhadap database yang sudah dibuat pada SESI-9.

#### 1. Membuat Laporan Sederhana

- Buatlah proyek baru
- Letakkan komponen Tbutton pada form dan rubahlah properti **Caption** nya menjadi **Contoh Laporan 1**

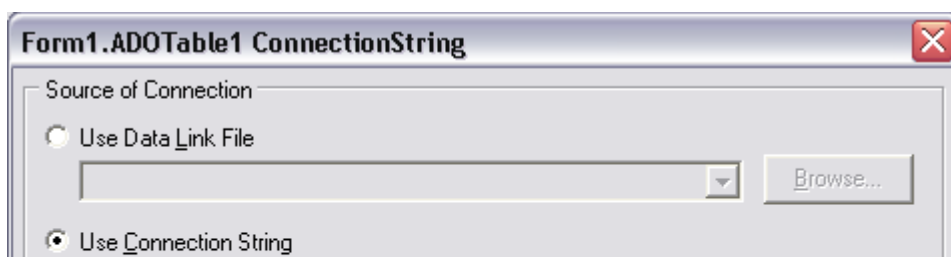


Gambar 10.1 Rancangan Form Untuk Menjalankan Laporan

- Tanganilah kejadian (buat kode program di tab Event) OnClick dengan pernyataan berikut:

**Form2.QuickRep1.Preview;**

- Selanjutnya buatlah form baru
- Letakkan komponen **TADOTable** (diambil pada tab page **ADO**) dan hubungkan dengan database yang telah disediakan. Caranya adalah mengisi properti **ConnectionString**

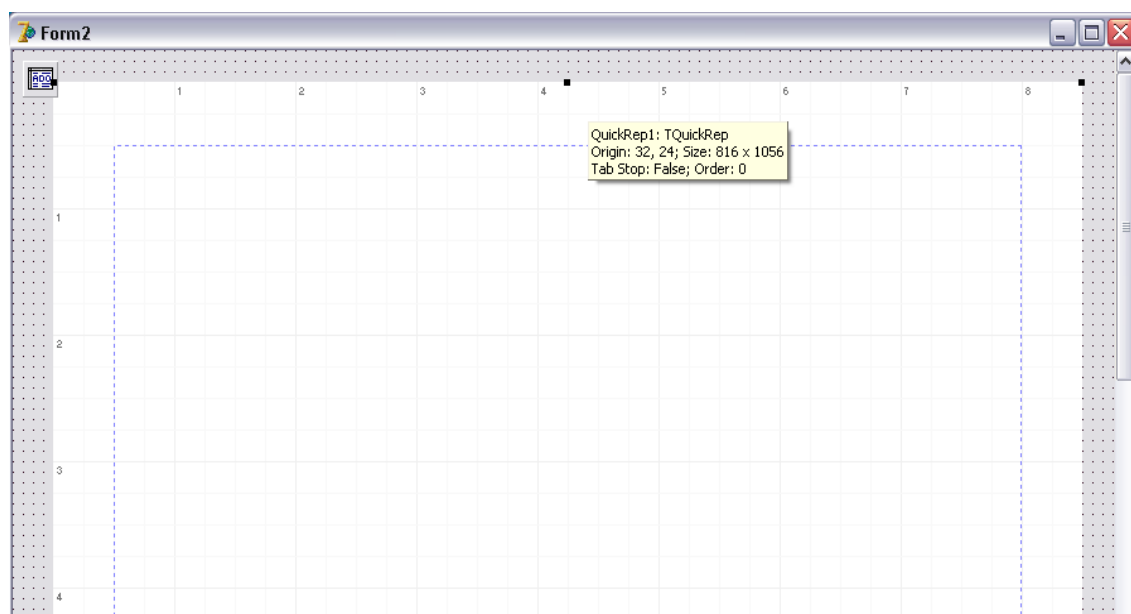




Gambar 10.2 Kotak Dialog ConnectionString

Teruskan proses sampai terhubung dengan *database* yang sesuai (sesuaikan letak *database* dan nama *database*) sampai selesai. Jika diperlukan silahkan lakukan tes koneksi.

- Ubahlah properti **TableName** dan tentukan tabel yang dipakai untuk membuat report adalah tabel dokter
- Ubahlah properti **Active** diubah menjadi true.
- Letakkan komponen **TQuickRep** pada form (diambil pada tab **Qreport**, yang ada di bagian paling kiri). Rancanglah form seperti terlihat pada Gambar 10.3. Jika komponen Tquickreport belum ada, maka tambahkan dengan cara :
  - Tampilkan menu **Project**
  - Pilih **Option**
  - Klik Tab **Packages**
  - Klik **Add**
  - Cari folder tempat komponen berada : C:\Program Files\Borland\Delphi7\Bin lalu pilih **dclqrt70.bpl**
- Hubungkan komponen TquickRep dengan tabel DOKTER dengan cara mengubah properti **DataSet** menjadi ADOTable1.



Gambar 10.3 Form Untuk Rancangan Laporan

- Atur properti Qreport1 pada bagian **Bands**. Ubahlah nilai properti **hasTitle** menjadi, properti **hasColumnHeader** menjadi **true**, properti **hasDetail** menjadi **true**.
- Letakkan komponen **TQRLabel** pada pita Title, dan ubahlah properti **Caption** menjadi **Laporan Daftar Dokter**. Ubah juga properti **Alignment** menjadi **taCenter**.
- Letakkan enam komponen **TQRLabel** pada pita **HasColumnHeader** dan enam **TQRDBText** pada pita **HasDetail**. Untuk masing-masing komponen TQRDBText, ubahlah properti **DataSet** menjadi **ADOTable1** dan **DataField** dengan masing-masing field dari tabel DOKTER. Hasil rancangan akan terlihat seperti Gambar 10.4

The screenshot shows a report design interface with a grid background. The report is titled 'Form2'. The layout consists of three main bands:

- Title Band:** Contains the title 'Laporan Daftar Dokter' centered.
- Column Header Band:** Contains six fields: {Kode Dokter}, {Nama Dokter}, {Alamat}, {Telepon}, {Kategori}, and {Tarif}.
- Detail Band:** Contains six corresponding data fields: {Kddokter}, {Nmdokter}, {Almdokter}, {Telpdokter}, {Kategori}, and {Tarif}.

Gambar 10.4 Hasil Rancangan Laporan Sederhana

- Jalankan program

## 2. Melengkapi Laporan Dengan Berbagai Pita

- Klik komponen Qreport1
  - Ubahlah properti HasPageHeader, HasPageFooter, dan HasSummary menjadi true.
    - a. Membuat Halaman  
Pada pita PageHeader, letakkan sebuah komponen TQRLabel dan TQRSysData. Ubahlah TQRLabel.Caption menjadi 'Halaman Ke: ' dan properti TQRSysData.Data menjadi qrsPageNumber. Ingat bahwa pita PageHeader akan dicetak pada setiap halaman dari laporan Anda.
    - b. Membuat Nomor Urut  
Pada pita ColumnHeader tambahkan sebuah komponen TQRLabel (letakkan pada kolom paling kiri) dan pada pita hasDetail tambahkan sebuah komponen TQRSysData. Ubahlah komponen TQRLabel menjadi 'Nomor' dan ubahlah komponen TQRSysData.Data menjadi qrsDetailNo.
    - c. Menampilkan Jumlah Record  
Pada pita Summary, tambahkan sebuah komponen TQRLabel dan sebuah komponen TQRSysData. Pita Summary akan ditampilkan setelah semua record ditampilkan. Ubahlah komponen TQRLabel menjadi 'Jumlah Record: ' dan ubahlah komponen TQRSysData.Data menjadi qrsDetailCount.
    - d. Menampilkan Tanggal Cetak  
Pada pita PageFooter, tambahkan sebuah komponen TQRLabel dan sebuah komponen TQRSysData. Pita PageFooter akan ditampilkan pada bagian bawah dari semua halaman yang dicetak. Ubahlah komponen TQRLabel menjadi 'Dicetak Tanggal: ' dan ubahlah komponen TQRSysData.Data menjadi qrsDateTime. Komponen ini akan menampilkan tanggal & waktu laporan dicetak.
- Sekarang laporan Anda akan menjadi seperti terlihat pada Gambar di bawah ini.

Gambar 10.5 Hasil Rancangan Laporan Lebih Detail

- Jalankan program

## 3. Melengkapi Laporan Dengan Pembatas Baris dan Kolom

Supaya laporan tampil lebih menarik lagi, Anda dapat menambahkan komponen QRShape. Untuk itu letakkan dan atur posisi QRShape yang digunakan untuk setiap kolom sehingga laporan berubah seperti gambar di bawah ini

The screenshot shows a report form titled "Form2" with a grid layout. The grid is divided into several sections: a page header section with "Halaman Ke" and "Page#", a title section with "Laporan Daftar Dokter", a table section with columns for "Nomor", "Kode Dokter", "Nama Dokter", "Alamat", "Telepon", "Kategori", and "Tarif", a summary section with "Jumlah Record" and "Detail count", and a footer section with "Dicetak Tanggal" and "Date". The form is designed to be printed as a report.

Gambar 10.6 Hasil Rancangan Laporan Lengkap

#### 4. Membuat Laporan Dengan Mengelompokkan Data

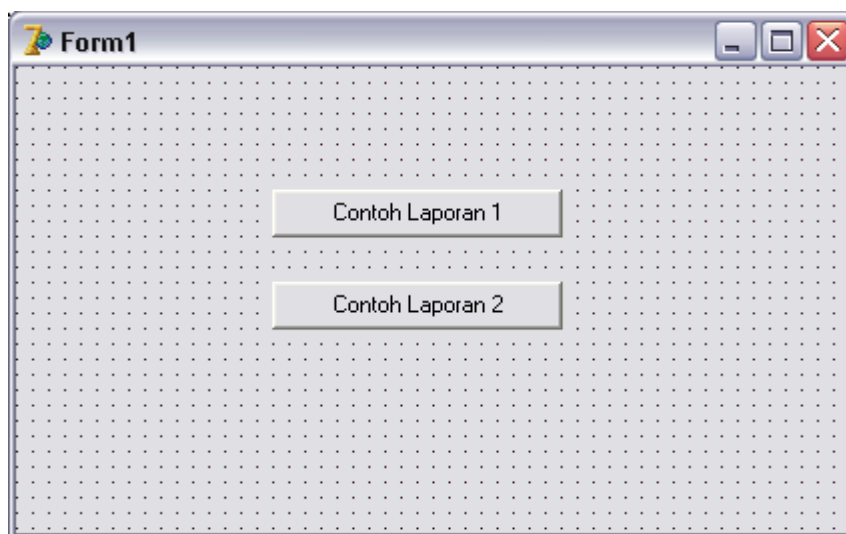
- Tambahkan form baru pada proyek.
- Letakkan komponen ADOQuery (komonen ini mirip dengan ADOTable, kita gunakan hanya sebagai pembanding)
- Atur Properti ADOQuery untuk menghubungkan dengan tabel DOKTER. Caranya adalah mengisi properti Connection String, setelah selesai atur properti pada bagian SQL, dan isikan dengan 'SELECT \* FROM Dokter ORDER BY Kategori'.
- Letakkan komponen TquickRep. Tampilkan pita Detail dan PageHeader dari komponen ini.
- Untuk mengelompokkan dokter menurut kategori, Anda memerlukan komponen TQRGroup. Letakkan komponen ini ke TQuickRep. Klik ganda properti Expression sehingga muncul kotak dialog Expression Wizard. Klik Database field dan pilih kategori. Setelah itu klik OK.
- Untuk menampilkan nama kategori, tambahkan komponen TQRExpr pada GroupHeader. Masukkan field kategori pada properti Expression. Caranya seperti saat memasukkan ekspresi pada komponen TQRGroup.
- Berikutnya letakkan beberapa komponen TQRLabel dan TQRDBText pada pita Page Header dan Detail untuk menampilkan record tiap-tiap kategori. Untuk lebih jelasnya lihat Gambar 10.7.

The screenshot shows a report form titled "Form3" with a grid layout. The grid is divided into several sections: a page header section with "DAFTAR DOKTER BERDASARKAN KELOMPOK KATEGORI", a table section with columns for "Kode Dokter", "Nama Dokter", "Alamat", "Telepon", and "Tarif", and a group header section with "ADOQuery1.kategori". The form is designed to be printed as a report, grouped by category.

Gambar 10.7 Hasil Rancangan Laporan Dengan Pengelompokan

- Ubah tampilan Form1, dengan menambahkan satu komponen Tbutton, ubah properti caption menjadi **Contoh Laporan 2** seperti terlihat pada Gambar 10.8, dan tangani kejadian (buat kode program) OnClick dengan pernyataan berikut:

**Form3.QuickRep1.Preview;**

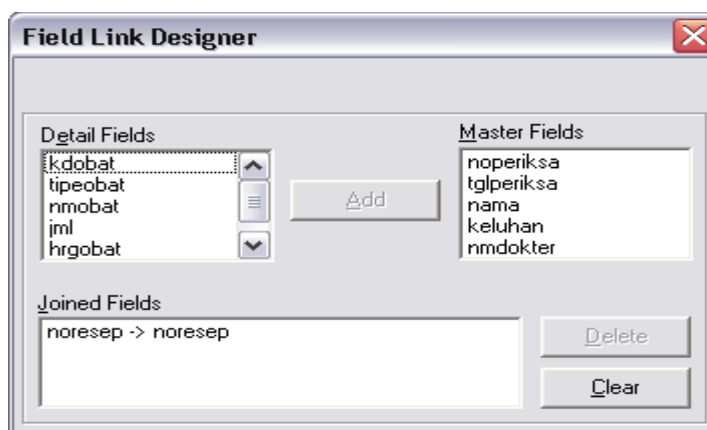


Gambar 10.8 Form Laporan Dengan Dua Pilihan

- Jalankan program

## 5. Membuat Laporan Dengan Master-Detail

- Tambahkan form baru.
- Letakkan dua komponen ADOTable dan sebuah komponen TDataSource. Hubungan kedua komponen ADOTable tersebut masing-masing dengan RESEP dan DETRESEP sedangkan TdataSource harus dihubungkan dengan ADOTable1. Ingat juga untuk menghubungkan properti ADOTable2.Master dan ADOTable2.MasterFields masing-masing dengan DataSource1. dan Noresep.



Gambar 10.9 Kotak Dialog Master Detail

- Berikutnya letakkan komponen TQuickRep serta beberapa komponen TQRLabel (untuk menampilkan tulisan judul-judul) dan TQRDBText (untuk menampilkan data pada fields) yang berhubungan dengan informasi dari RESEP.
- Letakkan komponen TQRSubDetail di bawah komponen pita Detail. Ubahlah properti DataSet dari pita ini menjadi ADOTable2. Sekarang Anda dapat menampilkan DETRESEP pada pita ini. Letakkan beberapa komponen TQRDBText seperti terlihat pada Gambar 10.10.

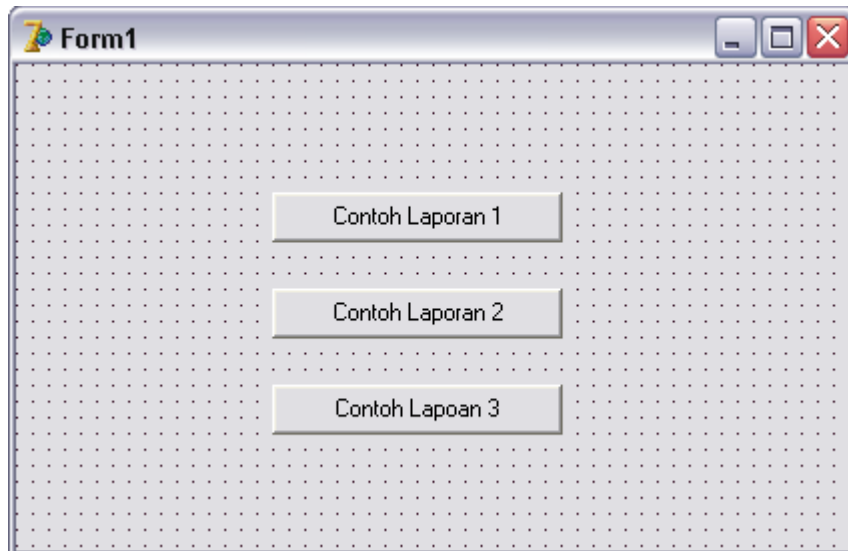
The screenshot shows a Delphi form titled "Form4" with a grid layout. The grid has 8 columns. The layout is as follows:

- Title:** DAFTAR RESEP
- Column Header:**
  - Row 1: No. Resep, noresep
  - Row 2: Tgl Periksa, tglperiksa
  - Row 3: Nama Pasien, nama
  - Row 4: Keluhan, keluhan
  - Row 5: Nama Dokter, nmdokter
- Detail:**
  - Row 6: Kode Obat, Nama Obat, Tipe Obat, Harga, Jumlah, Jml Total
  - Row 7: kdobat, nmobat, tipeobat, hrgobat, jml, jmlTotal
- Sub Detail:** (Empty section below the detail section)

Gambar 10.10 Hasil Rancangan Laporan Master Detail

- Ubah tampilan Form1, dengan menambahkan satu komponen Tbutton, ubah properti caption menjadi **Contoh Laporan 3** seperti terlihat pada Gambar 10.11, dan tangani kejadian (buat kode program) OnClick dengan pernyataan berikut:

**Form4.QuickRep1.Preview;**



Gambar 10.11 Form Laporan Dengan Tiga Pilihan

- Jalankan program

**G. Soal-soal**

4. Apa yang Anda rasakan dan Anda pahami setelah melakukan praktikum sampai pada modul 10 ini?
5. Kendala apa yang Anda alami untuk mempraktekkan proses pembuatan laporan dengan QuickReport yang ada pada Borland Delphi?
6. Silahkan buat laporan untuk tabel-tabel atau queries yang lain berdasarkan database yang Anda miliki.

# MODUL XI

## Membuat Grafik

### H. Pengantar

Pada modul ini akan dilakukan proses penyajian informasi dalam bentuk yang lain berupa grafik (*chart*). Dengan grafik pemahaman akan informasi lebih mudah sehingga keputusan juga akan lebih baik.

Chart adalah komponen pada Borland Delphi 7 yang dipergunakan untuk menampilkan data-data dalam bentuk angka ke dalam bentuk grafik sehingga mudah untuk dipahami. Pada dasarnya Delphi menyediakan 10 buah grafik standard yang dapat dipergunakan.

Jenis-jenisnya adalah sebagai berikut :

- a. Line Bar
- b. Bar
- c. Horizontal Bar
- d. Area
- e. Point
- f. Pie
- g. Fast line
- h. Shape
- i. Gantt
- j. Arrow
- k. Bubble

### I. Praktek

Sebelum memulai praktek untuk modul ini, terlebih dahulu siapkan folder untuk menyimpan pekerjaan pada modul ini. Misalnya disiapkan folder dengan nama SESI-11.

#### 1. Mengenal TeeChart

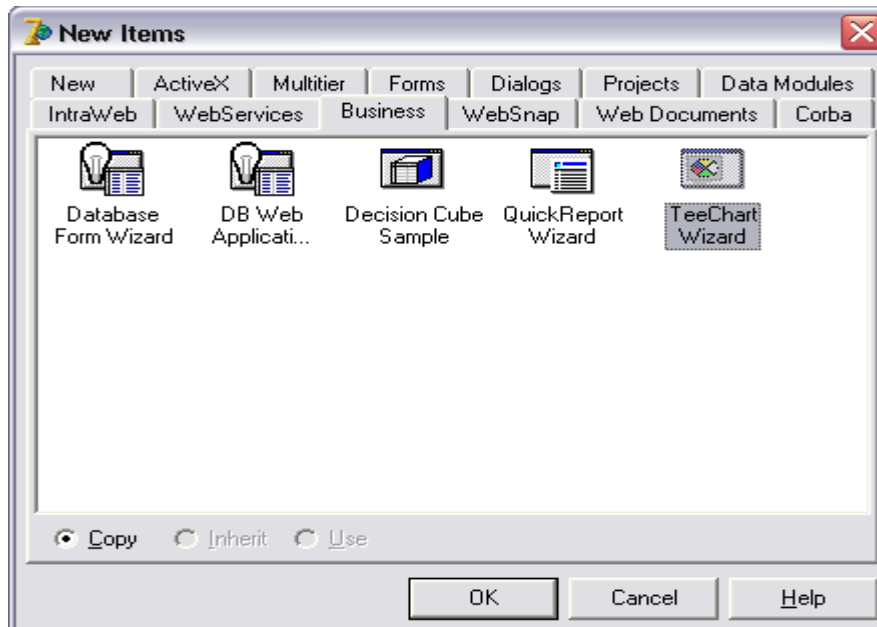
TeeChart adalah sebuah library yang berisi fasilitas-fasilitas untuk membuat grafik. Ada tiga icon TeeChar pada component palette:

- a. Komponen Chart dari page Additional untuk membuat grafik yang non database
- b. Komponen CBChart dari page Data Controls untuk membuat grafik berdasarkan data dari database
- c. Komponen DecisionGraph dari page DecisionCube untuk membuat grafik yang datanya cross tabel.

Cara paling mudah untuk membuat grafik adalah menggunakan TeeChart Wizard.

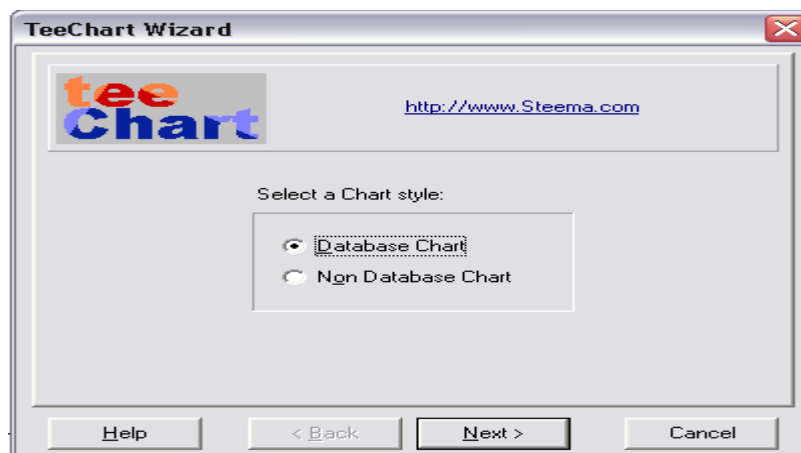
- Buat proyek baru dengan memilih menu **File | New | Other**.
- Pada kotak dialog New Items seperti Gambar 11.1, pilih page **Business**.





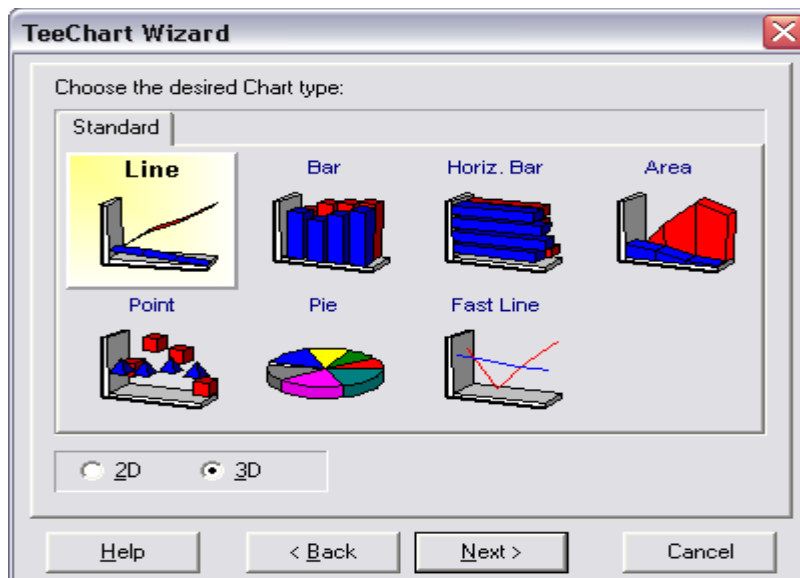
Gambar 11.1 Kotak Dialog New Item Untuk Chart

- Pilih icon TeeChart Wizard dan klik tombol OK, ditampilkan kotak dialog TeeChart Wizard seperti Gambar 11.2.



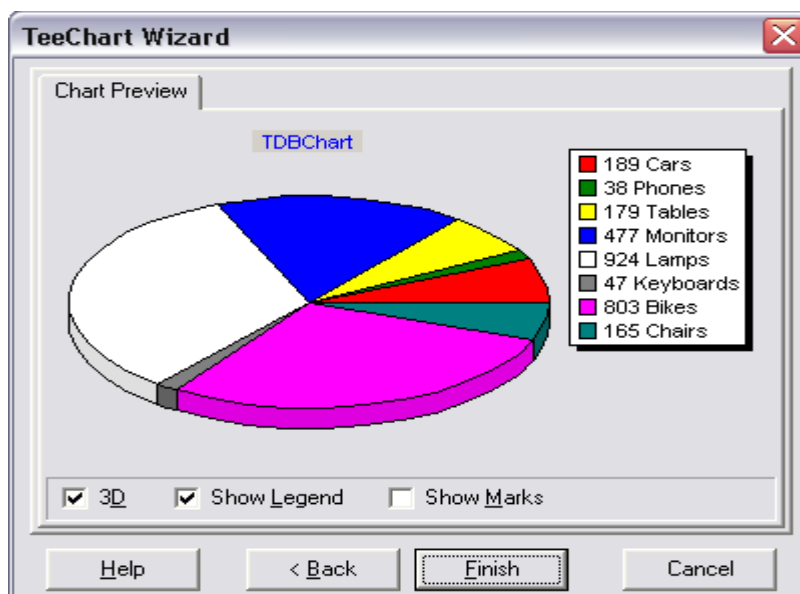
Gambar 11.2 Kotak Dialog TeeChart Wizard

- Sebagai contoh, Anda pilih **Non Database Chart** (karena grafik yang akan dihasilkan adalah grafik yang tidak memanfaatkan *database* sebagai sumber datanya). Lalu klik **Next**. Berikut akan ditampilkan tipe-tipe Chart seperti terlihat pada Gambar 11.3



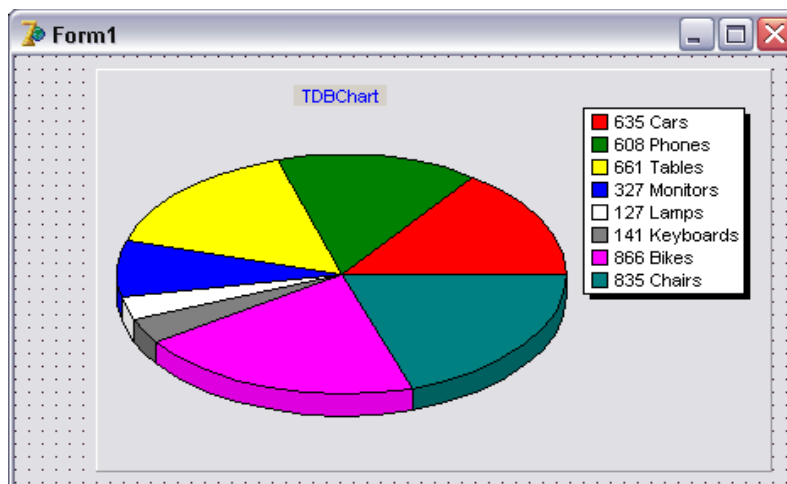
Gambar 11.3 Kotak Dialog Pilihan Jenis TeeChart

- Pilih Pie, lalu klik **Next**. Berikut akan ditampilkan **Chart Preview**, seperti terlihat pada Gambar 11.4.



Gambar 11.4 Kotak Dialog Chart Preview

- Klik **Finish**. Akan ditampilkan form dengan komponen Chart, seperti terlihat pada Gambar 11.5



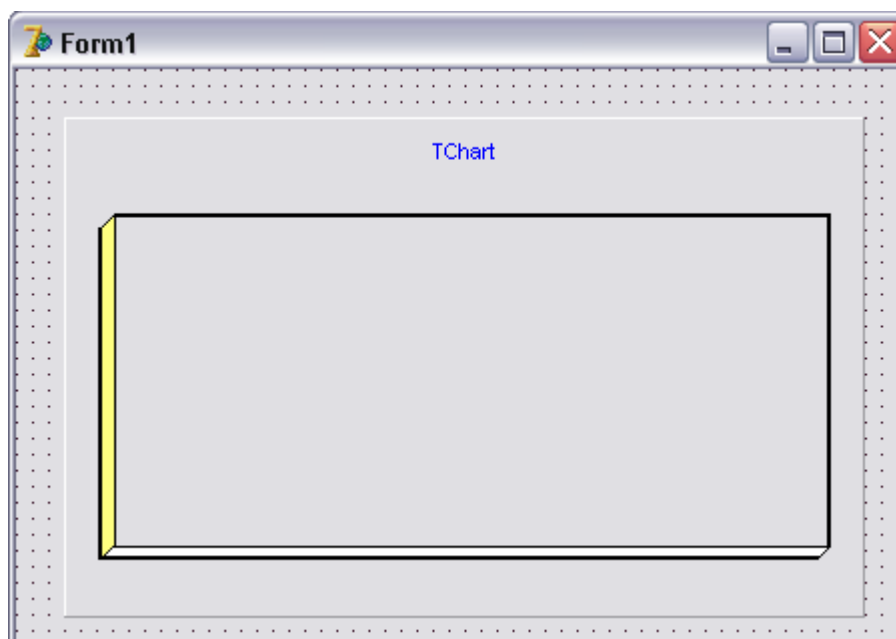
Gambar 11.5 Hasil TeeChart Sederhana

Chart tersebut adalah chart paling sederhana dengan data acak. Selanjutnya Anda dapat mengubahnya dengan mengklik kanan pada komponen Chart dan memilih menu Edit Chart.

## 2. Membuat Chart Secara Manual

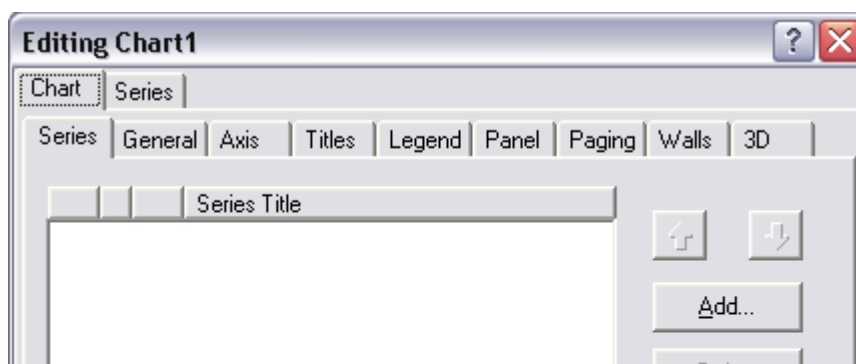
Anda dapat membuat gchart seperti di atas secara manual dengan langkah sebagai berikut:

- Buat proyek baru dengan memilih menu **File | New | Application**.
- Tambahkan komponen Chart dari page **Additional**, sehingga akan ditampilkan chart kosong pada form seperti terlihat pada Gambar 11.6



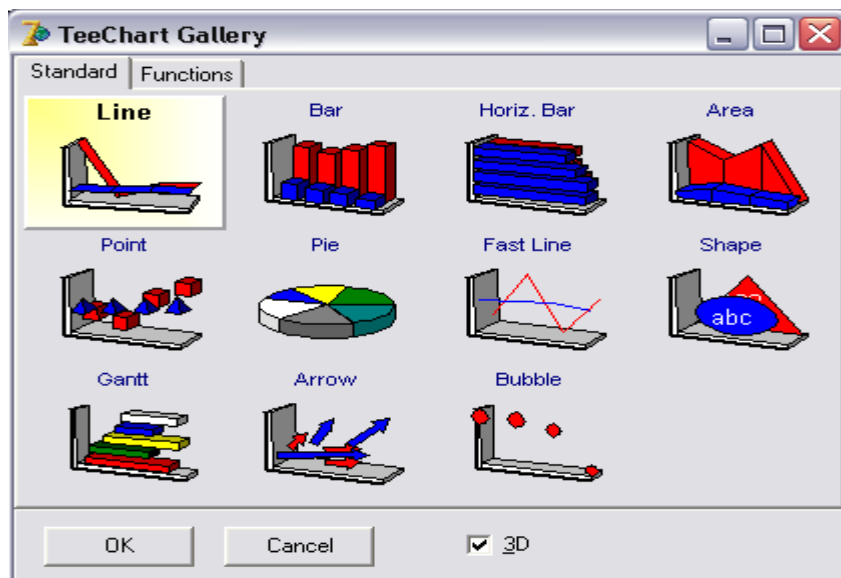
Gambar 11.6 Tampilkan Chart Kosong Pada Form

- Klik kanan pada komponen Chart dan pilih menu Edit Chart, selanjutnya akan ditampilkan kotak dialog Editing Chart, seperti terlihat pada Gambar 11.7.



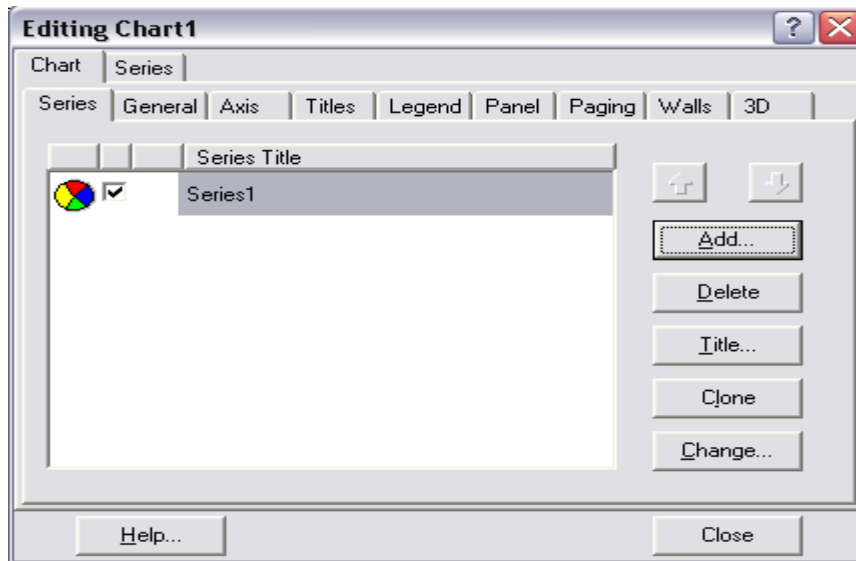
Gambar 11.7 Kotak Dialog Editing Chart

- Saat Anda berada pada tab **Chart | Series**, klik tombol **Add**. Berikut akan ditampilkan kotak dialog TeeChart Gallery (daftar pilihan jenis-jenis grafik) seperti terlihat pada Gambar 11.8.



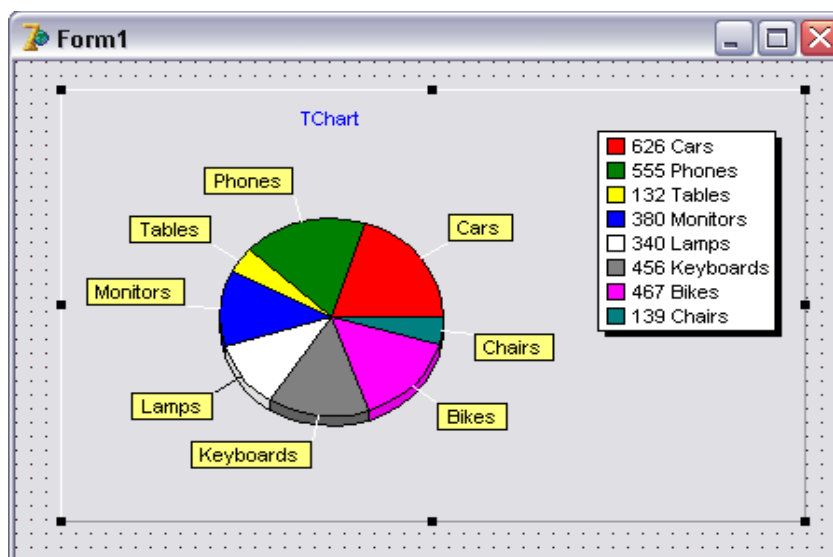
Gambar 11.8 Kotak Dialog Pilihan Chart

- Untuk uji coba, pilih Pie lanjutkan mengklik tombol OK sehingga akan ditampilkan kotak dialog seperti terlihat pada Gambar 11.9.



Gambar 11.9 Kotak Dialog Pengaturan Chart

- Klik tombol Close, dan grafik sederhana dengan data acak ditampilkan di form seperti terlihat pada Gambar 11.10.



Gambar 11.10 Tampilan Akhir Chart Secara Manual

### 3. Menambah Data Secara Manual

Jika ingin menambahkan data secara manual, Anda harus memakai beberapa method dari komponen Series:

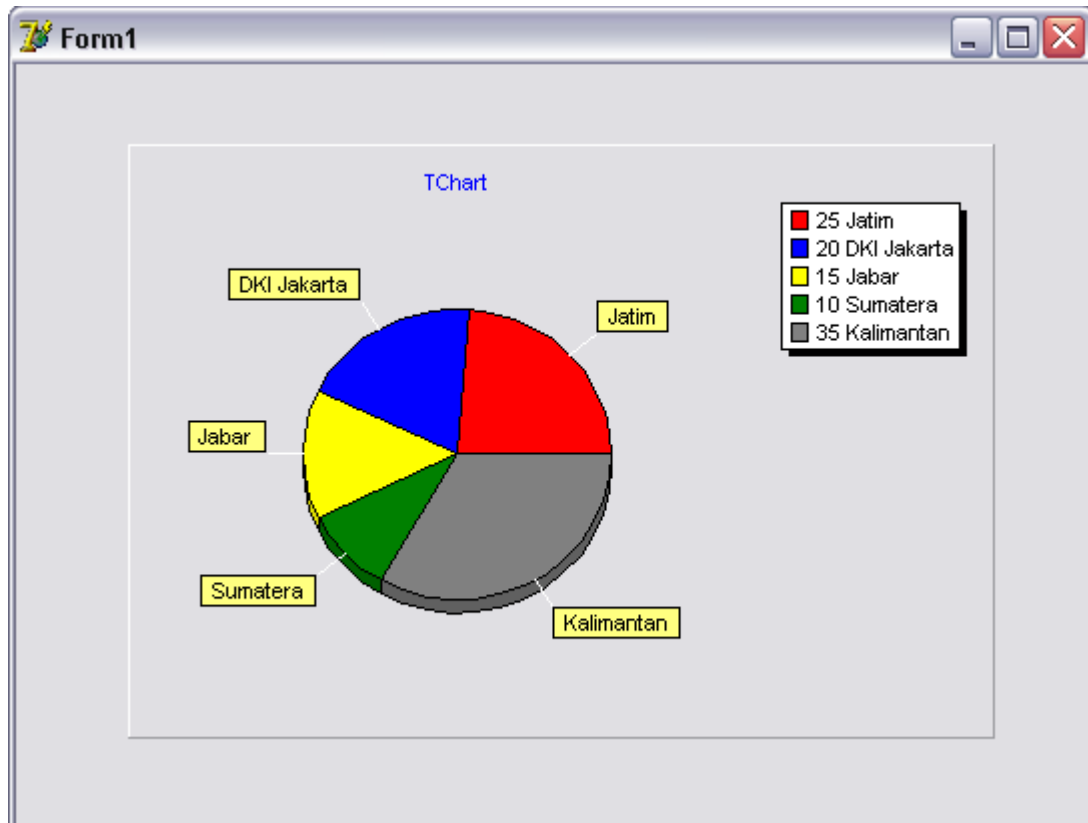
1. Clear untuk menghapus semua data yang ada
2. Add untuk menambah satu data

Isikan event handler OnActive dari form, sehingga kode program seperti ini:

```
procedure TForm1.FormActivate(Sender: TObject);
begin
  with series1 do
  begin
    clear;
```

```
add(25,'Jatim',clRed);  
add(20,'DKI Jakarta',clBlue);  
add(15,'Jabar',clYellow);  
add(10,'Sumatera',clGreen);  
Add(35,'Kalimantan',clGray);  
end;  
end;
```

Jalankan aplikasi (sehingga terlihat seperti Gambar 11.11)

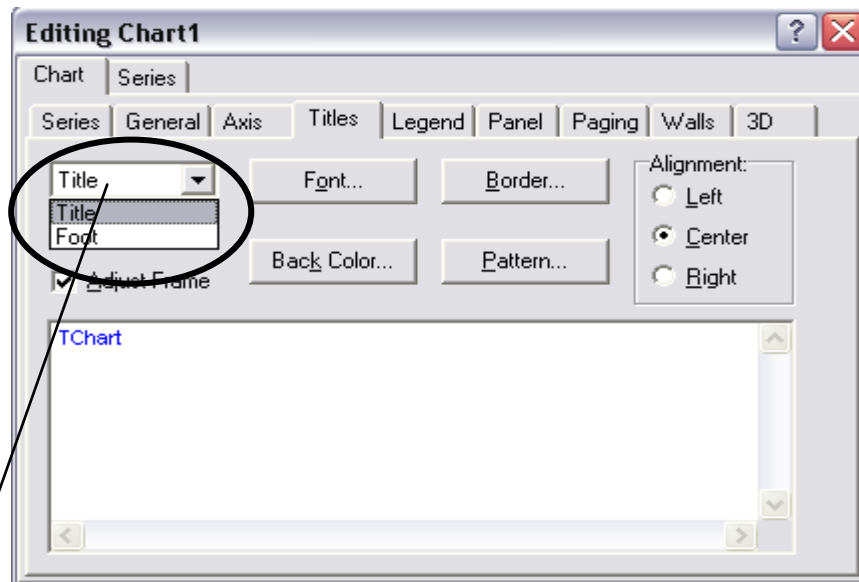


Gambar 11.11 Hasil Running Aplikasi Chart

#### 4. Mengubah Title dan Legend

Title adalah judul yang ditampilkan di atas dan di bawah Chart. Legend adalah keterangan tiap data. Anda dapat mengubahnya dari kotak dialog Editing Chart.

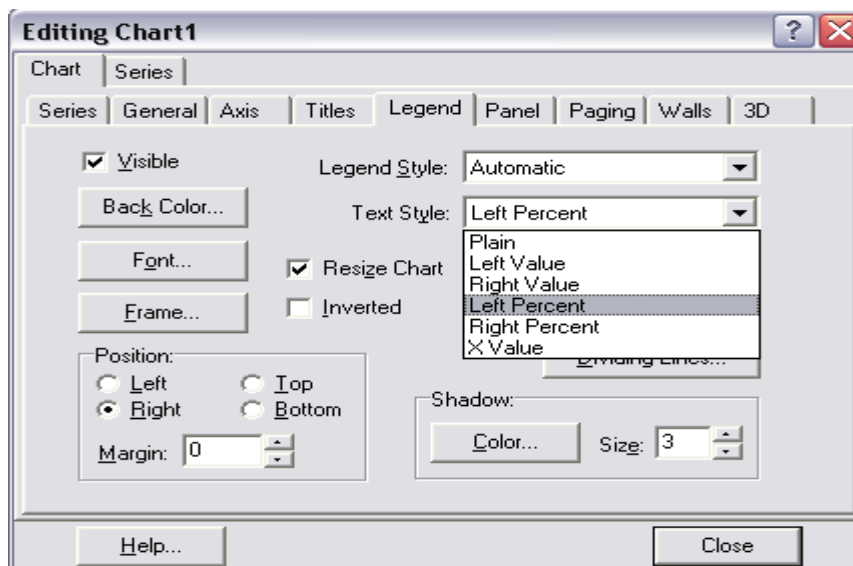
1. Klik kanan pada komponen Chart dan pilih Edit Chart
2. Pilih page Chart | Titles. Anda dapat mengisi Title untuk judul di atas grafik dan Foot untuk judul di bawah grafik.



Pilihan Titles

Gambar 11.12 Kotak Dialog Pengaturan Title

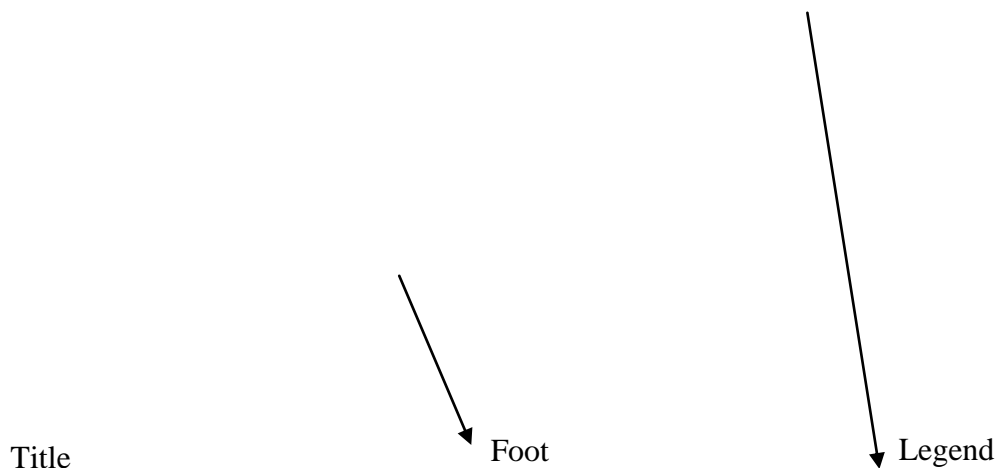
3. Anda dapat mengubah font, warna dan letaknya
4. Klik tab **Legend**, klik kotak Text Style dan pilih Left Percent, setelah selesai klik tombol **Close**



Gambar 11.13 Kotak Dialog Pengaturan Legend

5. Jalankan aplikasi (sehingga terlihat seperti Gambar 11.14)

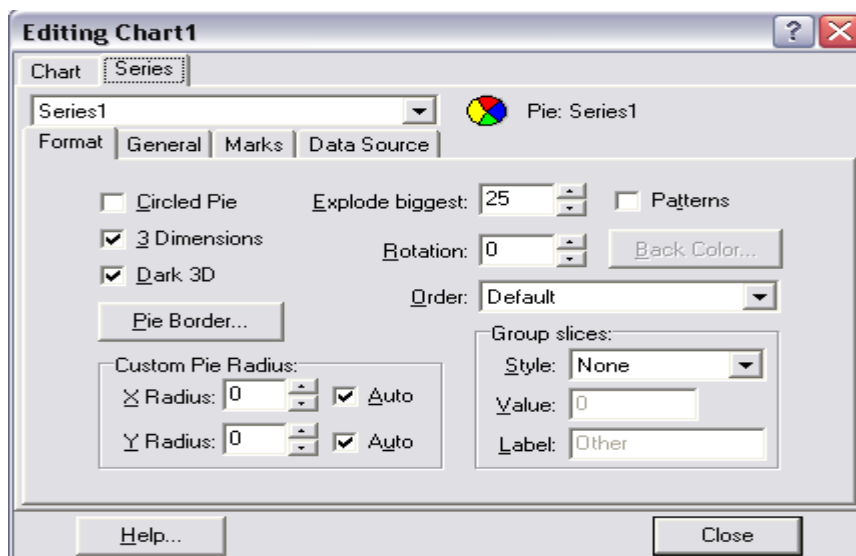




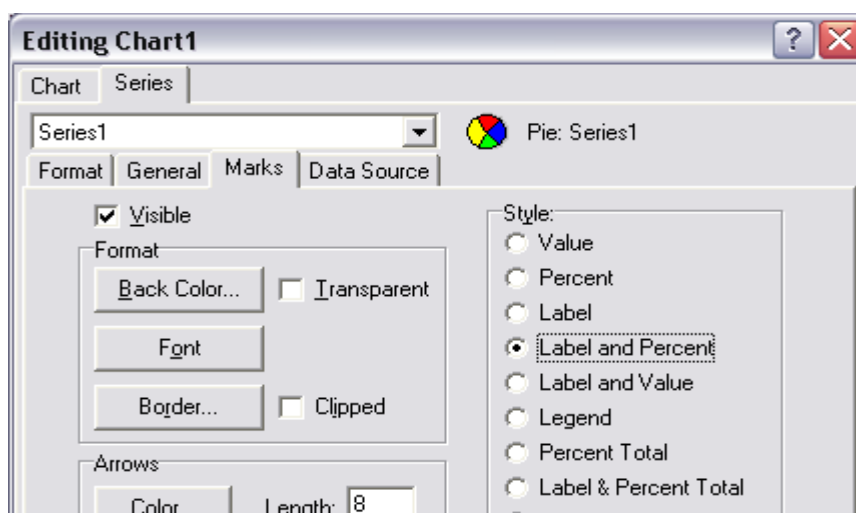
Gambar 11.14 Hasil Running Aplikasi Chart

## 5. Format dan Mark Pada Chart

Ada dua tab utama dari kotak dialog Editing Chart yaitu Chart dan Series. Klik tab Series dan ada beberapa tab, yaitu Format, General, Marks, Data Source. Salah satu hal penting pada tab Format adalah Explode Biggest yang dipakai untuk menonjolkan data terbesar. Ubah menjadi 25%



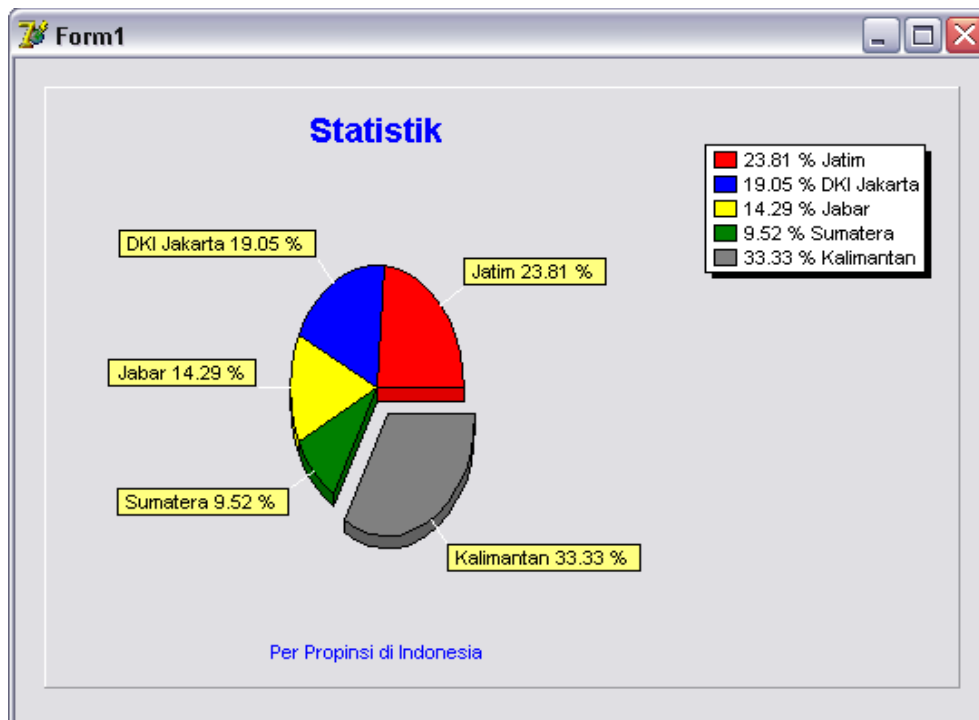
Gambar 11.15 Kotak Dialog Format dan Mark  
Klik tab Mark dan ubah Style menjadi Label dan Percent





Gambar 11.16 Kotak Dialog Mark

Tutup kotak dialog tersebut dan perhatikan perubahannya (baik dalam kondisi design maupun dalam kondisi Run)



Gambar 11.17 Hasil Running Program

## 6. Rotasi dan Zoom

Anda dapat membuat rotasi dan zoom secara dinamis. Hal ini dilakukan dengan pemrograman.

1. Dari aplikasi yang sudah ada, tambahkan komponen Timer dari page System. Ubah interval 100
2. Deklarasikan variabel *n* yang bertipe integer dan variabel tersebut diinisialisasi pada saat form diaktifkan
3. Event handler *OnTimer* dari komponen Timer dipakai untuk merotasikan grafik sebesar 5 derajat dan mengubah persentase zoom.

Listing Lengkap:

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, TeEngine, Series, ExtCtrls, TeeProcs, Chart;
```

```

type
  TForm1 = class(TForm)
    Chart1: TChart;
    Series1: TPieSeries;
    Timer1: TTimer;
    procedure FormActivate(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  n : integer;
implementation
{$R *.dfm}

procedure TForm1.FormActivate(Sender: TObject);
begin
  n:=5;
  with series1 do
  begin
    clear;
    add(25,'Jatim',clRed);
    add(20,'DKI Jakarta',clBlue);
    add(15,'Jabar',clYellow);
    add(10,'Sumatera',clGreen);
    Add(35,'Kalimantan',clGray);
  end;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  series1.Rotate(5);
  with chart1.View3DOptions do
  begin
    zoom:=zoom-n;
    if (zoom<60) or (zoom>=110) then n:=-n;
  end;
end;

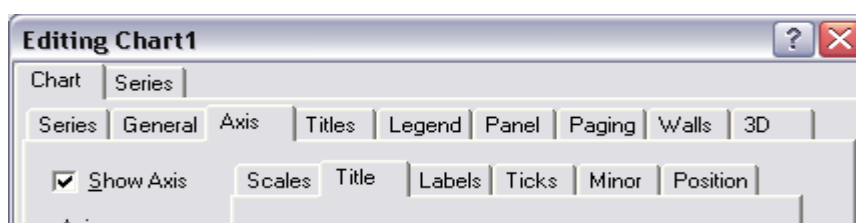
end.

```

## 7. Animasi

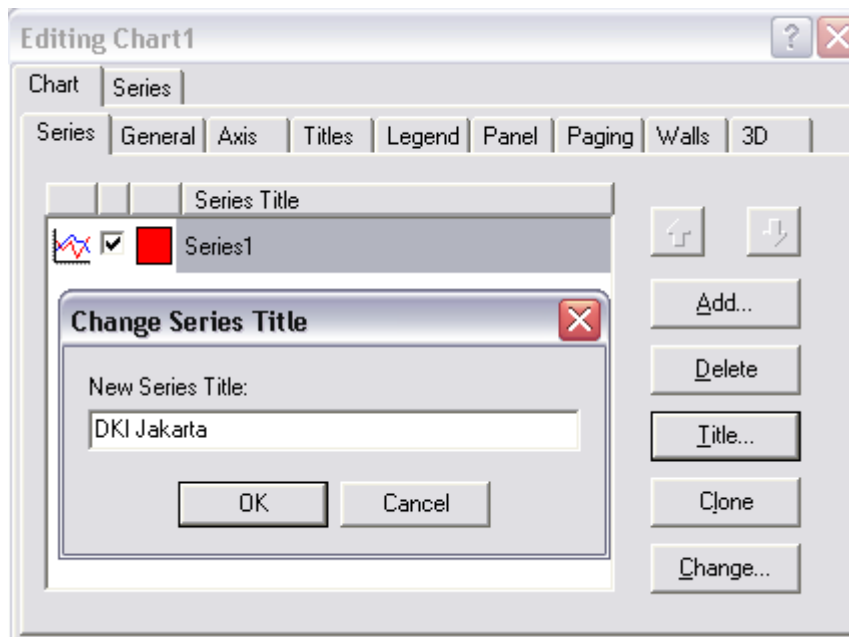
Properti Xvalues dan Yvalues dipakai untuk menyimpan kordinat pada saat aplikasi dijalankan. Properti tersebut dapat diubah isinya sehingga chart akan digambar ulang.

- Buatlah form baru
- Tambahkan komponen chart, klik kanan lalu pilih Edit Chart
- Ubah judul chart menjadi Pemakaian Bahan Bakar. Ubah warna latar belakangnya dan aktifkan bordernya
- Aktifkan tab Axis, dan klik tab Title, isikan Liter untuk judul kiri (Left)



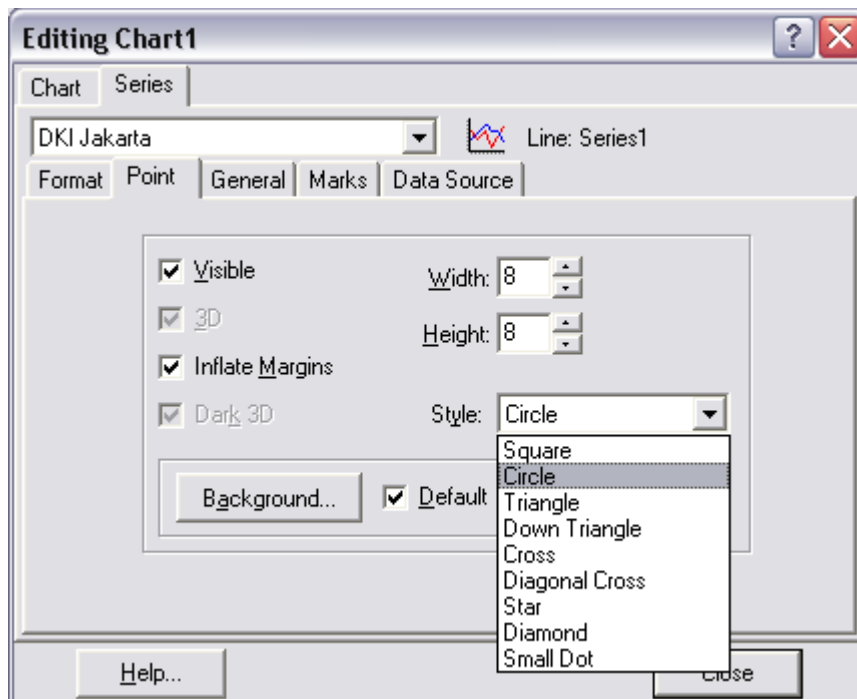
Gambar 11.18 Kotak Dialog Editing Chart

- e. Klik Bottom pada pilihan Axis dan isikan judulnya: Per Pompa Bensin
- f. Pada tab Chart | Series, klik tombol Add untuk menambahkan Series dengan tipe Line
- g. Untuk mengubah judulnya, klik tombol Title .... dan isikan judul baru



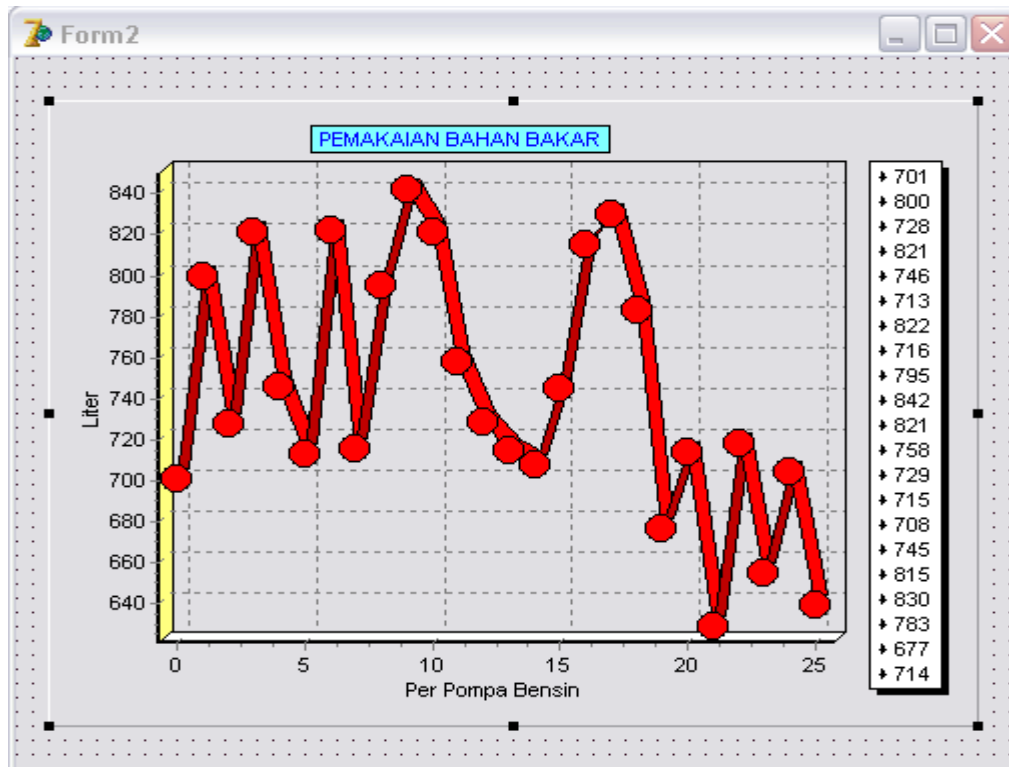
Gambar 11.19 Kotak Dialog Change Series Title

- h. Klik tab Series | Point untuk mengubah bentuk titik-titik pada grafik. Aktifkan Visible dan pilih salah satu style. Ubah width dan height menjadi 8



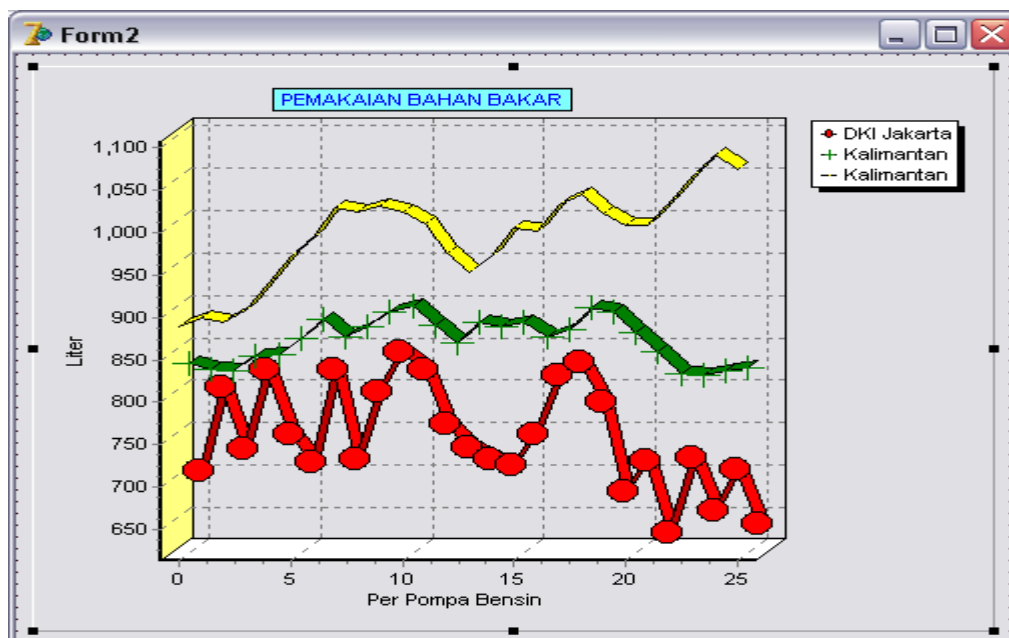
Gambar 11.20 Kotak Dialog Series Point

- i. Grafik sementara menjadi seperti Gambar 11.21 di bawah ini.



Gambar 11.21 Tampilan Sementara Grafik

- j. Tambahkan tiga buah Series lainnya dengan Point | Style yang berbeda-beda



Gambar 11.22 Tampilan Grafik Dengan Tambahan Series

- k. Tambahkan komponen Timer dengan Interval 100  
 l. Pada saat form diaktifkan, setiap Series akan diberi 15 data acak untuk  $x = 1$  sampai dengan 15.

- m. Event handler OnTimer dari komponen Timer dipakai untuk menghapus data pertama (ke-0) dan menambah sebuah data.

Listing Lengkapnya:

```
unit Unit2;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, TeEngine, Series, TeeProcs, Chart;

type
  TForm2 = class(TForm)
    Chart1: TChart;
    Series1: TLineSeries;
    Series2: TLineSeries;
    Series3: TLineSeries;
    Timer1: TTimer;
    procedure Timer1Timer(Sender: TObject);
    procedure FormActivate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation

{$R *.dfm}

procedure TForm2.Timer1Timer(Sender: TObject);
var i : longInt;
tmp : double;
begin
  with chart1 do
    begin
      for i := 0 to seriesCount-1 do
        with Series[i] do
          begin
            tmp:=XValues[1]-XValues[0];
            AddXY(XValues.Last+tmp,YValues.Last+Random(100)-50,"clTeeColor");
          end;
        repaint
      end;
    end;
end;

procedure TForm2.FormActivate(Sender: TObject);
var i,j : integer;
begin
  with chart1 do
    for i :=0 to SeriesCount-1 do
      for j :=1 to 15 do
```

```
Series[i].AddXY(j,Random(100),"clTeeColor");  
end;  
end.
```

**J. Soal-Soal**

1. Apa yang Anda rasakan setelah memanfaatkan fasilitas Grafik pada Borland Delphi?
2. Silahkan buat grafik baru dengan catatan data yang ditampilkan berhubungan dengan penjualan. Misalkan data dilihat berdasarkan tahun. Tahun 2008 jumlah penjualan 4500, tahun 2009 jumlah penjualan 5450, tahun 2010 jumlah penjualan 2350, tahun 2011 jumlah penjualan 3600, tahun 2012 jumlah penjumlahan 1550

## MODUL-XII

### DATA MODUL DAN FORM TRANSAKSI

#### A. Data Modul

Dalam membangun aplikasi *database* menggunakan Delphi, Anda dapat menempatkan komponen-komponen data-access dan kontrol data-aware pada satu form. Penempatan kontrol seperti ini sangat efektif untuk aplikasi sederhana, tetapi menempatkan terlalu banyak kontrol pada satu form akan mempersulit pengelolaan pada saat pembuatan aplikasi yang besar. Karena alasan ini Delphi mengimplementasikan ide “**data module**”, sebuah kontainer komponen-komponen *database* non-visual.

Pada waktu desain, sebuah data modul terlihat seperti sebuah form, tetapi pada saat aplikasi berjalan data module hanya berada di memori, tidak dapat dilihat sebagaimana form. Sebuah data module hanya mempunyai sedikit properties dan *event*. Karena itu data modul hanya digunakan sebagai kontainer.

Sebagaimana form dan frame, data module mempunyai desainer. Ini berarti bahwa data module mempunyai unit obyek pascal seperti seperti form untuk definisi kelasnya (termasuk kode program yang diperlukan) dan file definisi form yang mendaftarkan semua komponen yang ditampilkan beserta propertynya.

Terdapat beberapa alasan untuk menggunakan data module. Alasan pertama adalah untuk saling berbagi pakai (share) komponen-komponen data-access pada beberapa form. Teknik ini digunakan pada aplikasi yang menggunakan banyak form dan saling terkait sehingga diperlukan kemampuan untuk mengakses komponen pada form atau data module lain pada waktu desain (dengan menu File | Use Unit). Alasan kedua adalah untuk memisahkan data dari user interface, memperbaiki struktur suatu aplikasi.

#### B. Form Transaksi

Form transaksi adalah form yang membutuhkan banyak tabel dalam implementasinya. Bahkan dalam proses pengolahan datanya, harus mengacu pada data yang ada pada tabel lain.

Dengan memperhatikan kondisi yang demikian, maka pada form transaksi diperlukan beberapa teknik agar proses bisa berjalan dengan baik dan lebih informatif. Cara yang paling banyak ditemukan adalah proses lookup.

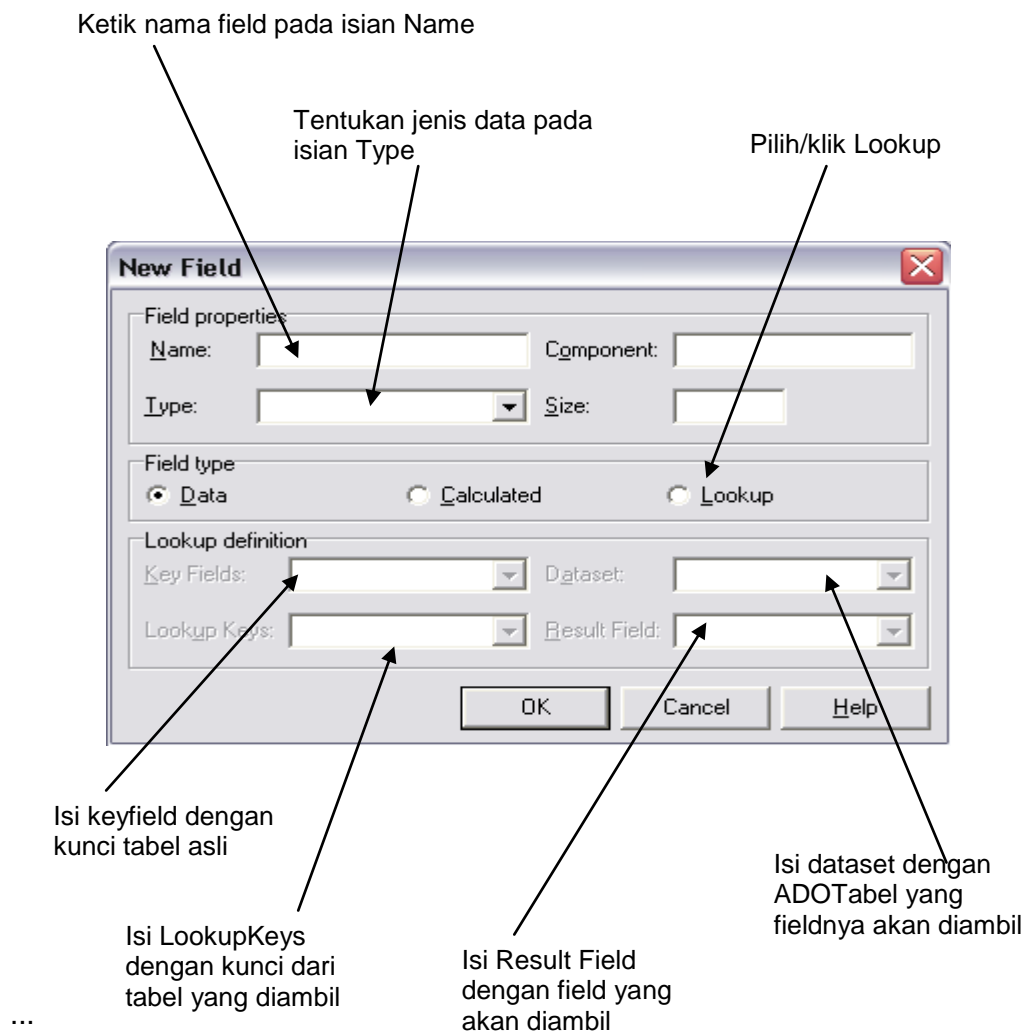
#### C. Praktek (Implementasi)

Kegiatan praktek pada kesempatan ini akan menerapkan data-module dan melakukan pengolahan data untuk form transaksi. Pastikan database (**dbpus**) yang dibuat pada sesi sebelumnya sudah ada pada folder tempat aplikasi akan disimpan. Untuk itu ikutilah langkah sebagai berikut:

1. Buka project baru
2. Saat berada di form desainer (sudah langsung ditawarkan form baru), tambahkan data modul dengan cara : tampilkan menu **File**, pilih **New**, pilih **Other**, pilih **Data Modul**
3. Masukkan komponen **ADO Connection** (diambil dari tab page ADO yang paling kiri) ke data modul, ubah properties connectionstring, masukkan database. Ubah LoginPrompt menjadi False. Ubah Connected menjadi true
4. Masukkan **ADO Table** ke data modul (sesuai jumlah tabel yang digunakan). Untuk kebutuhan praktek ini dibutuhkan tiga buah ADO Table. ADO Table1 untuk data pasien, ADO Table2 untuk data dokter, dan ADO Table3 untuk data periksa.
5. Ubah properties masing-masing ADO Table khususnya di bagian Connection, dan dipilih sesuai ADO Connection yang telah dibuat pada langkah 3. Selain itu diubah juga pada bagian TableName (diisi dengan nama tabel yang sesuai). Active diubah menjadi True
6. Untuk memasukkan field ke masing-masing ADO, klik dua kali pada ADO Table, klik kanan pada kotak yang ditampilkan, pilih Aadd all field. Field pada ADO dapat ditambahkan dengan field yang berasal dari ADO Table yang lain. Caranya adalah klik dua kali pada ADO Table. Saat ditampilkan kotak field, klik kanan, lalu pilih new field. Saat ditampilkan kotak dialog, maka isikan identitas field yang baru. Untuk praktek ini



dibuat field baru yaitu namapas, dan namadok yang ditambahkan ke ADO Table3 (tabel periksa).



**Gambar 12.1 Kotak dialog pembuatan field baru**

7. Sebelum membuat desain form, sebaiknya modul disimpan terlebih dahulu. Untuk menyimpan modul bisa menggunakan nama yang ditawarkan atau dengan memberi nama sesuai keinginan (sebaiknya nama jangan terlalu panjang, karena akan memperlama jika terkait untuk penulisan kode program).
8. Rancanglah form seperti Gambar 12.2. Sebelum menambahkan komponen, gantilah caption dari Form menjadi Data Periksa Pasien.

**Gambar 12.2 Rancangan form transaksi (periksa pasien)**

Komponen-komponen yang ditambahkan:

- Label1, caption diganti dengan Nomor Periksa
- Edit1
- Label2, caption diganti dengan Tanggal Periksa
- DateTimePicker (diambil pada tab page Win32)
- Label3, caption diganti dengan Pasien
- Combobox1
- Edit2, properties visible diganti false
- Label4, caption diganti dengan Keluhan
- Edit3
- Label5, caption diganti dengan Dokter
- Combobox2
- Edit4, properties visible diganti false
- DBGrid, dihubungkan dengan datasource yang ada di data modul
- Enam buah Button, berturut-turut:
  - Button1 : Caption diganti : B&aru dan Name diganti dengan TblBaru
  - Button2 : Caption diganti : &Simpan dan Name diganti dengan TblSimpan
  - Button3 : Caption diganti : &Batal dan Name diganti dengan TblBatal
  - Button4 : Caption diganti : &Ubah dan Name diganti dengan TblUbah
  - Button5 : Caption diganti : &Hapus dan Name diganti dengan TblHapus
  - Button6 : Caption diganti : &Keluar dan Name diganti dengan TblKeluar

9. Jika pada form dibutuhkan dbgrid, maka diperlukan komponen datasource. (datasource bisa ditempatkan di Form atau di datamodul). Untuk kebutuhan praktek ini dibutuhkan satu datasource yang dihubungkan dengan tabel periksa. Untuk menghubungkan form dengan datamodul, maka lakukan langkah : tampilkan menu File, pilih UseUnit, pilih modul yang akan dihubungkan, akhiri dengan mengklik tombol OK
10. Tambahkan atau ketikkan db pada bagian uses, sehingga pada bagian uses berubah secara lengkap menjadi :

```
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, DBGrids, StdCtrls, ComCtrls, db;
```



Yang ditambahkan

11. Deklarasikan variabel pada posisi Var sebelum Implementation dan tulis sebagai berikut:

```
tambahdata:boolean;
```

12. Tulis kode program pada obyek Form event Active (nomor baris tidak diketik):

```
1   tblsimpan.Enabled:=false;
2   tblbatal.Enabled:=false;
3   edit1.Enabled:=false;
4   datetimepicker1.Enabled:=false;
5   combobox1.Enabled:=false;
6   edit3.enabled:=false;
7   combobox2.Enabled:=false;
8   datamodule2.ADOTable1.Close;
9   datamodule2.ADOTable1.Open;
10  datamodule2.ADOTable2.Close;
11  datamodule2.ADOTable2.Open;
12  datamodule2.ADOTable3.Close;
13  datamodule2.ADOTable3.Open;
14  combobox1.Clear;
15  while not datamodule2.ADOTable1.Eof do
16  begin
17      combobox1.AddItem(datamodule2.ADOTable1nama.AsString, combobox1);
18      datamodule2.ADOTable1.Next;
19  end;
20  combobox2.Clear;
21  while not datamodule2.ADOTable2.Eof do
22  begin
23      combobox2.AddItem(datamodule2.ADOTable2nmdokter.AsString, combobox2);
24      datamodule2.ADOTable2.Next;
25  end;
```

13. Tulis kode program pada obyek Combobox1 event Change

```
1   if
    datamodule2.ADOTable1.Locate('nama', combobox1.Text, [lopartialkey, locaseinsensitive]) then
    edit2.Text:=datamodule2.ADOTable1kdpasien.AsString;
```

14. Tulis kode program pada obyek Combobox2 event Change

```
1   if
    datamodule2.ADOTable2.Locate('nmdokter', combobox2.Text, [lopartialkey, locaseinsensitive]) then
    edit4.Text:=datamodule2.ADOTable2kddokter.AsString;
```

15. Tulis kode program pada obyek TblBaru event Click

```
1   tblbaru.enabled:=false;
2   tblsimpan.enabled:=true;
3   tblbatal.Enabled:=true;
4   tblubah.Enabled:=false;
```

```

5      tblhapus.enabled:=false;
6      tblkeluar.Enabled:=false;
7      edit1.Enabled:=false;
8      datetimepicker1.Enabled:=true;
9      combobox1.Enabled:=true;
10     edit3.enabled:=true;
11     combobox2.Enabled:=true;
12     tambahdata:=true;
13     datamodule2.ADOTable3.Last;
14     no:=strtoint(copy(datamodule2.ADOTable3noperiksa.AsString,3,8));
15     edit1.Text:='P-'+formatfloat('0000000#',no+1);

```

## 16. Tulis kode program pada obyek TblSimpan event Click

```

1      if      tambahdata      then      datamodule2.ADOTable3.Append      else
      datamodule2.ADOTable3.Edit;
2      datamodule2.ADOTable3noperiksa.AsString:=edit1.Text;
3      datamodule2.ADOTable3tglperiksa.AsDateTime:=DateTimePicker1.Date;
4      datamodule2.ADOTable3kdpasien.AsString:=edit2.Text;
5      datamodule2.ADOTable3keluhan.AsString:=edit3.Text;
6      datamodule2.ADOTable3kddokter.AsString:=edit4.Text;
7      datamodule2.ADOTable3.Post;
8      tambahdata:=false;
9      tblbaru.enabled:=true;
10     tblsimpan.enabled:=false;
11     tblbatal.Enabled:=false;
12     tblubah.Enabled:=true;
13     tblhapus.enabled:=true;
14     tblkeluar.Enabled:=true;
15     edit1.Enabled:=false;
16     datetimepicker1.Enabled:=false;
17     combobox1.Enabled:=false;
18     edit3.enabled:=false;
19     combobox2.Enabled:=false;

```

## 17. Tulis kode program pada obyek TblBatal event Click

```

1      tblbaru.enabled:=true;
2      tblsimpan.enabled:=false;
3      tblbatal.Enabled:=false;
4      tblubah.Enabled:=true;
5      tblhapus.enabled:=true;
6      tblkeluar.Enabled:=true;
7      edit1.Enabled:=false;
8      datetimepicker1.Enabled:=false;
9      combobox1.Enabled:=false;
10     edit3.enabled:=false;
11     combobox2.Enabled:=false;

```

## 18. Tulis kode program pada obyek TblUbah event Click

```

1      tblbaru.enabled:=false;
2      tblsimpan.enabled:=true;
3      tblbatal.Enabled:=true;
4      tblubah.Enabled:=false;
5      tblhapus.enabled:=false;
6      tblkeluar.Enabled:=false;
7      tambahdata:=false;
8      edit1.Enabled:=true;
9      datetimepicker1.Enabled:=true;
10     combobox1.Enabled:=true;
11     edit3.enabled:=true;
12     combobox2.Enabled:=true;

```

#### 19. Tulis kode program pada obyek TblHapus event Click

```
1  if (Application.MessageBox('Benar mau di hapus ?', 'Konfirmasi',  
2  MB_YESNO +MB_ICONQUESTION)=idyesh) then  
3  begin  
4  datamodule2.ADOTable3.Delete;  
end;
```

#### 20. Tulis kode program pada obyek TblKeluar event Click

```
1  Application.Terminate;
```

#### 21. Tulis kode program pada obyek Grid1 event OnCellClick

```
1  edit1.Text:=datamodule2.ADOTable3noperiksa.AsString;  
2  DateTimePicker1.Date:=datamodule2.ADOTable3tglperiksa.AsDateTime;  
3  datamodule2.ADOTable1.Locate('kdpasien', datamodule2.ADOTable3kdpasien.AsString, []);  
4  combobox1.Text:=datamodule2.ADOTable1nama.AsString;  
5  edit3.Text:=datamodule2.ADOTable3keluhan.AsString;  
6  datamodule2.ADOTable2.Locate('kddokter', datamodule2.ADOTable3kddokter.AsString, []);  
7  combobox2.Text:=datamodule2.ADOTable2nmdokter.AsString;
```

Catatan khusus:

**Jika pencarian menggunakan Locate, maka harus ditambahkan library db pada bagian uses.**

**Jika grid akan digunakan sebagai navigator, maka supaya saat grid diklik bisa disorot satu baris, lakukan pengaturan properties yaitu pada bagian Option diganti dbRowSelect menjadi True**

## DAFTAR PUSTAKA

- Husni, 2004, Pemrograman Database dengan Delphi, Graha Ilmu, Yogyakarta
- Ichwan, M., 2011, Pemrograman Basis Data Delphi 7 danMySQL, Informatika, Bandung.
- Indriyawan, E., 2005, Pemrograman Database Meningkatkan kemampuan database dengan menggunakan Delphi, Andi, Yogyakarta
- Kadir, A., 2001, Pemrograman *Database* Menggunakan Delphi, Salemba Infotek, Jakarta.
- Martina, I., 2004, 36 Jam Belajar Komputer Pemrograman Visual Borland Delphi 7, PT Elex Media dan Wahana Komputer
- Zukhri, Z., 2005, Delphi 2005 Pemrograman Visual bagi pemula, Graha Ilmu, Yogyakarta