

Modul VII Menghubungkan Tabel Database ke Program Aplikasi

A. Menyiapkan Komponen Pengait Data

Komponen pengait data berfungsi untuk menghubungkan program aplikasi yang kita bangun dengan tabel-tabel yang ada di dalam database kita. Untuk keperluan ini, Delphi menyediakan sepasang komponen, masing-masing berfungsi sebagai;

1. DataSet, dan
2. DataSource.

Komponen DataSet yang digunakan untuk mengambil data dari tabel tertentu tergantung pada format database yang kita gunakan. Jika kita menggunakan database Paradox yang dikelola dengan Database Desktop, maka komponen DataSet-nya diambil dari pallet BDE. Jika kita menggunakan database berformat Microsoft Access, DataSetnya diambil dari pallet ADO. Dalam praktikum ini, kita menggunakan database MySQL.

Untuk mengakses database MySQL sebenarnya ada dua pilihan yang tersedia, pertama menggunakan MySQLODBC dan yang kedua DAC for MySQL. Kedua tools ini dibuat oleh Pendor yang berbeda. DAC for MySQL misalnya, software ini dapat diambil di internet dalam bentuk versi shareware. Alamat situsnya adalah <http://www.microolap.com>. Jika kita men-download tools ini dan menginstalkannya ke komputer kita, pada **Component Pallet** akan tersedia sebuah *pallet* bernama **DAC for MySQL**.

DataSet yang digunakan untuk menghubungkan table dataset ke program aplikasi secara umum ada dua jenis, yaitu; **TTable** dan **TQuery**. Pada pallet DAC for MySQL, kedua komponen ini masing-masing bernama MySQLTable dan MySQLQuery. **MySQLTable** digunakan untuk mengakses table database menggunakan fungsi-fungsi Delphi sedangkan **MySQLQuery** untuk mengakses table database menggunakan perintah-perintah SQL. Di bawah ini akan dicontohkan cara mengakses table MySQL menggunakan komponen MySQLTable.

Selain DataSet pasangan komponen ini untuk menghubungkan tabel database ke program aplikasi adalah komponen DataSource. Komponen ini sama untuk seluruh format database dan seluruh komponen DataSet. Dan komponen ini diambil dari *pallet* **DataSource**.

Berikut ini akan dicontohkan cara menghubungkan table mhs yang ada di dalam database akademik yang telah anda buat melalui kegiatan praktikum Modul V. Langkah-langkahnya adalah sebagai berikut;

1. Jalankan Delphi dan buka project **prjAkademik** yang telah anda buat melalui kegiatan praktikum Modul VI,
2. Tampilkan data modul Dm,
3. Ambil komponen **MySQLTable** dari pallet **DAC for MySQL** dan letakkan pada data modul,
4. Ganti property namanya menjadi **tbMHS**,
5. Isi *property Database*-nya dengan **dbServer** dan *property TableName*-nya dengan **mhs**,
6. Ambil komponen DataSource dari pallet **DataSource** dan letakkan besebelahan dengan *tbMHS*. Ganti property name-nya menjadi dsMHS,
7. Isi *property DataSet*-nya dengan **tbMHS**,
8. Ketika anda mengisi property TableName komponen tbMHS, *property connected dbServer* akan terisi dengan nilai True, oleh karena itu disarankan agar anda mengembalikan nilai property ini menjadi False setiap kali sebelum program dijalankan.
9. Simpan kembali file-file project anda. Dan jika anda menjalankan program anda, anda belum melihat terjadi perubahan apa-apa pada program anda, karena anda baru berhasil menghubungkan tabel database ke program aplikasi, tetapi anda belum mempunyai tempat untuk menampilkan data-data yang dibaca oleh komponen-komponen pengait data di atas. Oleh karena itu, praktikum selanjutnya anda akan diajarkan cara membuat form interface dan meletakkan komponen-komponen DataControl di dalamnya untuk menampilkan dan sekaligus menginputkan data ke table database (tabel mhs).

B. Menyiapkan Form Interface

Form interface yang disiapkan melalui kegiatan praktikum berikut adalah form interface ke table mhs. Fungsi form ini selain untuk menampilkan data yang ada di table database juga sekaligus dapat berfungsi untuk menginputkan data ke table mhs.

Langkah-langkah yang harus anda lakukan adalah sebagai berikut;

1. Tambahkan form baru ke project anda (prjAkademik),
2. Ganti *property name* form tersebut menjadi **frmMHS**,
3. Design tampilan *form* tersebut seperti dicontohkan pada Gambar 7.1 di bawah ini;
4. Simpan unit pasangan **frmMHS** ini dengan nama **uMHS.pas** di folder **akademik** (folder yang anda pada Modul Praktikum VI),
5. Tampilkan frmMenu, dan tulis kode program di bawah ini (kode program untuk menampilkan form frmMHS) pada *event on click* menu mnuMHS;

```

procedure TfrmMenu.mnuMhsClick(Sender: TObject);
begin
    frmMHS.ShowModal;
end;

```

Gambar 7.1 Contoh Form Interface ke Table mhs

Keterangan:

- Komponen yang bercaptionkan; Urutkan Tampilan, Cari Data, Input Data, dan Browsing adalah komponen GroupBox. Property name komponen ini boleh tidak anda ganti karena memang dalam contoh ini kita tidak akan menuliskan kode program untuk komponen-komponen tersebut.
- Komponen DataControl untuk mengakses field **Jenis Kelamin**, **Jurusan**, dan **Program Studi** adalah komponen **DBComboBox**. Komponen ini berbeda dengan

component ComboBox. Kalau komponen ComboBox tergolong pada komponen Standard maka komponen DBComboBox tergolong pada komponen DataControl (komponen untuk mengakses tabel database). Untutan nama index (DBComboBox1 – DBComboBox3) komponen ini mulai dari jenis kelamin sampai ke program studi.

- Komponen yang berada di dalam GroupBox yang bercaptionkan Browsing adalah komponen DBGrid. Komponen ini juga tergolong pada komponen DataControl yang diambil dari pallet DataControl.
 - Komponen yang berada di sudut kiri bawah form adalah komponen DBNavigator. Komponen ini juga tergolong pada komponen DataControl. Selain komponen **DBNavigator**, **DBGrid**, dan **DBComboBox**, komponen DataControl yang lain yang juga digunakan pada form interface di atas adalah komponen DBEdit.
7. Tampilkan kode program file unit uMHS dan tambahkan kode program berikut di bawah kata cadangan Implementation.

```
uses uDm;
```

Perintah di atas diberikan dengan tujuan agar komponen-komponen yang ada pada unit uDm dapat dihubungkan dengan komponen-komponen yang ada pada unit uMHS.

Contoh cara me-uses-kan unit uDm pada unit uMHS;

```
implementation
uses uDm;

{$R *.dfm}
end.
```

8. Tampilkan kembali *form frmMHS*, isi seluruh *property* DataSource komponen DataControl (**DBEdit**, **DBComboBox**, **DBGrid**, dan **DBNavigator**) dengan **Dm.dsMHS**,
9. Isi *property DataField* komponen **DBEdit1** dengan **nobp**, **DBEdit2** dengan **nama**, dan begitu seterusnya sampai pada **DBEdit5**, sesuai dengan nama field table mhs yang akan ditampilkan pada masing-masing komponen DataControl tersebut.
10. Untuk menyediakan opsi/pilihan jenis kelamin (Laki-laki dan Perempuan) pada DBComboBox1, klik tandan “...” di sebelah kanan property Items komponen tersebut. Pada *String List Editor*-nya ketik Laki-laki dan Perempuan secara berbaris (dipisahkan oleh ENTER antara nilai yang satu dengan nilai yang lainnya).
11. Tulis kode program pada event on active form frmMHS untuk mengaktifkan tabel mhs. Kode programnya adalah sebagai berikut;

Dm.tbMHS.Open;

Dengan perintah di atas, table mhs baru akan aktif setelah form frmMHS aktif (tampil).

12. Tulis kode program untuk menutup table mhs pada event on close frmMHS. Kode programnya adalah;

Dm.tbMHS.Close;

13. Simpan kembali file-file project anda dan coba menjalankan program, dan menakses perintah yang baru saja ditambahkan.

C. Membuat Index pada Tabel

Index pada table database berfungsi untuk;

1. Mengurutkan tampilan data berdasarkan field yang diindeks terwebut, dan
2. Memudahkan dalam pencarian data.

Desing form (interface) yang ada pada gambar 7.1 di atas menyediakan fasilitas untuk menampilkan data terurut berdasarkan field tertentu. Untuk dapat merencanakan program seperti dikehendaki oleh tampilan, kita harus membuat indeks pada table mahasiswa berdasarkan pada field-field yang memang sudah kita rencanakan untuk dapat ditampilkan datanya secara terurut.

Sebenarnya pada table mhs yang anda buat pada Modul Praktikum VI telah memiliki indeks, yaitu indeks primer. Indeks primer adalah indeks yang terbentuk ketika kita merancang table dan menetapkan salah satu fieldnya sebagai kunci primer. Jadi table mhs tersebut secara otomatis telah terindeks (terurut) berdasarkan field nobp, karena field inilah yang menjadi kunci primer bagi table mhs.

Jika kita membuat indeks berdasarkan field lain selain kunci primer, maka indeks yang kita buat tersebut disebut indeks skunder. Berikut ini akan dicontohkan cara membuat indeks skunder pada table mhs berdasarkan field nama. Langkah-langkahnya adalah sebagai berikut;

1. jalankan MySQL client dari computer anda dan login ke database server,
2. Aktifkan database akademik yang telah anda buat pada Modul Praktikum V,
3. Jika kita ingin membuat indeks berdasarkan field nama pada table mhs dengan nama indeks **namamhsidx**, perintahnya adalah;

create index namamhsidx on mhs(nama);

4. Melalui contoh di atas, coba anda buat sendiri index di table mhs dengan nama **alamatmhsidx** berdasarkan field alamat.

D. Kode Program untuk Menampilkan Data Secara Urut

Pada form interface Gambar 7.1, komponen yang digunakan untuk menentukan urutan tampilan adalah komponen ComboBox1. Sesuai dengan indeks yang telah kita buat melalui praktikum di atas, di table mhs telah ada satu buah indeks primer dan ditambah dua buah indeks skunder. Indeks primer adalah indeks yang dibuat berdasarkan kunci primer nama indeks ini adalah null (ditulis ''). Sedangkan dua indeks skunder yang lainnya masing-masing bernama namamhsidx dan alamatmhsidx berdasarkan field nama dan alamat. Untuk itu, pada ComboBox1, kita sediakan juga item sebanyak tiga buah, masing-masing berdasarkan NOBP, Nama, dan Alamat. Untuk membuat item di ComboBox1, lakukan langkah-langkah berikut ini;

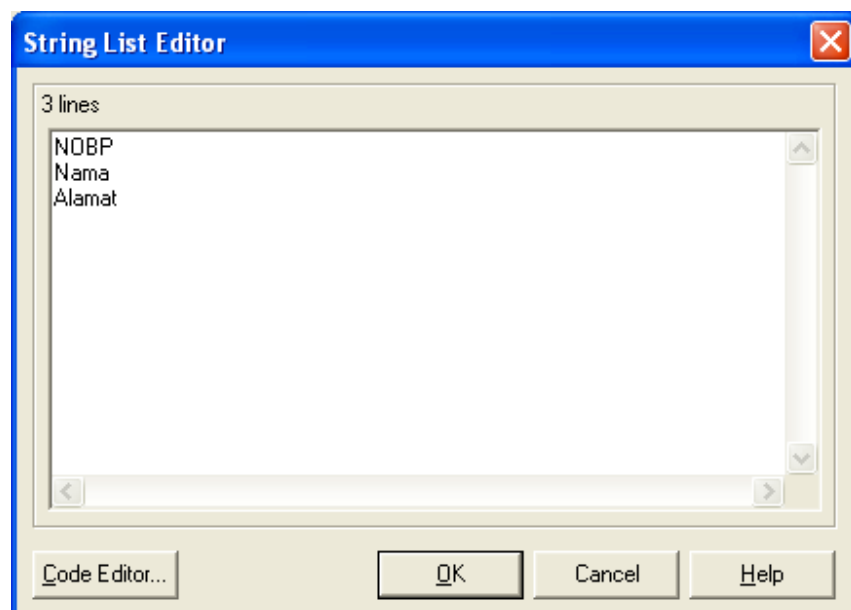
1. Tampilkan form frmMHS,
2. Sorot ComboBox1
3. Cari property Items di Object Inspector, klik tanda ... di sebelah kanan property Items tersebut,
4. Pada **String List Editor**, ketik ketiga nama field yang akan dijadikan sebagai dasar pengurutan (NOBP, Nama, dan Alamat).

Catatan:

- Huruf besar atau huruf kecil pada field tersebut tidak jadi soal karena nilai property Items tidak akan kita acui didalam

penulisan kode program. Yang kita acui adalah nilai *property Item Index*.

- Nilai *property Item Index* dimulai dari angka 0. Jadi **NOBP** yang kita tuliskan pertama kali, nilai tersebut jika dilihat dari urutannya berada di urutan pertama, maka nilai *Item Index*-nya sama dengan 0.



Gambar 7.2 Isi *Property ItemIndex ComboBox1*

5. Isi nilai *property ItemIndex ComboBox1* di **Object Inspector** dengan nilai **0**. Perhatikan perubahan tampilan komponen tersebut!
6. Agar ketika user mengklik **ComboBox1** dan memilih **NOBP** atau **Nama**, maka data tampil secara terurut berdasarkan field-field tersebut, maka ketik kode program untuk memanggil nama indeks yang kita buat (sesuai dengan nilai *property Item index ComboBox1*) yang dipilih user saat itu. Berikut ini adalah contoh kode program untuk memanggil indeks-indeks tersebut;

```
if ComboBox1.ItemIndex = 0 then
    Dm.tbMHS.IndexName := ""
Else if ComboBox1.ItemIndex = 1 then
```

```

Dm.tbMHS.IndexName := namamhsidx
else
Dm.tbMHS.IndexName := alamatmhsidx;

```

Tulis kode program di atas pada *event on change* **ComboBox1**. Ingat bahwa nama indeks yang anda panggil adalah nama indeks yang telah anda buat di database. Penamaannya harus *Case Sensitive*.

7. Simpan kemabli file-file project anda, dan coba jalankan. Yang harus anda lihat pada tahapan ini adalah data di table mhs anda harus telah dapat ditampilkan secara berutur berdasarkan field; nobp, nama, dan alamat. Lakukan pengujian pada program anda untuk melihat keberhasilan tersebut.
8. Jika langkah kerja 7 telah berhasil, untuk latihan, buatlah indeks di table mhs berdasarkan field jurusan dan program studi. Nama indeksnya terserah anda. Setelah itu, tulis kode programnya pada event on change ComboBox1 (di frmMHS) untuk dapat menampilkan data dari table mhs terurut berdasarkan field-field tersebut. Perhatikan hasil latihan anda ini pada dosen pembimbing.
9. Simpan kembali file-file project anda, dan silahkan lanjut ke materi praktikum berikutnya.

E. Mencari Data

Pencarian pada suatu table pada prinsipnya dapat dibagi atas;

1. Pencarian data pada field yang telah diindeks, dan
2. Pencarian data pada field yang belum diindeks.

Untuk pencarian data pada table yang telah diindeks, pada prinsipnya ada dua bentuk, yaitu;

1. Mencari data yang sama (*Find Key*), dan
2. Mencari data yang paling mirip (*Find Nearest*)

Berikut ini adalah langkah-langkah untuk membuat kode program pencarian data pada table yang telah diindeks, dan pencariannya dilakukan terhadap data yang sama (*Find Key*). Langkah-langkahnya adalah sebagai berikut;

1. Tampilkan form interface ke table mhs (*frmMHS*),
2. Tambahkan **SpeedButton** di sebelah kanan **Edit1**,
3. Ganti *property* name **SpeedButton1** menjadi **spbCari**, dan isi *property* captionnya dengan **Cari**
4. Agar ketika form ini tampil, *property* Text Edit1 telah terhapus, hapus nilai *property* **Text**-nya,
5. Klik ganda **spbCari**, dan ketik kode program di bawah ini;

```
Dm.tbMHS.FindKey([Edit1.Text]);  
Dm.tbMHS.GotoKey;
```

6. Simpan kembali file-file *project* anda, dan jalankan program ini.
7. Silahkan anda login dan setelah itu klik menu **File | Data Mahasiswa**. Ketika Tampilan data terurut berdasarkan *field nobp*, ketik salah satu nobp yang telah ada di table mhs kemudian klik tombol **Cari**. Apakah data (nobp) yang ada cari ketemu?.
8. Tampilkan data terurut berdasarkan field nama dengan cara mengklik dan memilih Nama melalui *ComboBox Berdasarkan:*, kemudian ketik salah satu nama yang telah ada di table **mhs** pada **Edit1**, kemudian klik tombol **Cari**. Apakah nama mahasiswa yang anda cari ketemu? Buat kesimpulan anda terhadap hasil pengujian yang telah anda lakukan pada langkah 7 dan 8 di atas di lembar **Jawaban dan Tugas** di bawah ini.

Latihan dan Tugas:

1. Ketika anda mencari nobp melalui langkah 7 di atas, jika anda mengetikkan **nobp** yang tidak ada di table mhs dan anda mengklik tombol **Cari**, maka *pointer* akan berada di record terakhir (program menampilkan data terakhir) tanpa menampilkan pesan apa pun. Jika program yang anda buat harus menampilkan pesan. "**Pencarian tidak berhasil...**" ketika proses pencarian tidak berhasil dan record data berada di record pertama, tuliskanlah kode programnya
2. Tambahkan fasilitas untuk memilih pilihan **Mencari Data yang Sama** dan **Mencari Data yang Paling Mirip** di program anda. Dan tambahkan kode programnya (kode program untuk melaksanakan kedua jenis pencarian di atas.

Catatan:

- Untuk menampilkan pilihan; **Mencari Data yang Sama** dan **Mencari Data yang Paling Mirip** dapat anda gunakan komponen *Radio Group*.
- Perintah untuk mencari data yang paling mirip adalah;

```
Dm.tbMHS.FindNearest([Edit1.Text]);  
Dm.tbMHS.GotoNearest;
```

Source Code praktikum ini dapat dilihat di situs:

<http://www.sufayasa.com/>

