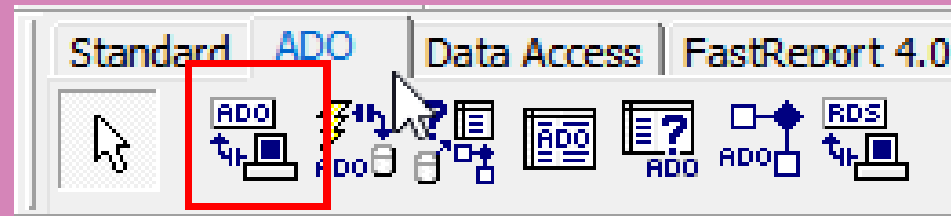


Koneksi ke Database

# Menghubungkan Microsoft Access

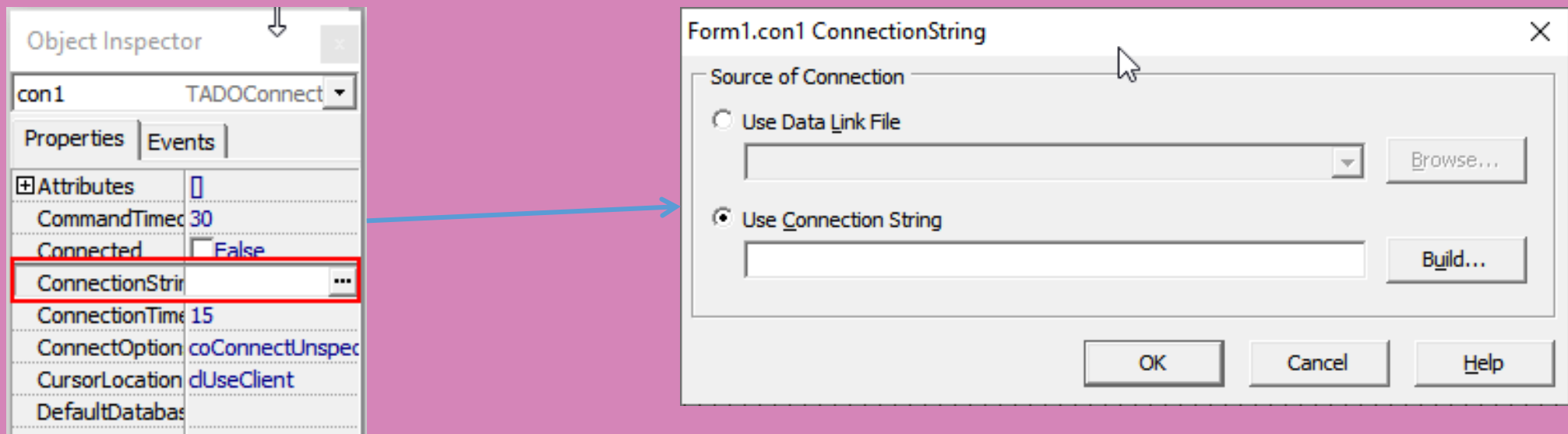
- Komponen yang diperlukan untuk menghubungkan Delphi 7 dengan Access adalah **TADOConnection**.
- Komponen ini ada pada **Toolbox** bagian **ADO**



AdoConnection

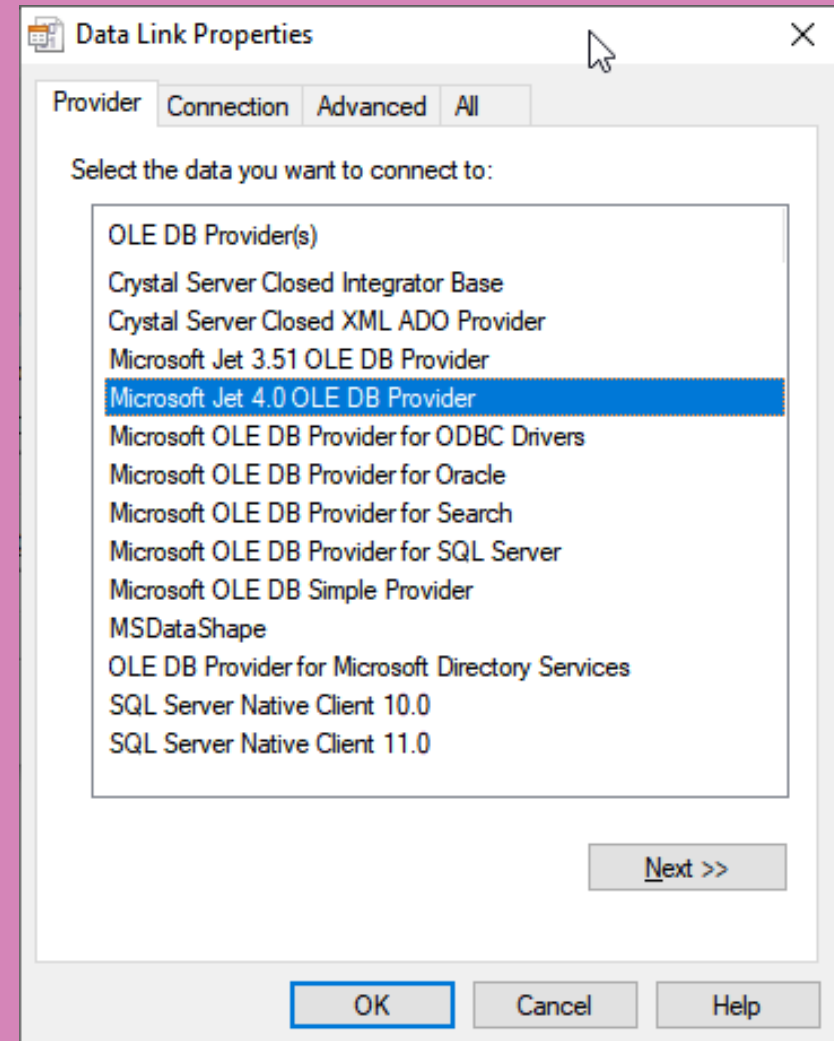
# TADOConnection

- Komponen yang digunakan untuk menghubungkan ke database Access
- Properties yang digunakan untuk terhubung ke database Access adalah **Connection String**
- Selain melalui **Properties**, Connection String juga bisa diakses dengan mengklik 2x pada komponen **TADOConnection**.



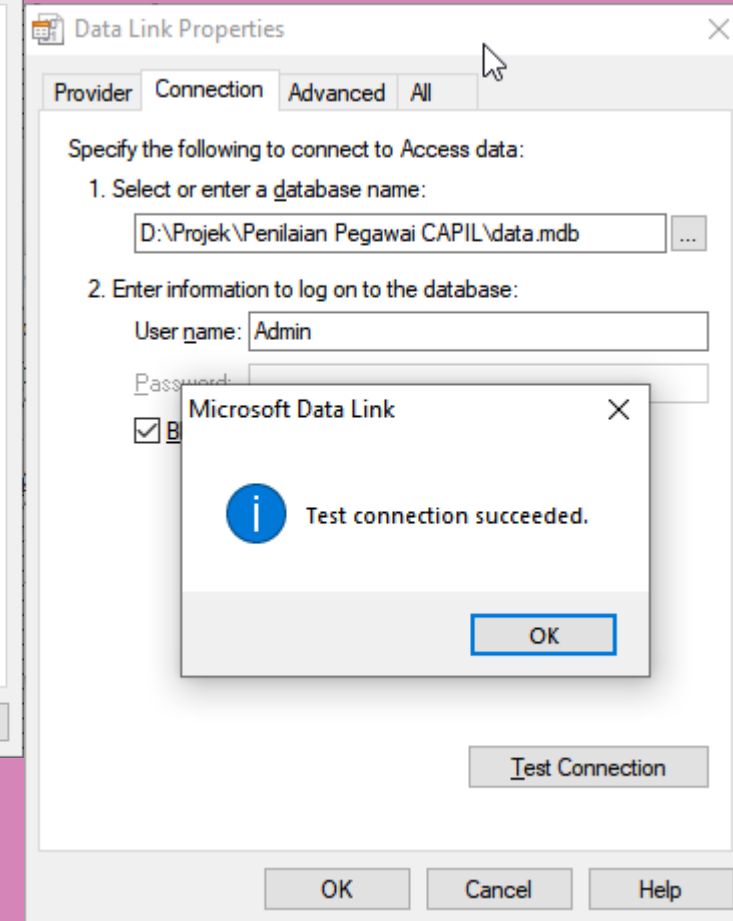
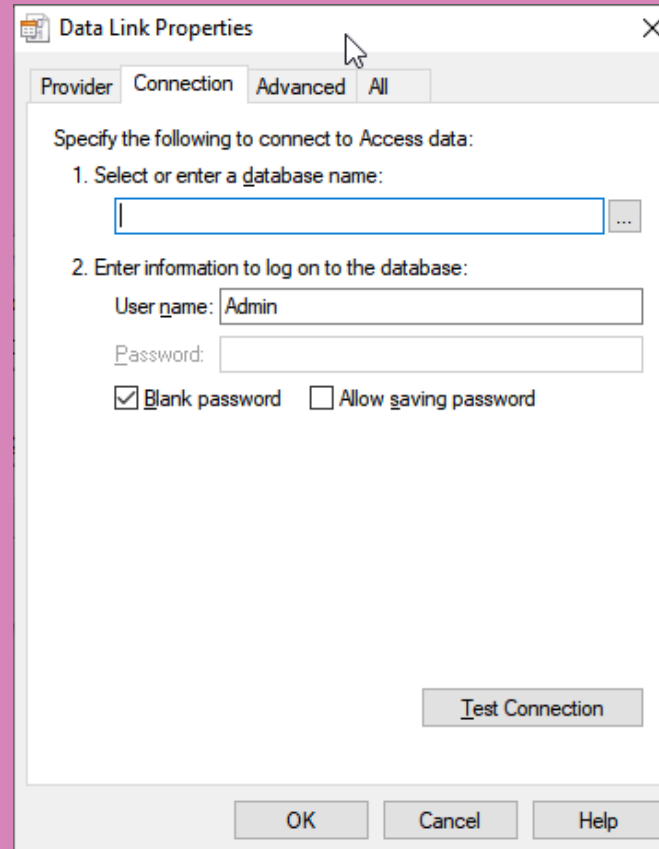
# Mengatur Connection String (1)

- Klik **Build** pada jendela **Connection String** yang aktif.  
Akan muncul jendela **Data Link Properties**.
- Ada banyak pilihan Providers Database yang bisa kita gunakan.  
Untuk provider database **Access**, gunakan provider **Microsoft Jet OLE DB Provider**.
- Kemudian **Next**



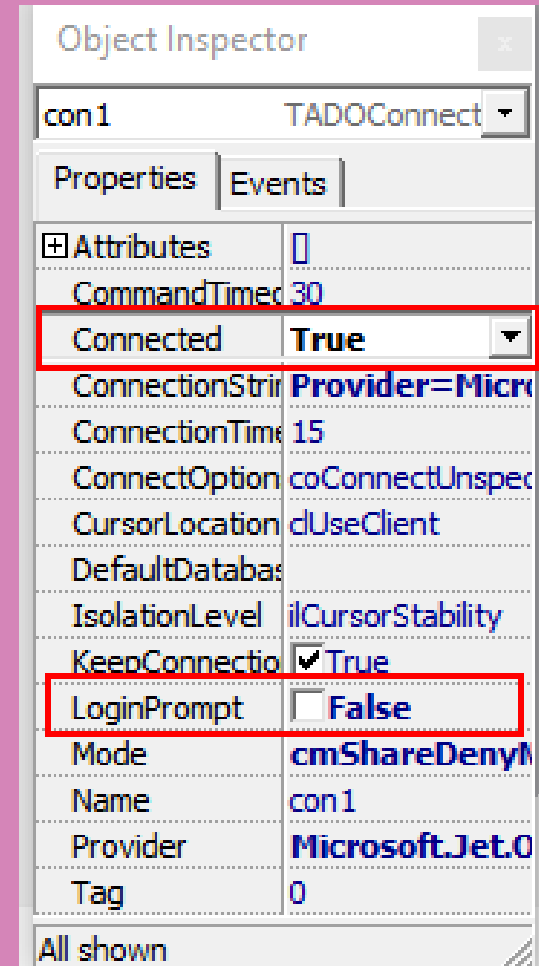
# Mengatur Connection String (2)

- Pada bagian **Connection**, pilih file Database Access yang sudah dibuat sebelumnya.
- Untuk mengecek apakah sudah terhubung dengan database yang dipilih, klik tombol **Test Connection**.
- Pastikan yang muncul adalah **Test Connection Succeeded**.



# Property Lanjutan TAdoConnection

- Property yang dilakukan setelah mengatur **Connection String** pada TAdoConnection adalah **Login Prompt** dan **Connected**.
- Atur Property **Login Prompt** menjadi **False** dan **Connected** menjadi **True**.
- Apabila setelah mengganti property Connected menjadi True terjadi Error, maka atur kembali Connection String

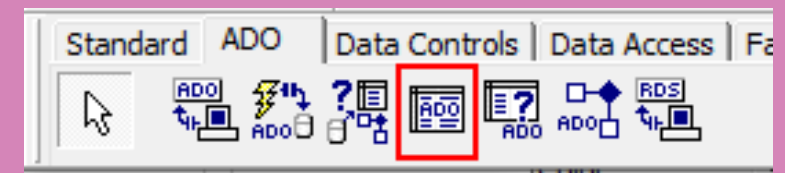


AdoTable



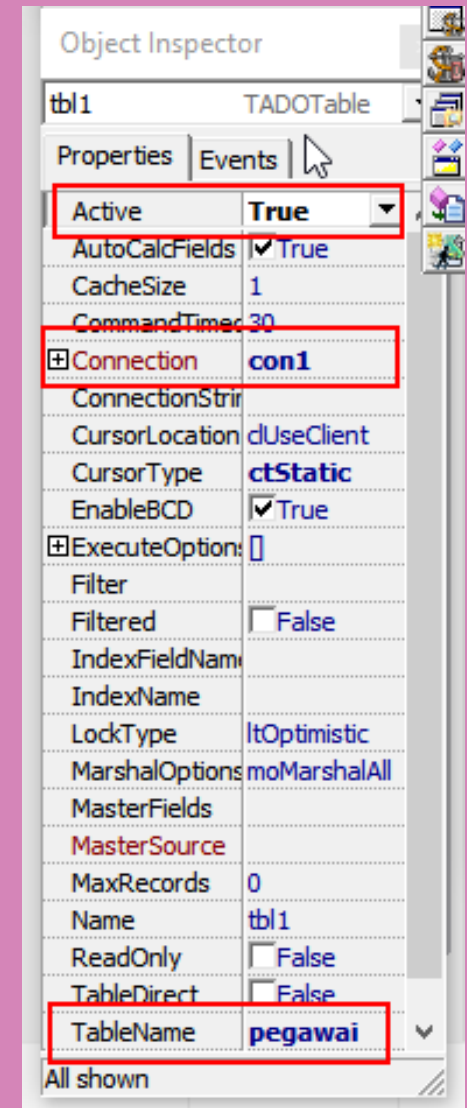
# TAdoTable

- Komponen berikutnya yang digunakan untuk keperluan terhubung ke database adalah TAdoTable.
- Digunakan untuk terhubung pada tabel dari database yang digunakan. Dapat digunakan juga untuk proses menambah, menghapus dan mengubah data pada tabel.
- Property yang perlu diperhatikan adalah: **Active, Connection, Table Name**.
- **Urutan** property nya adalah atur **Connection** terlebih dahulu, kemudian atur **Table Name**, baru atur property **Active**.



# Property TAdoTable

- Property **Connection** diisi dengan nama TAdoConnection
- Property **Table Name** dipilih tabel yang ingin digunakan.
- Property **Active** atur menjadi **True**.
- Untuk property name, disarankan namanya sesuai dari tabel yang dipilih di table name. Misal TableName nya pegawai, maka name dari AdoTable ini tbPegawai atau tbl\_pegawai.
- Ini bertujuan agar memudahkan memanggil komponen tersebut pada source code.



# TAdoTable - Fungsi Dalam Menelusuri Data

## Mengatur Kursor ke

- Awal record gunakan method **First**
- Akhir record gunakan method **Last**
- Maju 1 record gunakan method **Next**
- Mundur 1 record gunakan method **Prior**
- Memeriksa Awal Record gunakan property **BOF**
- Memeriksa Akhir Record gunakan property **EOF**

## Mencari Data:

- Locate
- FindFirst
- FindNext

# TAdoTable - Fungsi Dalam Manipulasi Data

- Insert : Untuk menambahkan record di kursor yang sedang aktif.
- Append: Untuk menambahkan record di akhir record.
- Edit: Untuk mengubah record yang aktif.
- Delete: Untuk menghapus record yang aktif.
- Post: Untuk menyimpan data ke database.
- Cancel: Untuk membatalkan pengubahan data.

AdoDataset

# AdoDataset

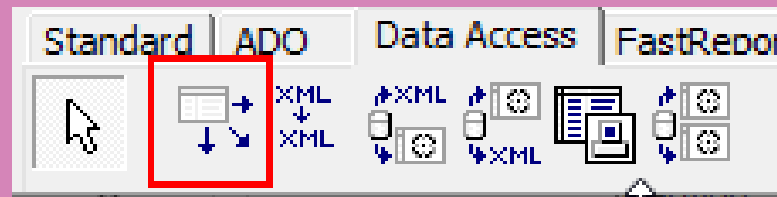
- Fungsi AdoDataset ini sama seperti AdoTable untuk mengambil data dari suatu tabel.
- Juga bisa digunakan untuk melakukan proses manipulasi data. Insert, Update, Delete, First, Last, Next, Prior
- Properties: **Active, Connection, CommandText**
- Urutan Propertiesnya: atur **Connection** lalu isi **CommandText** dan ubah **Active** menjadi **True**

# AdoDataset dan AdoTable

- AdoTable sangat mudah dalam menentukan tabel mana yang mau diambil. Hanya lewat properties **TableName**. Sedangkan AdoDataset mengambil data melalui **query sql** yang ada di properties **CommandText**
- AdoTable hanya bisa digunakan untuk 1 tabel saja, tidak bisa **dijoinkan** dengan tabel lain.
- AdoDataset bisa dijoiinkan dengan tabel lain, karena kita bisa tentukan pada **CommandText** (isi query sql Join)

# Data Source

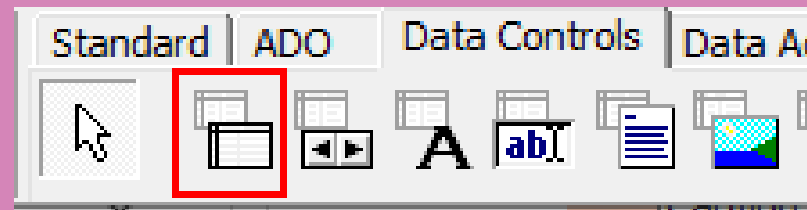
- Sebagai jembatan untuk mengirimkan data ke tabel Delphi
- **Property** yang perlu diperhatikan adalah: **Data Set**.
- **Data Set** bisa diatur dengan memilih AdoTable yang sudah dibuat sebelumnya.
- Komponen ini ada di bagian tab **Data Access**





# DBGrid

- Komponen ini berbentuk tabel.
- Komponen yang digunakan untuk menampilkan data dari database
- Property yang perlu diatur adalah **Data Source**. Yaitu dipilih dari Data Source yang sudah dibuat sebelumnya.
- Komponen ini ada di bagian tab **Data Control**



# Projek CRUD Mahasiswa

- Buat Folder dengan nama Projek Kampus
- Buat database di Access dengan nama: kampus\_uniska.mdb simpan di Folder Projek Kampus
- Buat tabel dengan nama: Mahasiswa
- Struktur Tabelnya:
  - id auto number primary key
  - npm Short Text (15)
  - nama\_lengkap Short Text (50)
  - prodi Short Text (50)
  - tanggal\_lahir date
  - tempat\_lahir Short Text(50)
  - jenkel Short Text(10)

# Projek CRUD Mahasiswa (2)

- Buat Projek Delphi dengan nama: Projek\_Kampus
- Buat Form (Unit1) dengan nama: unitMahasiswa
- Desain Form Mahasiswa seperti berikut:

# Form Mahasiswa

Form1

## FORM DATA MAHASISWA

Input Data

NPM

Nama Lengkap

Prodi ☐ Teknik Informatika ☐ Sistem Informasi

Tanggal Lahir

Tempat Lahir

Jenis Kelamin ☐ Pria ☐ Wanita

Cari data mahasiswa berdasarkan nama mahasiswa

1	

# Tambahkan Komponen Database

- Tambahkan 2 buah komponen dari tab ADO yaitu **AdoConnection** dan **AdoDataset**
- Tambahkan 1 komponen pada tab Data Access yaitu **Data Source**

# Property AdoConnetion

- Atur Properties Connection String agar bisa terhubung dengan database yang dibuat yaitu database: **kampus\_uniska.mdb**
- Pastikan berhasil konek ke database tersebut.
- Atur **LoginPrompt** menjadi **False**
- Atur **Connected** menjadi **True**
- Atur **Name** menjadi conDB

# Property ADODataSet

- Atur Properties pada Connection dengan memilih **conDB** (nama ADOConnection)
- Isi properties **CommandText** menjadi **Select \* from mahasiswa;**
- Atur properties **Active** menjadi **True**
- Kemudian **name** ubah jadi **dsMahasiswa**

# Property DataSource dan DBGrid

- Pada komponen data source, atur properties **DataSet** dengan memilih **dsMahasiswa** (nama dari AdoDataset)
- Pada komponen DbGrid atur properties **Data Source** dengan memilih **ds1** (nama dari DataSource).
- Setelah diubah properties dbGrid, maka dbGrid tersebut akan menampilkan data dari tabel mahasiswa.



# Event Form\_Create dan Button Bersih

```
6 procedure TForm1.FormCreate(Sender: TObject);  
7 begin  
8   edt1.Clear;  
9   edt2.Clear;  
0   edt3.Clear;  
1   edt4.Clear;  
2   btn1.Enabled := True; //button tambah aktif  
3   btn2.Enabled := False; //button ubah tidak aktif  
4   btn3.Enabled := False; //button hapus tidak aktif  
5 end;
```

Form\_Create

Button Bersih

```
118 procedure TForm1.btn4Click(Sender: TObject);  
119 begin  
120   edt1.Clear;  
121   edt2.Clear;  
122   edt3.Clear;  
123   edt4.Clear;  
124   btn1.Enabled := True; //button tambah aktif  
125   btn2.Enabled := False; //button ubah tidak aktif  
126   btn3.Enabled := False;  
127 end;
```

# Event Button Tambah

```
procedure TForm1.btn1Click(Sender: TObject);  
begin  
    if edt1.Text = '' then  
        ShowMessage('NPM belum diisi!')  
    else if edt2.Text = '' then  
        ShowMessage('Nama lengkap belum diisi')  
    else if edt3.Text = '' then  
        ShowMessage('Tempat lahir belum diisi')  
    else  
        begin  
            dsMahasiswa.Append;  
            dsMahasiswa.FieldValues['npm'] := edt1.Text;  
            dsMahasiswa.FieldValues['nama_lengkap'] := edt2.Text;  
            dsMahasiswa.FieldValues['prodi'] := rg1.Items[rg1.ItemIndex];  
            dsMahasiswa.FieldValues['tanggal_lahir'] := edtdtp1.Date;  
            dsMahasiswa.FieldValues['tempat_lahir'] := edt3.Text;  
            dsMahasiswa.FieldValues['jenkel'] := rg2.Items[rg2.ItemIndex];  
            dsMahasiswa.Post;  
            ShowMessage('Data mahasiswa berhasil ditambahkan!');  
            edt1.Clear;  
            edt2.Clear;  
            edt3.Clear;  
            edt4.Clear;  
            btn1.Enabled := True; //button tambah aktif  
            btn2.Enabled := False; //button ubah tidak aktif  
            btn3.Enabled := False; //button hapus tidak aktif  
        end  
    end;  
end;
```

# Event DBGrid Cell\_Click

```
95 procedure TForm1.db1CellClick(Column: TColumn);
96 var prodi, jenkel:string;
97 begin
98   edt1.Text:= dsMahasiswa.FieldValues['npm'];
99   edt2.Text:= dsMahasiswa.FieldValues['nama_lengkap'];
100   prodi := dsMahasiswa.FieldValues['prodi'];
101   if prodi = 'Teknik Informatika' then
102     rg1.ItemIndex := 0
103   else
104     rg1.ItemIndex := 1;
105   edtdtp1.date:= dsMahasiswa.FieldValues['tanggal_lahir'];
106   edt3.Text:= dsMahasiswa.FieldValues['tempat_lahir'];
107   jenkel := dsMahasiswa.FieldValues['jenkel'];
108   if jenkel = 'Pria' then
109     rg2.ItemIndex := 0
110   else
111     rg2.ItemIndex := 1;
112     btn1.Enabled := false;
113     btn2.Enabled := True;
114     btn3.Enabled := True;
115
116 end;
117
```

# Event Button Ubah

```
129 procedure TForm1.btn2Click(Sender: TObject);
130 begin
131     if edt1.Text = '' then
132         ShowMessage('NPM belum diisi!')
133     else if edt2.Text = '' then
134         ShowMessage('Nama lengkap belum diisi')
135     else if edt3.Text = '' then
136         ShowMessage('Tempat lahir belum diisi')
137     else
138         begin
139             dsMahasiswa.Edit;
140             dsMahasiswa.FieldValues['npm'] := edt1.Text;
141             dsMahasiswa.FieldValues['nama_lengkap'] := edt2.Text;
142             dsMahasiswa.FieldValues['prodi'] := rg1.Items[rg1.ItemIndex];
143             dsMahasiswa.FieldValues['tanggal_lahir'] := edtdtp1.Date;
144             dsMahasiswa.FieldValues['tempat_lahir'] := edt3.Text;
145             dsMahasiswa.FieldValues['jenkel'] := rg2.Items[rg2.ItemIndex];
146             dsMahasiswa.Post;
147             ShowMessage('Data mahasiswa berhasil diperbaharui!');
148             edt1.Clear;
149             edt2.Clear;
150             edt3.Clear;
151             edt4.Clear;
152             btn1.Enabled := True; //button tambah aktif
153             btn2.Enabled := False; //button ubah tidak aktif
154             btn3.Enabled := False; //button hapus tidak aktif
155         end;
156     end;
157
```

# Event Button Hapus

```
158 procedure TForm1.btn3Click(Sender: TObject);  
159 begin  
160   if (MessageDlg('Ingin hapus data ini?', mtConfirmation, [mbYes, mbNo], 0) = mrYes) then  
161   begin  
162     dsMahasiswa.Delete;  
163     ShowMessage('Data mahasiswa berhasil diperbaharui!');  
164     edt1.Clear;  
165     edt2.Clear;  
166     edt3.Clear;  
167     edt4.Clear;  
168     btn1.Enabled := True; //button tambah aktif  
169     btn2.Enabled := False; //button ubah tidak aktif  
170     btn3.Enabled := False; //button hapus tidak aktif  
171   end;  
172 end;  
173
```

# Event Edt Cari (edt4)

```
175 procedure TForm1.edt4Change(Sender: TObject);  
176 begin  
177     dsMahasiswa.Active:= false;  
178     dsMahasiswa.CommandText := 'select * from mahasiswa where nama_lengkap like "%" + edt4.Text + "%";  
179     dsMahasiswa.Active:= true;  
180 end;  
181
```

# Tugas

- Kerjakan 1 CRUD tambahan dengan ketentuan sebagai berikut:
- Nama Tabel: Dosen
  - Struktur tabel
    - Id autonumber
    - Nidn shorttext (10)
    - nama\_dosen short\_text(50)
    - tanggal\_masuk date
    - Alamat short\_text(50)
    - no\_wa short\_text(16)
    - Agama short\_text(15)
    - status (PNS/DTY) short\_text(3)