



**MODUL PERKULIAHAN PEMROGRAMAN II (BORLAND DELPHI 7)**  
**JURUSAN TEKNIK INFORMATIKA**  
**UNIVERSITAS KOMPUTER INDONESIA**

# **STUDI KASUS DATABASE**

## **(SISTEM INFORMASI PENJUALAN BARANG)**

**OLEH : ANDRI HERYANDI, S.T.**

# Spesifikasi Software

2

- Dapat mengelola data Barang, Pelanggan, Supplier, dan Transaksi Penjualan. Untuk setiap pengolahan data harus tersedia fasilitas Penambahan, Pengeditan, Penghapusan dan Pencarian data.
- Dapat melakukan operasi penjualan barang.
- Dapat mencetak Laporan Barang, Pelanggan, Supplier, dan Penjualan baik keseluruhan atau berdasarkan kriteria tertentu

# Persiapan Pengembangan Software

3

Siapkan sebuah folder untuk menyimpan semua source code dan file databasenya.

Contoh : Buat folder “Program Penjualan Barang” di folder D:\Data Mahasiswa\NIM\

# Struktur Database

4

Berdasarkan spesifikasi software, dapat disimpulkan bahwa sistem yang akan dibangun terdiri dari data berikut :

- Data Barang
- Data Pelanggan
- Data Supplier
- Data Penjualan

# Data Barang

5

## Spesifikasi untuk data Barang

- ❑ Kode barang harus dapat dimasukan dengan menggunakan pengkodean Barcode
- ❑ Data barang harus menampung gambar produk
- ❑ Harga dibedakan menjadi HargaBeli dan HargaJual
- ❑ Data barang harus dapat digunakan untuk membuat rencana penambahan barang ketika stok telah menipis. Hal ini bisa dilakukan dengan membuat field Stok dan StokMinimal.
- ❑ Dapat melakukan pencarian pada field Kode, Nama, Harga Jual dan Stok.

# Struktur Data Barang

6

Restructure Paradox 7 Table: Barang.DB

Field roster:

	Field Name	Type	Size	Key
1	Kode	A	13	*
2	Nama	A	50	
3	HargaBeli	\$		
4	HargaJual	\$		
5	Stok	I		
6	StokMinimal	I		
7	Gambar	G		

Enter a field name up to 25 characters long.

☐ Pack Table

Save Save As... Cancel Help

Table properties:

Validity Checks: [Define...]

☐ 1. Required Field

2. Minimum value: [ ]

3. Maximum value: [ ]

4. Default value: [ ]

5. Picture: [ ]

Assist...

## Keterangan :

- Field Kode menggunakan tipe AlphaNumeric (A) dengan panjang 13 Karakter, sesuai dengan aturan Barcode EAN (standar pengkodean barcode untuk produk retail).

Nama File : Barang.DB

Simpan di Folder D:\Data Mahasiswa\NIM\Program Penjualan Barang

Disusun Oleh : Andri Heryandi, S.T.

# Secondary Index Data Barang

7

Nama Index	Indexed Field	Unique	Maintained	Case Sensitive	Descending
IdxNama	Nama		√		
IdxHargaJual	HargaJual		√		
IdxStok	Stok		√		
Silahkan tambahkan Secondary Index lain yang sekiranya diperlukan					

Untuk menambahkan Secondary Index pada suatu tabel paradox, caranya adalah :

- Buka Database Desktop
- Klik Tools → Utilities → Restructure, cari file Barang.DB, klik Open
- Di Table Properties, pilih Secondary Index
- Klik tombol Define, untuk membuat Secondary Index.
- Pilih Field yang akan diindex dan pilih IndexOption-nya. Klik tombol OK.
- Isi IndexName-nya sesuai dengan Nama Index. Klik OK.
- Jika telah selesai, klik tombol Save untuk menyimpan hasil restruktur tabel.

# Data Supplier

8

## Spesifikasi untuk data Supplier

- Setiap supplier memiliki kode yang bersifat unik berupa angka yang terus bertambah (autoincrement)
- Data yang harus disimpan terdiri dari Nama Supplier, Alamat, Kota, No Telepon, Fax, dan EMail.



# Struktur Data Supplier

9

Restructure Paradox 7 Table: Supplier.db

Field roster:

	Field Name	Type	Size	Key
1	Kode	+		*
2	Nama	A	50	
3	Alamat	M	240	
4	Kota	A	20	
5	NoTelepon	A	20	
6	Fax	A	20	
7	EMail	A	30	

Enter a field name up to 25 characters long.

☐ Pack Table

Table properties:

Validity Checks

Define...

☐ 1. Required Field

2. Minimum value:

3. Maximum value:

4. Default value:

5. Picture:

Assist...

Save Save As... Cancel Help

Nama File : Supplier.DB

Simpan di Folder D:\Data Mahasiswa\NIM\Program Penjualan Barang

Disusun Oleh : Andri Heryandi, S.T.

# Secondary Index Data Supplier

10

Nama Index	Indexed Field	Unique	Maintained	Case Sensitive	Descending
IdxNama	Nama		√		
IdxKota	Kota		√		
IdxEMail	EMail		√		
Silahkan tambahkan Secondary Index lain yang sekiranya diperlukan					

# Data Pelanggan

11

## Spesifikasi untuk data pelanggan

- Setiap pelanggan memiliki kode yang bersifat unik berupa angka yang terus bertambah (autoincrement)
- Data yang harus disimpan terdiri dari Nama Pelanggan, Alamat, Kota, No Telepon, Fax, dan EMail.

# Struktur Data Pelanggan

12

Restructure Paradox 7 Table: Pelanggan.db

Field roster:

	Field Name	Type	Size	Key
1	Kode	+		*
2	Nama	A	50	
3	Alamat	M	240	
4	Kota	A	20	
5	NoTelepon	A	20	
6	Fax	A	20	
7	EMail	A	30	

Enter a field name up to 25 characters long.

☐ Pack Table

Table properties:

Validity Checks

Define...

☐ 1. Required Field

2. Minimum value:

3. Maximum value:

4. Default value:

5. Picture:

Assist...

Save Save As... Cancel Help

Nama File : Pelanggan.DB

Simpan di Folder D:\Data Mahasiswa\NIM\Program Penjualan Barang

Disusun Oleh : Andri Heryandi, S.T.

# Secondary Index Data Pelanggan

13

Nama Index	Indexed Field	Unique	Maintained	Case Sensitive	Descending
IdxNama	Nama		√		
IdxKota	Kota		√		
IdxEMail	EMail		√		
Silahkan tambahkan Secondary Index lain yang sekiranya diperlukan					

# Data Penjualan

14

## Spesifikasi untuk data Penjualan

- Sebuah faktur memiliki nomor faktur berupa angka yang terus bertambah (autoincrement)
- Sebuah faktur boleh terdiri dari lebih dari 1 item penjualan. Ini menandakan bahwa kita memiliki 2 tabel yaitu tabel penjualan (Faktur) dan tabel Isi Faktur (Detail Penjualan) yang mempunyai relasi One-To-Many (satu ke banyak)
- Data yang disimpan dalam faktur terdiri dari pelanggan, tanggal dan waktu transaksi, total pembayaran.
- Data yang disimpan dalam item faktur adalah data barang, harga, qty.

# Struktur Data Penjualan

15

Restructure Paradox 7 Table: Penjualan.db

Field roster:

	Field Name	Type	Size	Key
1	Nomor	+		*
2	KodePelanggan	I		
3	WaktuTransaksi	@		
4	Total	\$		

Field size is not allowed for this field type.

Table properties:

Validity Checks

Define...

☐ 1. Required Field

2. Minimum value:

3. Maximum value:

4. Default value:

5. Picture:

Assist...

☐ Pack Table

Save Save As... Cancel Help

Nama File : Penjualan.DB

Simpan di Folder D:\Data Mahasiswa\NIM\Program Penjualan Barang

Disusun Oleh : Andri Heryandi, S.T.

# Struktur Data Penjualan

16

Restructure Paradox 7 Table: DetailPenjualan.db

Field roster:

	Field Name	Type	Size	Key
1	NomorPenjualan	I		*
2	KodeBarang	A	13	*
3	Qty	I		
4	Harga	\$		

Field size is not allowed for this field type.

Table properties:

Validity Checks

Define...

☐ 1. Required Field

2. Minimum value:

3. Maximum value:

4. Default value:

5. Picture:

Assist...

☐ Pack Table

Save Save As... Cancel Help

## Keterangan :

- Pada field NomorPenjualan dan KodeBarang pada kolom Key keduanya diisi \*. Itu artinya primary key pada tabel ini merupakan kombinasi dari 2 field yaitu Nomor dan KodeBarang. Ini akan mengakibatkan tidak boleh ada NomorPenjualan dan KodeBarang yang sama keduanya.

Nama File : DetailPenjualan.DB

Simpan di Folder D:\Data Mahasiswa\NIM\Program Penjualan Barang

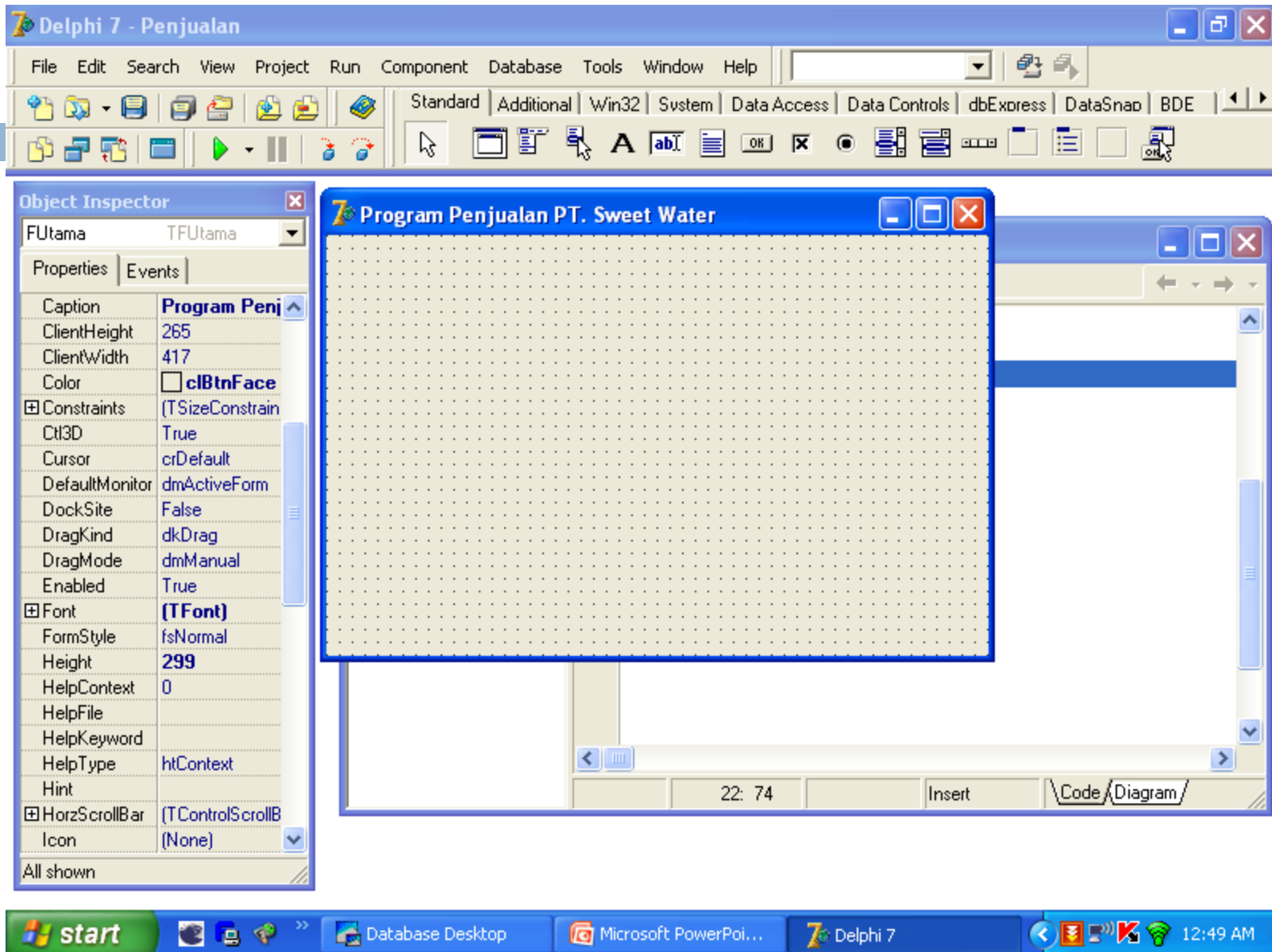
Disusun Oleh : Andri Heryandi, S.T.



# MEMULAI MENYUSUN PROGRAM

Disusun Oleh : Andri Heryandi, S.T.

- Buka Delphi
- Aktifkan Form1, ganti properti berikut :
  - ▣ Name : FUtama (Form Utama)
  - ▣ Caption : Program Penjualan PT. Sweet Water
- Simpan form utama dengan mengklik File → Save atau Save As. Simpan dengan nama file UFUtama.pas (Unit Form Utama) di folder yang telah dibuat.
- Simpan project dengan mengklik File → Save Project As. Simpan dengan nama file Penjualan.dpr.



# Membuat Data Module

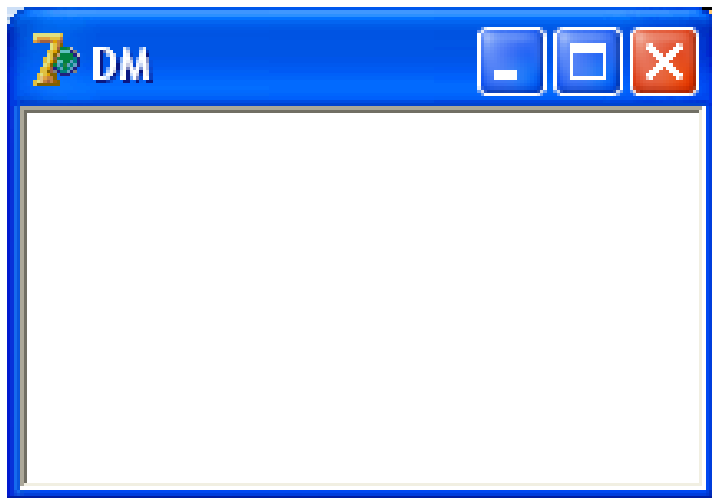
20

- Data Module digunakan untuk memusatkan komponen-komponen non visual (komponen yang ketika dirun tidak terlihat di form seperti komponen database, dialog, dan lain-lain)
- Langkah membuat Data Module, adalah :
  - ▣ Klik File → New → Data Module
  - ▣ Klik Data Module, ganti property Name dengan DM (agar lebih singkat)
  - ▣ Simpan data module dengan mengklik File → Save. Simpan dengan nama file UDM.pas (Unit Data Module)

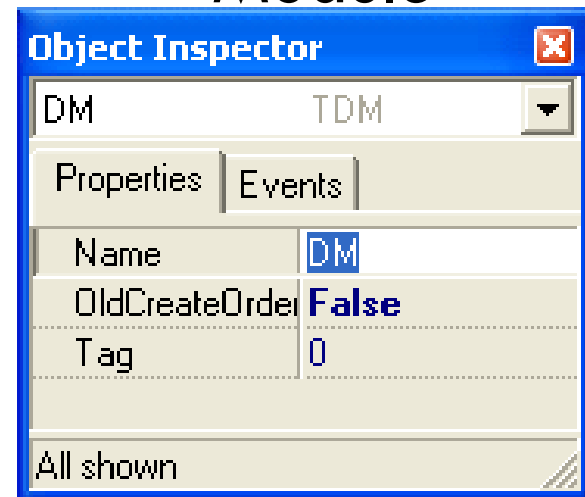
# Tampilan Data Module

21

## Data Module



## PropertyData Module



# Pengolahan Data Barang <sup>(1)</sup>

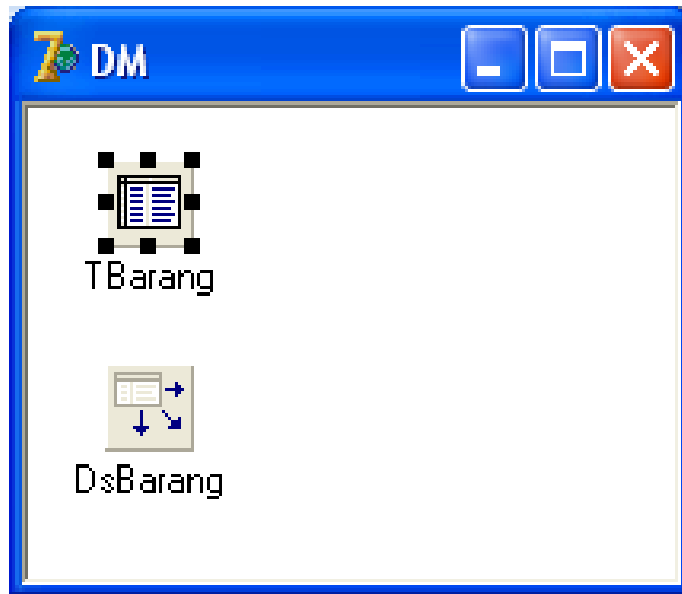
22

1. Tempatkan sebuah komponen TTable ke Data Module DM. Isi properti tersebut
  - ▣ DatabaseName : (Kosongkan saja/tanpa alias)
  - ▣ TableName : Barang.DB
  - ▣ Name : TBarang
  - ▣ Active : True
2. Tempatkan sebuah komponen TDataSource ke Data Module DM. Isi properti tersebut
  - ▣ DataSet : TBarang
  - ▣ Name : DsBarang
3. Untuk mempermudah dalam mengakses field, buatlah komponen-komponen fieldnya dengan cara :
  - ▣ Double klik di table TBarang
  - ▣ Klik kanan di Field Editor, klik Add All Fields

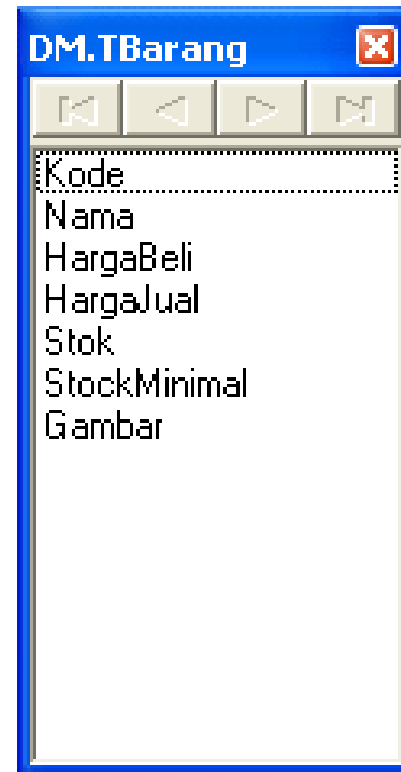
# Pengolahan Data Barang (2)

23

Data Module



FieldEditor  
TBarang



# Pengolahan Data Barang (3)

24

4. Buat sebuah form untuk mengolah data Barang.
  - ▣ Klik File → New → Form
  - ▣ Ganti properti dari form tersebut
    - Caption : Pengolahan Data Barang
    - Name : FBarang
  - ▣ Simpan form tersebut. Klik File → Save. Simpan dalam nama file UBarang.pas.
  - ▣ Karena form ini akan menggunakan data dari TBarang yang ada di data module, maka form ini harus dikenalkan dengan data modul DM dengan cara File → Use Unit. Pilih file UDM.pas, kemudian klik tombol OK.
5. Aturlah FBarang seperti pada halaman berikutnya



# Pengolahan Data Barang (4)

25

**Pengolahan Data Barang**

PENGOLAHAN DATA BARANG Panel1

Kode	Nama	HargaBeli
9780201571684	The Unified Modeling Language User Guide	\$150,000.00
978979333830X	Memahami Sistem Informasi	\$30,000.00
9789797311421	Pemrograman Berorientasi Objek Menggunakan Delphi	\$45,000.00
9789797633448	Pemrograman Database (Delphi + MSSQL Server)	\$40,000.00

DBGrid1

Data Diurutkan Berdasarkan :

- ☐ Kode RbKode
- ☐ Nama Barang RbNama
- ☐ Harga Jual RbHargaJual
- ☐ Stok RbStok

Pencarian Data :

EKeyword RbStok

Cari EKeyword TCari

Panel3

DBNavigator

TTambah

THapus

TEdit

TLaporan

TTutup

Disusun Oleh : Andri Heryandi, S.T.

Tambah Edit Hapus Laporan Tutup

Panel2

# Pengolahan Data Barang (5)

26

Urutan penempatan komponen di FBarang

1. Panel, dengan property :
  - Name : Panel1
  - Caption : PENGOLAHAN DATA BARANG
  - Align : alTop
2. Panel, dengan property :
  - Name : Panel2
  - Caption : (kosongkan)
  - Align : alBottom
3. Panel di bagian tengah form, dengan property :
  - Name : Panel3
  - Caption : (kosongkan)
  - Align : alRight
4. DBGrid, dengan property :
  - Name : DBGrid1
  - DataSource : DM.DsBarang
  - Align : alClient
5. DBNavigator di Panel2, dengan property :
  - Name : DBNavigator1
  - DataSource : DM.DsBarang
  - VisibleButton : nbFirst, nbPrior, nbNext, nbLast
6. BitBtn di Panel2, dengan property :
  - Name : TTambah
  - Caption : Tambah
7. BitBtn di Panel2, dengan property :
  - Name : TEdit
  - Caption : Edit
8. BitBtn di Panel2, dengan property :
  - Name : THapus
  - Caption : Hapus
9. BitBtn di Panel2, dengan property :
  - Name : TLaporan
  - Caption : Laporan
10. BitBtn di Panel2, dengan property :
  - Name : TTutup
  - Caption : Tutup

Disusun Oleh : Andri Heryandi, S.T.

# Pengolahan Data Barang (6)

27

11. RadioButton di Panel3, dengan property :

- Name : RbKode
- Caption : Kode
- Checked : True

12. RadioButton di Panel3, dengan property :

- Name : RbNama
- Caption : Nama Barang

13. RadioButton di Panel3, dengan property :

- Name : RbHargaJual
- Caption : Harga Jual

14. RadioButton di Panel3, dengan property :

- Name : RbStok
- Caption : Stok

15. Edit di Panel3, dengan property :

- Name : EKeyword
- Caption : (kosongkan)

16. BitBtn di Panel3, dengan property :

- Name : TCari
- Caption : Cari

# Pengolahan Data Barang (7)

28

## 6. Isi event OnClick tombol Tutup dengan :

```
procedure TFBarang.TTutupClick(Sender: TObject);  
begin  
    Close;  
end;
```

# Pengolahan Data Barang (8)

29

## 7. Isi event OnClick tombol RbKode dengan :

```
procedure TFBarang.RbKodeClick(Sender: TObject);  
begin  
    DM.TBarang.IndexName:=''; // Index aktif adalah Primary Index  
end;
```

# Pengolahan Data Barang (9)

30

## 8. Isi event OnClick tombol RbNama dengan :

```
procedure TFBarang.RbNamaClick(Sender: TObject);  
begin  
    DM.TBarang.IndexName:='IdxNama'; // Index aktif adalah IdxName  
end;
```

# Pengolahan Data Barang (10)

31

## 9. Isi event OnClick tombol RbHargaJual dengan :

```
procedure TFBarang.RbHargaJualClick(Sender: TObject);  
begin  
    DM.TBarang.IndexName:='IdxHargaJual'; // Index aktif adalah IdxHargaJual  
end;
```

# Pengolahan Data Barang (11)

32

## 10. Isi event OnClick tombol RbStok dengan :

```
procedure TFBarang.RbStokClick(Sender: TObject);  
begin  
    DM.TBarang.IndexName:='IdxStok'; // Index aktif adalah IdxStok  
end;
```



# Pengolahan Data Barang (12)

33

## 11. Isi event OnClick tombol TCari dengan :

```
procedure TFBarang.TCariClick(Sender: TObject);  
var  
    DataDitemukan:Boolean;  
begin  
    // Cari berdasarkan key sesuai dengan isi Text yang ada di EKeyword  
    DataDitemukan:=DM.TBarang.FindKey([EKeyword.Text]);  
    // ganti FindKey dengan FindNearest jika ingin pencarian  
    // menggunakan metode pencarian data terdekat (boleh sebagian)  
    if DataDitemukan=false then  
        Showmessage('Data tidak ditemukan!');  
end;
```

# Pengolahan Data Barang (13)

34

## 12. Isi event OnClick tombol TLaporan dengan :

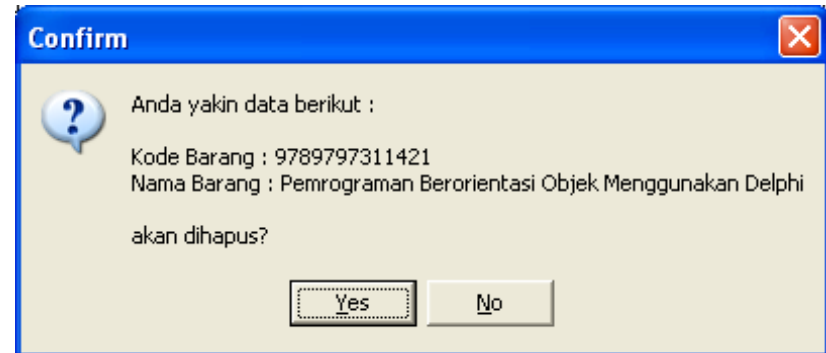
```
procedure TFBarang.TLaporanClick(Sender: TObject);  
begin  
    MessageDlg('Under Construction!', mtInformation, [mbOK], 0);  
end;
```

# Pengolahan Data Barang (14)

35

## 13. Isi event OnClick tombol THapus dengan :

```
procedure TFBarang.THapusClick(Sender: TObject);  
var  
    Pesan:String;  
begin  
    Pesan:='Anda yakin data berikut : '+#13+#13+ // #13 = Enter (pindah baris)  
        'Kode Barang : '+DM.TBarangKode.value+#13+  
        'Nama Barang : '+DM.TBarangNama.value+#13+#13+  
        'akan dihapus?';  
    if MessageDlg(Pesan,mtConfirmation,[mbYes,mbNo],0)=mrYes then  
    begin  
        DM.TBarang.Delete;  
    end;  
end;
```



Disusun Oleh : Andri Heryandi, S.T.

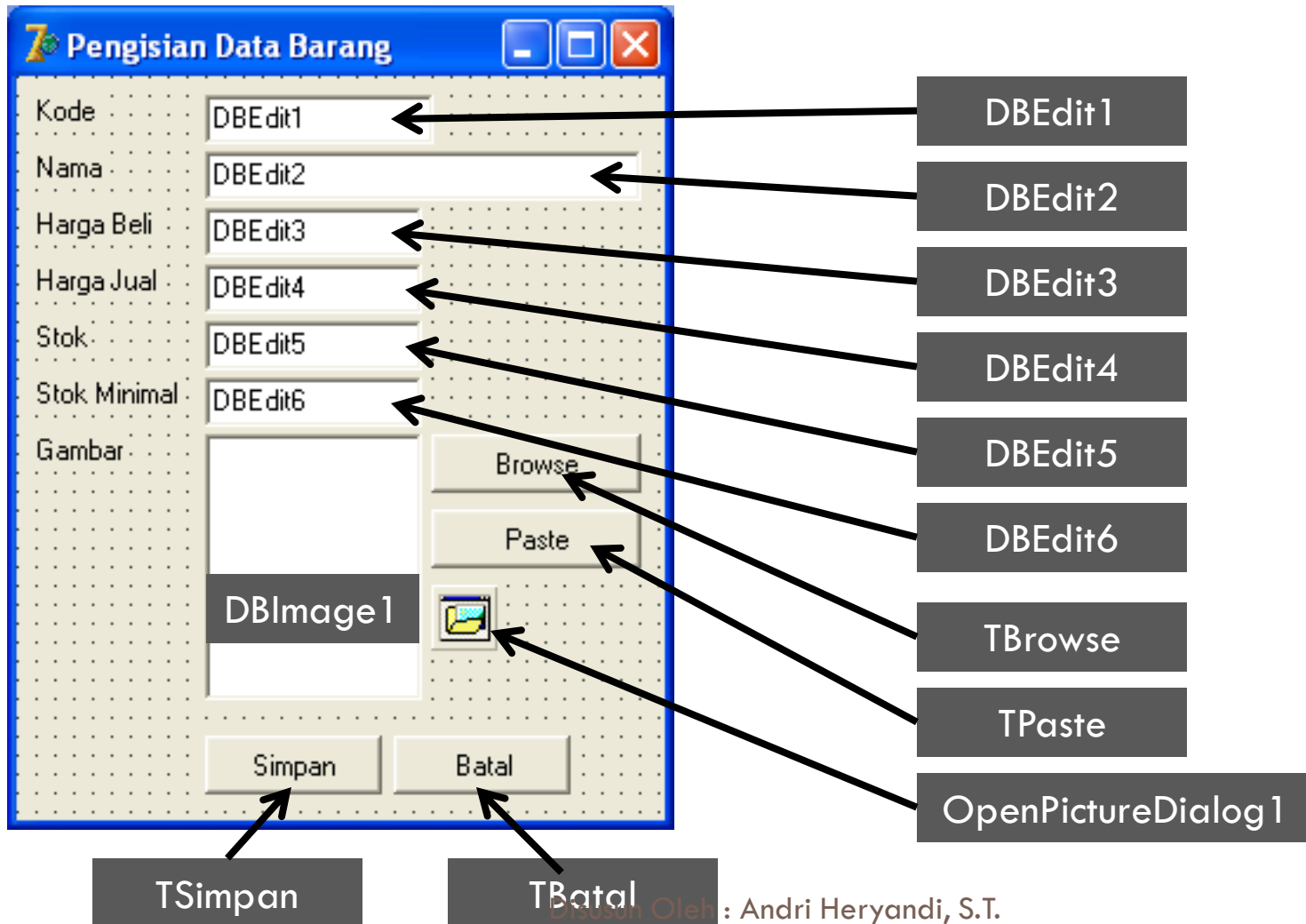
# Pengolahan Data Barang (15)

36

14. Buatlah sebuah form yang akan digunakan untuk pengisian data barang, baik untuk penambahan data atau pengeditan data. Langkah yang dilakukan adalah :
  - ▣ Buat form baru. Klik File → New → Form
  - ▣ Atur property form tersebut dengan :
    - Name : FlsiBarang
    - Caption : Pengisian Data Barang
    - BorderStyle : bsDialog
  - ▣ Save dengan nama UFlsiBarang.pas
15. Atur tampilan seperti di slide berikutnya.

# Pengolahan Data Barang <sup>(16)</sup>

37



Dibuat Oleh : Andri Heryandi, S.T.

# Pengolahan Data Barang (17)

38

1. DBEdit1, atur property :
  - ▣ DataSource : DM.DsBarang
  - ▣ DataField : Kode
2. DBEdit2, atur property :
  - ▣ DataSource : DM.DsBarang
  - ▣ DataField : Nama
3. DBEdit3, atur property :
  - ▣ DataSource : DM.DsBarang
  - ▣ DataField : HargaBeli
4. DBEdit4, atur property :
  - ▣ DataSource : DM.DsBarang
  - ▣ DataField : HargaJual
5. DBEdit5, atur property :
  - ▣ DataSource : DM.DsBarang
  - ▣ DataField : Stok
6. DBEdit6, atur property :
  - ▣ DataSource : DM.DsBarang
  - ▣ DataField : StokMinimal
7. DBImage1, atur property :
  - ▣ DataSource : DM.DsBarang
  - ▣ DataField : Gambar
8. OpenPictureDialog1. Ada di tab komponen Dialog.
9. TSimpan (TBitBtn), atur property :
  - ▣ Caption : Simpan
10. TBatal (TBitBtn), atur property :
  - ▣ Caption : Batal
11. TBrowse (TBitBtn), atur property :
  - ▣ Caption : Browse
12. TPaste (TBitBtn), atur property :
  - ▣ Caption : Paste

Jika ketika mengisi DataSource ternyata pilihannya kosong, maka UDM harus dikenalkan ke FIsiBarang dengan cara File → Use Unit → Pilih UDM.

# Pengolahan Data Barang (18)

39

## 16. Isi event OnClick tombol TBrowse dengan :

```
procedure TFIsiBarang.TBrowseClick(Sender: TObject);
begin
    if OpenPictureDialog1.Execute then
    begin
        // Isi field Gambar di Load dari File yang dipilih di OpenPictureDialog1
        DM.TBarangGambar.LoadFromFile(OpenPictureDialog1.FileName);
    end;
end;
```

# Pengolahan Data Barang <sup>(19)</sup>

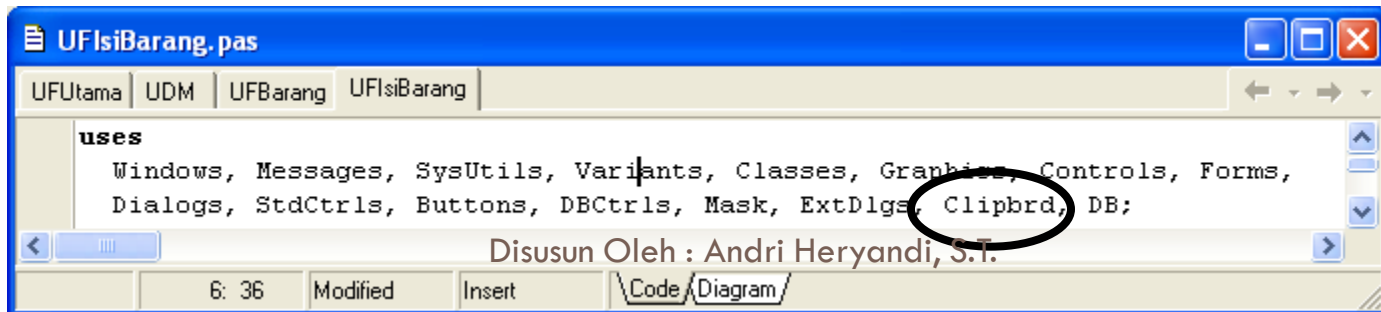
40

## 17. Isi event OnClick tombol TPaste dengan :

```
procedure TFIsiBarang.TPasteClick(Sender: TObject);
begin
    if Clipboard.HasFormat(CF_PICTURE) then // Jika clipboard berisi gambar
    begin
        DBImage1.PasteFromClipboard; // Simpan isi clipboard ke DBImage1
    end
    else
        ShowMessage('Data di clipboard bukan gambar.');
```

end;

Karena dalam prosedur ini menggunakan fungsi clipboard, maka agar perintah ini dikenal, di pernyataan uses (ada di bagian atas unit) harus ditambah dengan nama unit ClipBrd





# Pengolahan Data Barang (20)

41

## 18. Isi event OnClick tombol TSimpan dengan :

```
procedure TFIsiBarang.TSimpanClick(Sender: TObject);
begin
    // Sebelum disimpan, silahkan di validasi dulu disini
    // Contoh : -----
    // DM.TBarangKode.Value:=Trim(DM.TBarangKode.Value); // Hapus spasi yang tidak perlu
    // if DM.TBarangKode.IsNull then
    // begin
    //     ShowMessage('Kode barang tidak boleh kosong');
    //     DBEdit1.SetFocus;
    //     Exit;
    // end;
    // ----- Akhir dari validasi
    DM.TBarang.Post; // Simpan data
    Close;// Tutup form FIsiBarang (Kembali ke FBarang)
end;
```

# Pengolahan Data Barang (21)

42

## 18. Isi event OnClick tombol TBatal dengan :

```
procedure TFIsiBarang.TBatalClick(Sender: TObject);
begin
    if MessageDlg('Pengisian data dibatalkan ?',mtConfirmation,
        [mbYes,mbNo],0)=mrYes then
    begin
        DM.TBarang.Cancel;
        Close;// Tutup form FIsiBarang (Kembali ke FBarang)
    end;
end;
```

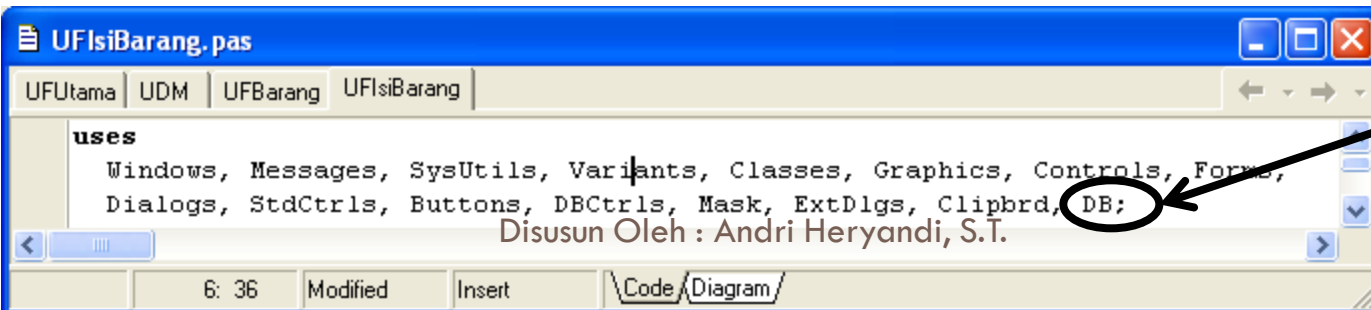
# Pengolahan Data Barang (22)

43

## TIPS dan TRIK

Pada FlsiBarang kita menginginkan bahwa user hanya akan keluar kalau menekan tombol Simpan atau Batal. FlsiBarang yang tadi dibuat masih mempunyai kelemahan yaitu form dapat ditutup dengan mengklik tombol Close (X) atau Alt+F4. Untuk menangani masalah di atas, klik FlsiBarang, isi event OnCloseQuery dengan perintah berikut :

```
procedure TFIsiBarang.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
    if DM.TBarang.State in [dsInsert, dsEdit] then // Jika Status masih kondisi Tambah / Edit
    begin
        CanClose:=False; // Layar tidak boleh ditutup
        ShowMessage('Gunakan tombol Simpan atau Batal untuk menutup window ini');
    end;
end;
```



Tambahkan unit DB di Uses karena ada penggunaan dsInsert & dsEdit

# Pengolahan Data Barang (23)

44

19. Kembali ke FBarang (gunakan View → Form /Shift+F12) untuk berganti unit/form. Isi event **OnClick** tombol **TTambah** dengan :

```
procedure TBarang.TTambahClick(Sender: TObject);  
begin  
    DM.TBarang.Insert; // atau DM.TBarang.Append;  
    FIsiBarang.ShowModal; // Tampilkan Form Pengisian Data Barang  
end;
```

# Pengolahan Data Barang (24)

45

20. Masih di FBarang, isi event OnClick tombol TEdit dengan :

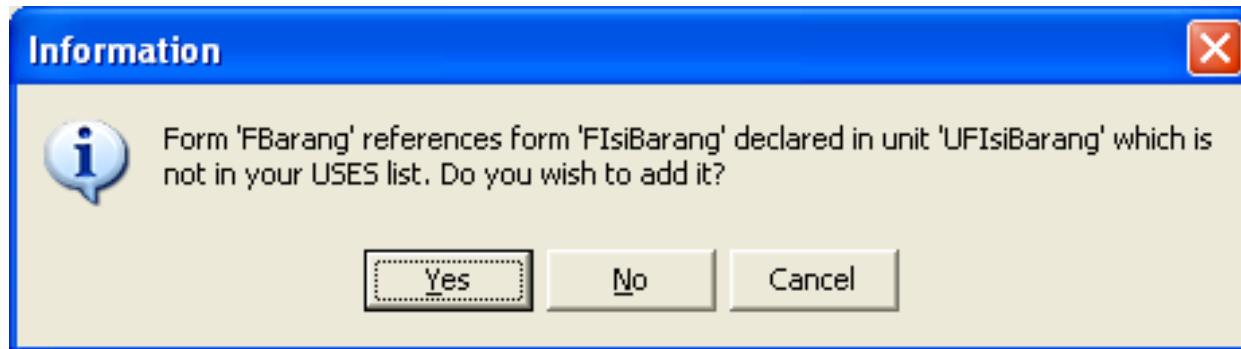
```
procedure TFBarang.TEditClick(Sender: TObject);  
begin  
    DM.TBarang.Edit; // atau DM.TBarang.Append;  
    FIsiBarang.ShowModal; // Tampilkan Form Pengisian Data Barang  
end;
```

# Pengolahan Data Barang (25)

46

Peringatan :

Ketika program ini dirun, ada kemungkinan muncul layar seperti di bawah ini.



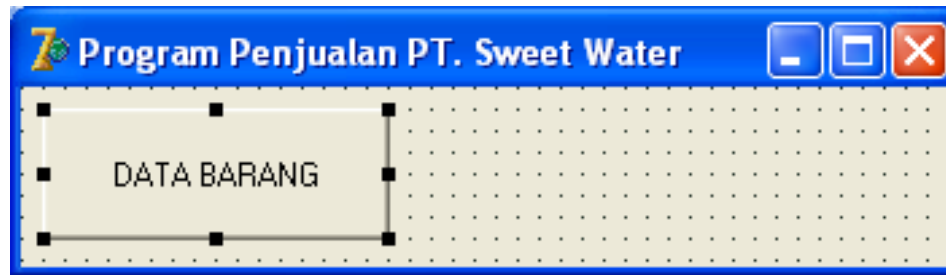
Dialog ini menyatakan bahwa Form FBarang memerlukan FisiBarang yang ada di unit UFIsiBarang. Apakah anda ingin menambahkan Uses UFIsiBarang di form FBarang. Klik Yes.

# Menghubungkan FBarang ke FUtama

47

Menghubungkan FBarang ke FUtama berguna agar FBarang dapat dipanggil oleh FUtama. Caranya :

- Buka FUtama (gunakan Shift+F12, pilih FUtama)
- File → Use Unit → Pilih UFBarang → Klik OK
- Buat sebuah tombol dengan caption “DATA BARANG”. Isi Name dengan TDataBarang



Isi event **OnClick** tombol **TDataBarang** dengan perintah di bawah ini :

```
procedure TFUtama.TDataBarangClick(Sender: TObject);  
begin  
    FBarang.ShowModal; // Tampilkan Form Pengolahan Data Barang  
end;
```



# Pengolahan Data Supplier

49

Langkah-langkah untuk pengolahan Data Supplier mirip dengan pengolahan data Barang. Hanya saja semua yang mengakses TBarang diganti dengan TSupplier

# Pengolahan Data Supplier

50

- Tempatkan Tabel ke DataModule untuk mengakses data Supplier



# Pengolahan Data Supplier

51

- Buat Form seperti di bawah ini

Pengolahan Data Supplier

PENGOLAHAN DATA SUPPLIER

Kode	Nama	Alamat
		(Memo)

Data Diurutkan Berdasarkan :  
☒ Kode  
☐ Nama Supplier  
☐ Kota  
☐ E-Mail

Pencarian Data :  
  
Cari

Tambah Edit Hapus Laporan Tutup

**Form FSupplier (UFSupplier.pas)**

# Pengolahan Data Supplier

52

- Buatlah form untuk pengolahan data Supplier

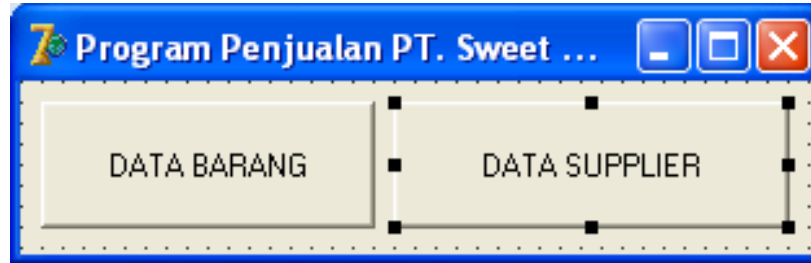
The screenshot shows a Windows-style form titled "Pengisian Data Supplier". It has a blue title bar with standard window controls. The form's background is a light gray grid. It contains the following controls:

- Kode:** A single-line text box labeled "DBEdit1".
- Nama:** A single-line text box labeled "DBEdit2".
- Alamat:** A multi-line memo area labeled "DBMemo1".
- Kota:** A single-line text box labeled "DBEdit3".
- NoTelepon:** A single-line text box labeled "DBEdit4".
- Fax:** A single-line text box labeled "DBEdit5".
- Mail:** A single-line text box labeled "DBEdit6".
- Buttons:** Two buttons at the bottom, "Simpan" (Save) and "Batal" (Cancel).

**Form FlsiSupplier (UFlsiSupplier.pas)**

# Pengolahan Data Supplier

53



Isi event **OnClick** tombol **TDataBarang** dengan perintah di bawah ini :

```
procedure TFUtama.TDataSupplierClick(Sender: TObject);  
begin  
    FSupplier.ShowModal; // Tampilkan Form Pengolahan Data Supplier  
end;
```

# Pengolahan Data Pelanggan

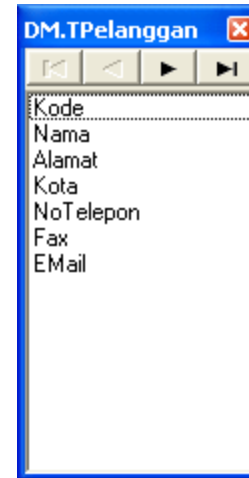
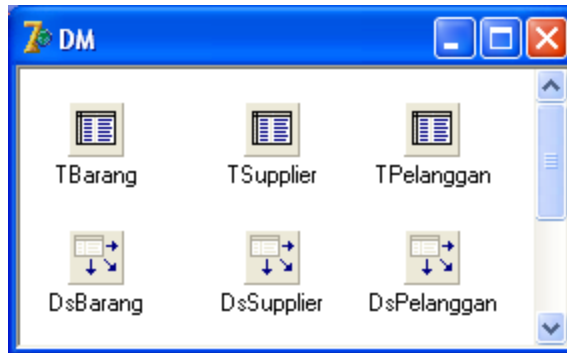
54

Langkah-langkah untuk pengolahan Data Pelanggan mirip dengan pengolahan data Barang. Hanya saja semua yang mengakses TBarang diganti dengan TPelanggan

# Pengolahan Data Pelanggan

55

- Tempatkan Tabel ke DataModule untuk mengakses data Pelanggan



# Pengolahan Data Pelanggan

56

- Buat Form seperti di bawah ini

Pengolahan Data Pelanggan

PENGOLAHAN DATA PELANGGAN

Kode	Nama	Alamat	Kota
1	Andri Heryandi	(MEMO)	Bandung

Data Diurutkan Berdasarkan :

- ☒ Kode
- ☐ Nama Supplier
- ☐ Kota
- ☐ E-Mail

Pencarian Data :

Cari

Tambah Edit Hapus Laporan Tutup

**Form FPelanggan (UFPelanggan.pas)**

Disusun Oleh : Andri Heryandi, S.T.



# Pengolahan Data Pelanggan

57

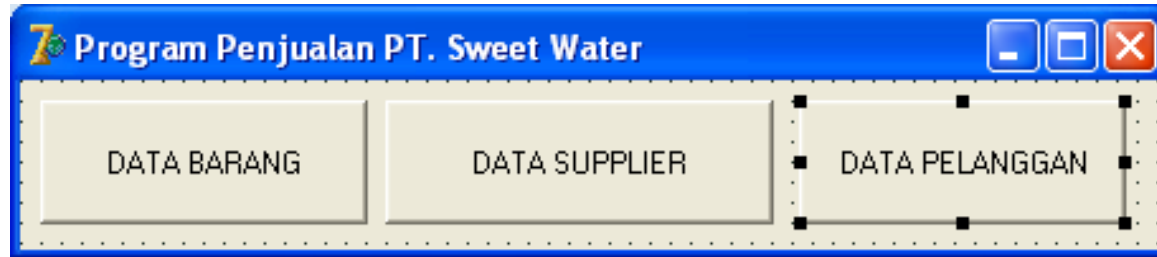
- Buat Form untuk pengisian data Pelanggan

The image shows a Windows application window titled "Pengisian Data Pelanggan". The window has a blue title bar with standard minimize, maximize, and close buttons. The main area has a light gray background with a dotted pattern. It contains several input fields: "Kode" (DBEdit1), "Nama" (DBEdit2), "Alamat" (DBMemo1), "Kota" (DBEdit3), "NoTelepon" (DBEdit4), "Fax" (DBEdit5), and "EMail" (DBEdit6). At the bottom, there are two buttons: "Simpan" and "Batal".

**Form FlsiPelanggan  
(UFlsiPelanggan.pas)**

# Pengolahan Data Pelanggan

58



Isi event **OnClick** tombol **TDataBarang** dengan perintah di bawah ini :

```
procedure TFUtama.TDataPelangganClick(Sender: TObject);  
begin  
    FPelanggan.ShowModal; // Tampilkan Form Pengolahan Data Pelanggan  
end;
```

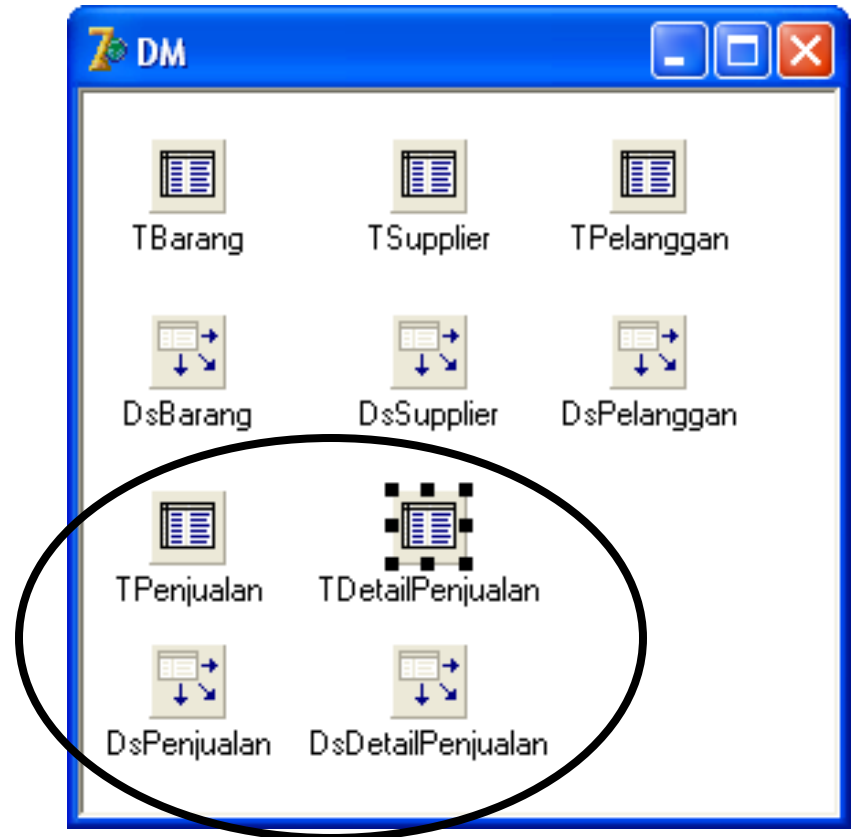
# Pengolahan Data Penjualan

59

# Pengolahan Data Penjualan

60

- Tempatkan 2 buah Table dan 2 buah Datasource ke Data Modul, Atur propertinya seperti di bawah ini :
  - Tabel pertama diisi dengan
    - Name : TPenjualan
    - TableName : Penjualan.DB
    - Active : True
  - Tabel kedua diisi dengan
    - Name : TDetailPenjualan
    - TableName : DetailPenjualan.DB
    - Active : True
  - Datasource pertama diisi dengan
    - Name : DsPenjualan
    - Dataset : TPenjualan
  - Datasource kedua diisi dengan
    - Name : DsDetailPenjualan
    - Dataset : TDetailPenjualan



# Pengolahan Data Penjualan

61

- Tambahkan komponen Field untuk TPenjualan dan TDetailPenjualan

The image displays two database table structure windows side-by-side. The left window, titled 'DM.TPenjualan', lists the fields: Nomor, KodePelanggan, WaktuTransaksi, and Total. The right window, titled 'DM.TDetailPenjualan', lists the fields: NomorPenjualan, KodeBarang, Qty, and Harga. Both windows have a standard database interface with a title bar, a close button, and a set of navigation buttons (back, forward, etc.) above the field list.

Table Name	Fields
DM.TPenjualan	Nomor, KodePelanggan, WaktuTransaksi, Total
DM.TDetailPenjualan	NomorPenjualan, KodeBarang, Qty, Harga

# Pengolahan Data Penjualan

62

- Buat form baru untuk pengolahan data penjualan
  - ▣ File → New → Form
  - ▣ Ganti properti Name : FPenjualan
  - ▣ Caption : Pengolahan Data Penjualan
  - ▣ File → Save dengan nama UFPenjualan.pas
- Susun form seperti pada gambar di slide berikutnya

# Pengolahan Data Penjualan

63

**Pengolahan Data Penjualan**

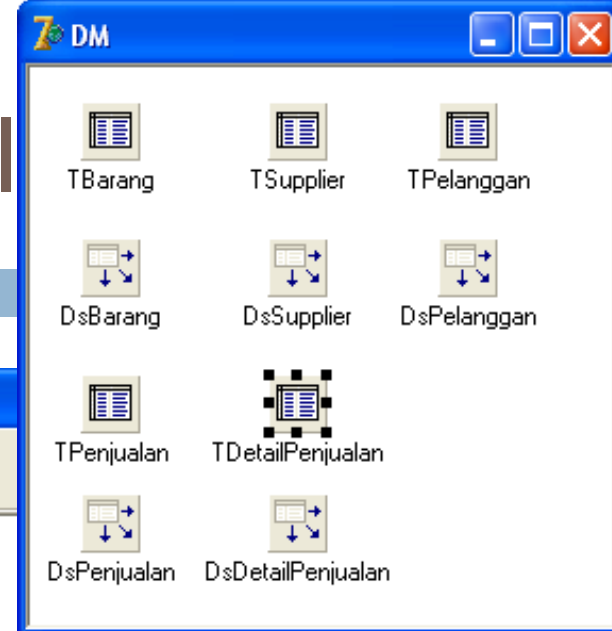
**PENGOLAHAN DATA PENJUALAN**

Nomor	KodePelanggan	WaktuTransaksi	Total
1	1	5/20/2008 8:00:00 AM	\$77,500.00
2	2	5/22/2008 2:11:50 AM	\$0.00
3	2	5/22/2008 2:12:58 AM	\$0.00
4	1	5/22/2008 2:14:43 AM	1,018,115.00
5	2	5/22/2008 8:12:58 AM	1,018,115.00

PERINCIAN FAKTUR PENJUALAN

NomorPenjualan	KodeBarang	Qty	Harga
1	4970129727514	1	\$7,500.00
1	4970129727538	10	\$7,500.00

Penjualan Baru    Lihat Faktur    Tutup



# Pengolahan Data Penjualan

64

## Urutan penempatan komponen di FBarang

1. Panel, dengan property :
  - Caption : PENGOLAHAN DATA PENJUALAN
  - Align : alTop
2. Panel, dengan property :
  - Caption : (kosongkan)
  - Align : alBottom
3. Panel di bagian tengah form, dengan property :
  - Name : Panel3
  - Caption : (kosongkan)
  - Align : alClient
4. DBGrid di Panel3, dengan property :
  - Name : DBGrid1
  - DataSource : DM.DsPenjualan
  - Align : alTop
5. Splitter (ada di tab Additional) di Panel3, dengan property :
  - Align : alTop
6. DBGrid di Panel3, dengan property :
  - Name : DBGrid2
  - DataSource : DM.DsDetailPenjualan
  - Align : alClient
7. Panel di Panel3, dengan property :
  - Align : alTop
  - Caption : Perincian Faktur Penjualan
8. DBNavigator di Panel2, dengan property :
  - Name : DBNavigator1
  - DataSource : DM.DsPenjualan
  - VisibleButton : nbFirst, nbPrior, nbNext, nbLast
9. BitBtn di Panel2, dengan property :
  - Name : TPenjualanBaru
  - Caption : Penjualan Baru
10. BitBtn di Panel2, dengan property :
  - Name : TLihatFaktur
  - Caption : Lihat Faktur
11. BitBtn di Panel2, dengan property :
  - Name : TTutup
  - Caption : Tutup

Disusun Oleh : Andri Heryandi, S.T.



# Pengolahan Data Penjualan

65

- Isi event OnClick tombol TTutup dengan

```
procedure TFPenjualan.TTutupClick(Sender: TObject);  
begin  
    Close;  
end;
```

# Pengolahan Data Penjualan

66

## □ Isi event OnClick pada tombol TLihatFaktur

```
procedure TFPenjualan.TLihatFakturClick(Sender: TObject);  
begin  
    Showmessage('Under construction');  
end;
```

# TIPE-TIPE FIELD

# Tipe-tipe Field

68

## □ Delphi mengenal 3 tipe field, yaitu

### 1. Field Data (Data Field)

Field data adalah tipe field yang secara fisik disimpan dalam file data. Jadi field ini dibuat ketika kita membuat struktur tabelnya.

### 2. Field Lookup (Lookup Field)

Field lookup adalah tipe field yang isi datanya berada pada tabel lain yang diakses berdasarkan kunci tertentu / relasi tertentu (misalnya Nama Pelanggan pada tabel Penjualan berasal dari field Nama yang ada di tabel Pelanggan)

### 3. Field Perhitungan (Calculated Field)

Field perhitungan adalah tipe field yang isi datanya merupakan hasil perhitungan atau operasi tertentu (misalnya field Keuntungan yang diambil berdasarkan perhitungan HargaJual dikurangi HargaBeli.

# Tipe-Tipe Field

69

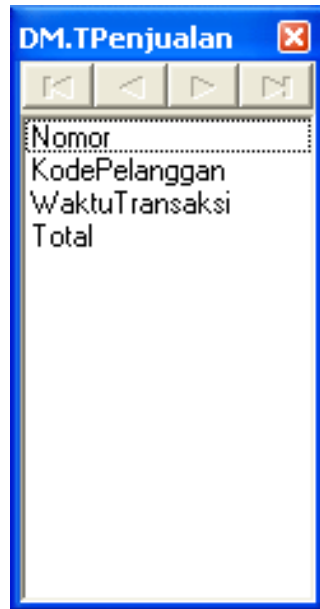
## □ Membuat Field Data

- ▣ Double klik di tabel yang akan dibuat fieldnya
- ▣ Klik kanan di Field Editor, pilih Add All Field atau Add Fields jika ingin memilih field yang akan ditambahkan ke table.

# Tipe-Tipe Field

70

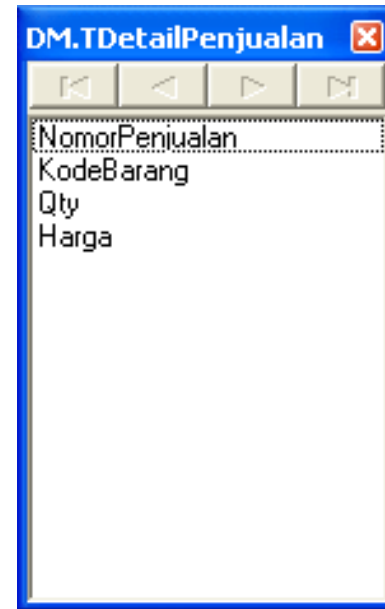
## □ Contoh Field Data



DM.TPenjualan

Nomor
KodePelanggan
WaktuTransaksi
Total

**Field Data di Tabel Penjualan**



DM.TDetailPenjualan

NomorPenjualan
KodeBarang
Qty
Harga

**Field Data di Tabel DetailPenjualan**

# Tipe-Tipe Field

71

## □ Membuat Field Lookup

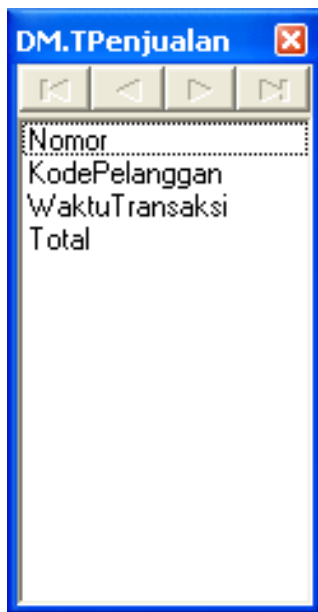
- ▣ Double klik di tabel yang akan dibuat fieldnya
- ▣ Klik kanan di Field Editor, pilih New Fields kemudian isi propertiesnya.

- Name : Nama field baru
- Component : Nama komponent field baru
- Type : Tipe data field baru
- Size : Ukuran field baru
- Field Type : Lookup
- Key Fields : Field kunci relasi ke tabel lain
- Dataset : Tabel sumber data yang akan dilookup
- Lookup Key : Field kunci yang ada di tabel sumber data
- Result Field : Field yang akan diambil datanya (dari tabel 2)

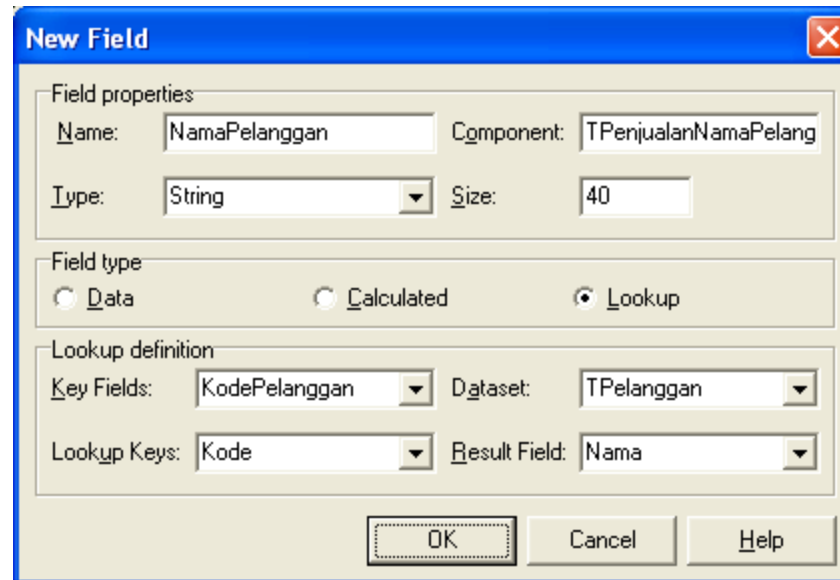
# Tipe-Tipe Field

72

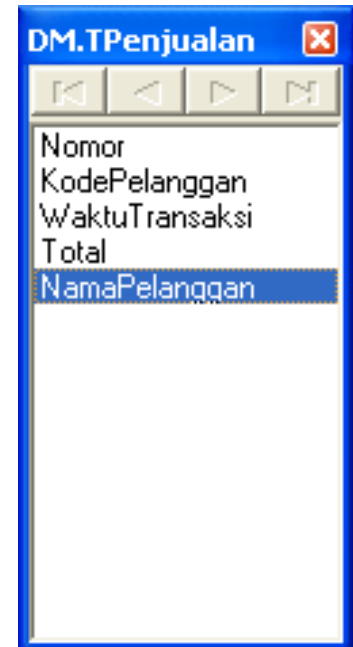
## □ Contoh Field Lookup



**Tabel TPenjualan**



**Pengaturan Field Baru  
(NamaPelanggan)**



**Tabel TPenjualan  
setelah ditambah  
field baru**



# Tipe-Tipe Field

73

**PENGOLAHAN DATA PENJUALAN**

Nomor	KodePelanggan	WaktuTransaksi	Total	NamaPelanggan
1	1	5/20/2008 8:00:00 AM	\$77,500.00	Andri Heryandi

**PERINCIAN FAKTUR PENJUALAN**

NomorPenjualan	KodeBarang	Qty	Harga
1	4970129727514	1	\$7,500.00
1	4970129727538	10	\$7,500.00

Hasil Lookup

Penjualan Baru   Lihat Faktur   Tutup

# Tipe-Tipe Field

74

- Membuat Field Perhitungan
  - ▣ Double klik di tabel yang akan dibuat fieldnya
  - ▣ Klik kanan di Field Editor, pilih New Fields kemudian isi propertiesnya.
    - Name : Nama field baru
    - Component : Nama komponent field baru
    - Type : Tipe data field baru
    - Size : Ukuran field baru
    - Field Type : Calculated
  - ▣ Cari event OnCalcField di table yang ditambah field perhitungan. Isi dengan rumus perhitungannya.

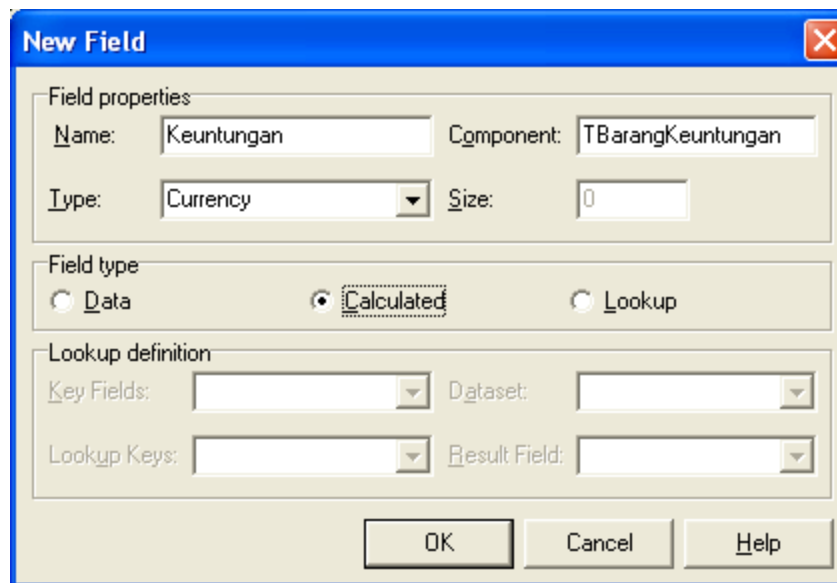
# Tipe-Tipe Field

75

## □ Contoh Field Perhitungan (Calculated Field)



**Tabel TBarang**



**Pengaturan Field Baru (Keuntungan)**



**Tabel TBarang  
setelah ditambah  
field baru**

# Tipe-Tipe Field

76

Cari event OnCalcField pada TBarang. Isi event tersebut dengan perintah di bawah ini :

```
procedure TDM.TBarangCalcFields(DataSet: TDataSet);  
begin  
    TBarangKeuntungan.Value:=TBarangHargaJual.value - TBarangHargaBeli.value;  
end;
```

Perhatian :

Field Perhitungan hanya bisa dilihat datanya ketika program telah dirun.

# Tipe-Tipe Field

77

**Pengolahan Data Barang**

PENGOLAHAN DATA BARANG

	HargaBeli	HargaJual	Stok	StokMinimal	Gambar	Keuntungan
▶	\$123.00	\$123.00	11	11	(GRAPHIC)	\$0.00
	\$40,000.00	\$45,000.00	20	3	(GRAPHIC)	\$5,000.00
	\$120,000.00	\$133,000.00	10	1	(GRAPHIC)	\$13,000.00
	\$5,000.00	\$7,500.00	10	4	(GRAPHIC)	\$2,500.00
	\$5,000.00	\$7,500.00	10	4	(GRAPHIC)	\$2,500.00
	\$150,000.00	\$175,000.00	3	0	(GRAPHIC)	\$25,000.00
	\$1,000.00	\$15,000.00	10	5	(GRAPHIC)	\$14,000.00
	\$30,000.00	\$40,000.00	3	3	(Graphic)	\$10,000.00
	\$45,000.00	\$55,000.00	10	5	(GRAPHIC)	\$10,000.00
	\$40,000.00	\$50,000.00	5	1	(Graphic)	\$10,000.00

Data Diurutkan Berdasarkan :  
☒ Kode  
☐ Nama Barang  
☐ Harga Jual  
☐ Stok

Pencarian Data :  
  
Cari

Tambah Edit Hapus Laporan Tutup

Field Hasil Perhitungan

## Kembali ke Pengolahan Data Penjualan

# Pengolahan Data Penjualan

79

## □ Buat Field baru di Tabel TBarang

### ▣ Field SubTotal

- Name : Keuntungan
- Type/Size : Currency/sudah ditentukan
- Field Type : Calculated
- Perintah yang ada dalam event OnCalcFields milik TBarang adalah :

```
procedure TDM.TBarangCalcFields(DataSet: TDataSet);  
begin  
    TBarangKeuntungan.Value:=TBarangHargaJual.value - TBarangHargaBeli.value;  
end;
```

# Pengolahan Data Penjualan

80

## □ Buat field-field berikut di table TPenjualan

### ▣ Field Nama Pelanggan

- Name : Nama Pelanggan
- Type/Size : String/50
- Field Type : Lookup
- Key Field : KodePelanggan
- Dataset : TPelanggan
- Lookup Key : Kode
- Result Field : Nama



# Pengolahan Data Penjualan

81

## ▣ Field Alamat Pelanggan

- Name : Alamat Pelanggan
- Type/Size : String/250
- Field Type : Lookup
- Key Field : KodePelanggan
- Dataset : TPelanggan
- Lookup Key : Kode
- Result Field : Alamat

# Pengolahan Data Penjualan

82

## ▣ Field Kota

- Name : Kota
- Type/Size : String/20
- Field Type : Lookup
- Key Field : KodePelanggan
- Dataset : TPelanggan
- Lookup Key : Kode
- Result Field : Kota

# Pengolahan Data Penjualan

83

## □ Buat field-field berikut di table TDetailPenjualan

### ▣ Field Nama Barang

- Name : Nama Barang
- Type/Size : String/50
- Field Type : Lookup
- Key Field : Kode
- Dataset : TBarang
- Lookup Key : KodeBarang
- Result Field : Nama

# Pengolahan Data Penjualan

84

## □ Field SubTotal

- Name : SubTotal
- Type/Size : Currency/sudah ditentukan
- Field Type : Calculated
- Perintah yang ada dalam event OnCalcFields milik TDetailPenjualan adalah :

```
procedure TDM.TDetailPenjualanCalcFields(DataSet: TDataSet);  
begin  
    TDetailPenjualanSubTotal.value:=TDetailPenjualanQty.value*TDetailPenjualanHarga.Value;  
end;
```

# Pengolahan Data Penjualan

85

DM.TBarang

Kode  
Nama  
HargaBeli  
HargaJual  
Stok  
StokMinimal  
Gambar  
Keuntungan

Fields

DM.TPenjualan

Nomor  
KodePelanggan  
WaktuTransaksi  
Total  
Nama Pelanggan  
Alamat  
Kota

DM.TDetailPenjualan

NomorPenjualan  
KodeBarang  
Qty  
Harga  
Nama Barang  
SubTotal

# Pengolahan Data Penjualan

86

- ❑ Save semua dokumen
- ❑ Test Program, lihat apakah field-field baru yang dibuat telah berfungsi dengan benar.

# RELASI MASTER/DETAIL

# Relasi Master/Detail

88

## □ Pengertian/Definisi

Relasi Master/Detail adalah suatu relasi antara 2 buah tabel yang mempunyai hubungan master/detail. Contoh relasi master detail adalah :

- Relasi antara Penjualan dan DetailPenjualan
  - Relasi antara Data Pegawai dengan Data Anak, Data Pendidikan, Data Prestasi
  - Relasi antara Data Mahasiswa dengan Data Nilai
  - Relasi antara Data Kelas dengan Data Mahasiswa
- Master/detail biasanya digunakan untuk relasi 2 tabel yang mempunyai derajat relasi One-To-Many.
- Jika sebuah data di tabel master dipilih, maka tabel detailnya hanya menampilkan data sesuai dengan data yang ditunjuk di tabel master. Contoh jika di tabel master dipilih Nomor Faktur 1, maka di tabel detail hanya akan menampilkan daftar detail penjualan nomor faktur 1 saja.



# Relasi Master/Detail

89

- Membuat relasi master detail
  - ▣ Klik tabel yang menjadi detail
  - ▣ Isi properti MasterSource dengan DataSource milik tabel master
  - ▣ Di properti MasterField klik tombol [...]
  - ▣ Di pilihan Detail Field isi dengan field yang ada di tabel detail yang berelasi dengan tabel master
  - ▣ Di pilihan Master Field isi dengan field yang ada di tabel master yang berelasi dengan tabel detail
  - ▣ Klik tombol Add untuk menambahkan relasi tersebut
  - ▣ Klik OK.

# Relasi Master/Detail

90

## □ Contoh Relasi Master Detail

The Field Link Designer dialog box is shown with the 'Available Indexes' set to 'Primary'. The 'Detail Fields' list contains 'NomorPenjualan' and 'KodeBarang'. The 'Master Fields' list contains 'Nomor', 'KodePelanggan', 'WaktuTransaksi', 'Total', and 'Nama Pelanggan'. The 'Add' button is visible between the two lists. The 'Joined Fields' list is empty, and the 'Delete' and 'Clear' buttons are visible next to it. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

The Field Link Designer dialog box is shown with the 'Available Indexes' set to 'Primary'. The 'Detail Fields' list contains 'KodeBarang'. The 'Master Fields' list contains 'KodePelanggan', 'WaktuTransaksi', 'Total', 'Nama Pelanggan', and 'Alamat'. The 'Add' button is visible between the two lists. The 'Joined Fields' list contains 'NomorPenjualan -> Nomor'. The 'Delete' and 'Clear' buttons are visible next to it. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

## Kembali ke Pengolahan Data Penjualan

# Pengolahan Data Penjualan

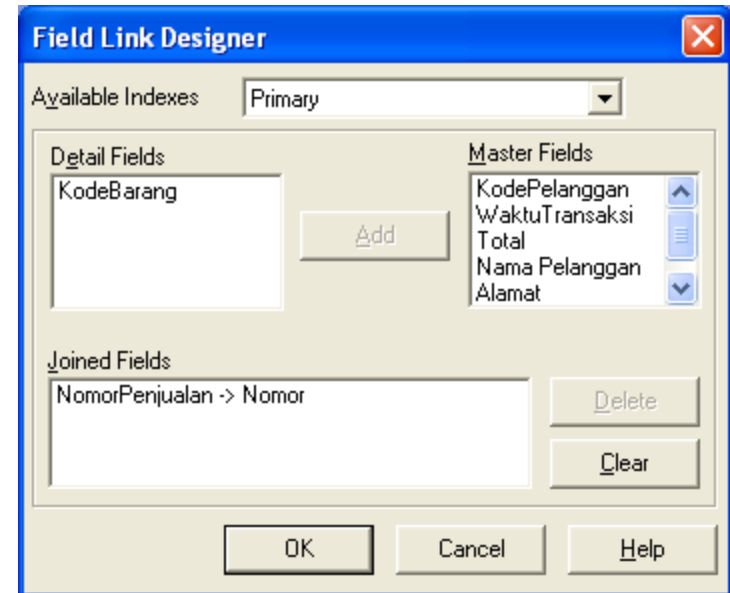
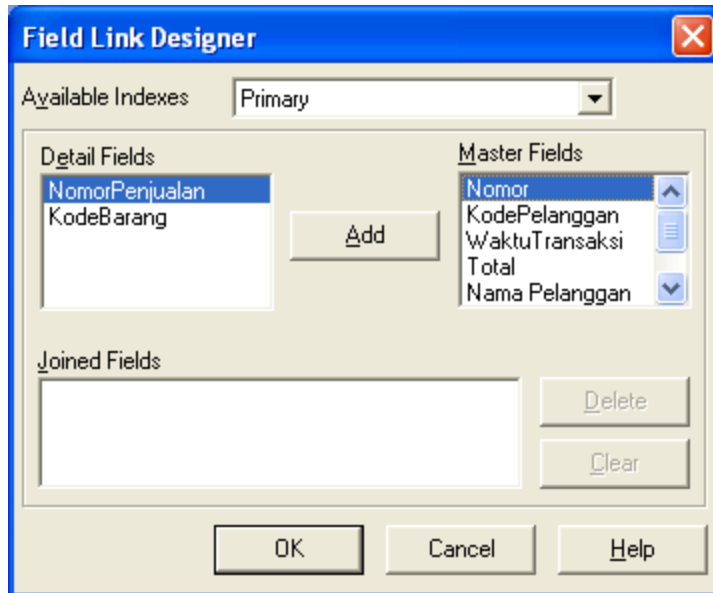
92

- Buat relasi master/detail antara tabel TPenjualan sebagai tabel master dan tabel TDetailPenjualan sebagai tabel detail. Caranya adalah :
  - Aktifkan Data Modul
  - Klik tabel yang menjadi detail (TDetailPenjualan)
  - Isi properti MasterSource : TPenjualan
  - Di properti MasterField klik tombol [...]
  - Di pilihan Detail Field isi dengan Field NomorPenjualan
  - Di pilihan Master Field isi dengan Field Nomor
  - Klik tombol Add (lihat slide berikutnya)
  - Klik OK.

# Pengolahan Data Penjualan

93

- Relasi Master Detail antar Tpenjualan dan TDetailPenjualan



Dengan pengaturan master/detail ini, maka setiap ada perpindahan record di tabel master (TPenjualan), maka isi dari tabel detail hanya akan menampilkan data dengan kode penjualan yang sedang ditunjuk di TPenjualan.

Disusun Oleh : Andri Heryandi, S.T.

# Pengolahan Data Penjualan

94

Sampai langkah ini, pengolahan penjualan baru dapat menampilkan data penjualan secara master detail saja. Tetapi proses penambahan penjualan baru belum dilakukan. Materi selanjutnya adalah bagaimana membuat menyusun transaksi penjualan.

# Pengolahan Data Penjualan

95

## □ Transaksi Penjualan Baru

- ▣ Hal yang harus diperhatikan ketika membuat transaksi penjualan baru.
  - Disarankan untuk membuat tabel temporary item penjualan. Jika transaksinya telah benar-benar terjadi (tidak dibatalkan), maka data yang ada di tabel item temporary penjualan akan dipindahkan ke tabel penjualan. Jika ternyata transaksi dibatalkan, maka data yang ada di tabel temporary item akan dihapus dan tidak ditambahkan ke tabel penjualan

# Pengolahan Data Penjualan

96

- Membuat Transaksi Penjualan Baru



# Pengolahan Data Penjualan

97

- Buat Tabel untuk temporary item penjualan

Restructure Paradox 7 Table: TempItemJual.db

Field roster:

	Field Name	Type	Size	Key
1	KodeBarang	A	13	*
2	Qty	I	1	

Enter a field name up to 25 characters long.

Table properties:

Validity Checks: [dropdown]  
Define...  
☐ 1. Required Field  
2. Minimum value: [text box]  
3. Maximum value: [text box]  
4. Default value: [text box]  
5. Picture: [text box]  
Assist...  
☐ Pack Table  
Save Save As... Cancel Help

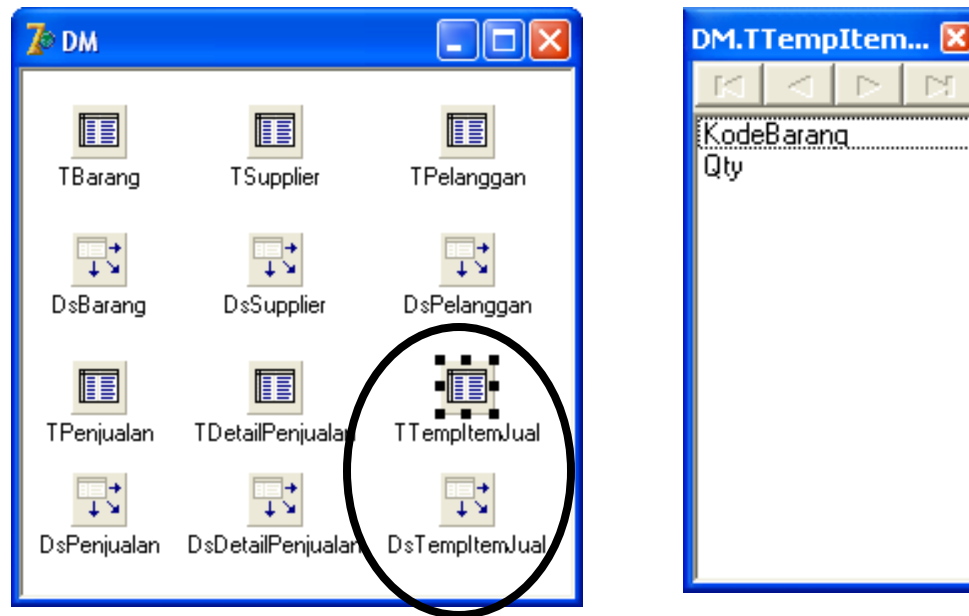
Nama File : TempItemJual.db

Disusun Oleh : Andri Heryandi, S.T.

# Pengolahan Data Penjualan

98

- Tempatkan Komponen Table di data module untuk TemplItemJual.DB dengan nama tabel TTemplItemJual. Tempatkan pula Datasource dengan Nama DsTemplItemJual.



# Pengolahan Data Penjualan

99

## □ Buat field-field berikut di table TTempltemJual

### ▣ Field Nama Barang

- Name : Nama Barang
- Type/Size : String/50
- Field Type : Lookup
- Key Field : KodeBarang
- Dataset : TBarang
- Lookup Key : Kode
- Result Field : Nama

**New Field**

Field properties

Name: Nama Barang Component: TTempltemJualNamaBa

Type: String Size: 50

Field type

☐ Data ☐ Calculated ☒ Lookup

Lookup definition

Key Fields: KodeBarang Dataset: TBarang

Lookup Keys: Kode Result Field: Nama

OK Cancel Help

# Pengolahan Data Penjualan

100

□ Buat field-field berikut di table TTemplItemJual

▣ Field Harga Jual

- Name : Harga Jual
- Type/Size : Currency/-
- Field Type : Lookup
- Key Field : KodeBarang
- Dataset : TBarang
- Lookup Key : Kode
- Result Field : HargaJual

The screenshot shows the 'New Field' dialog box with the following configuration:

- Field properties:**
  - Name: Harga Jual
  - Component: TTemplItemJualHargaJual
  - Type: Currency
  - Size: 0
- Field type:**
  - ☐ Data
  - ☐ Calculated
  - ☒ Lookup
- Lookup definition:**
  - Key Fields: KodeBarang
  - Dataset: TBarang
  - Lookup Keys: Kode
  - Result Field: HargaJual

# Pengolahan Data Penjualan

101

## □ Buat field-field berikut di table TTempItemJual

### □ Field Sub Total

- Name : Sub Total
- Type/Size : Currency/-
- Field Type : Calculated

The screenshot shows the 'New Field' dialog box. In the 'Field properties' section, the 'Name' is 'Sub Total' and the 'Component' is 'TTempItemJualSubTota'. The 'Type' is set to 'Currency' and the 'Size' is '0'. In the 'Field type' section, the 'Calculated' radio button is selected. The 'Lookup definition' section is empty. The 'OK' button is highlighted.

Isi event OnCalcField milik TTempItemJual dengan perintah di bawah ini

```
procedure TDM.TTempItemJualCalcFields(DataSet: TDataSet);  
begin  
    TTempItemJualSubTotal.value:=TTempItemJualQty.value*TTempItemJualHargaJual.Value;  
end;
```

# Pengolahan Data Penjualan

102

- Buat sebuah form untuk Pengisian Penjualan
  - ▣ File → New → Form
  - ▣ Ganti Name menjadi FlsiPenjualan
  - ▣ Isi Caption dengan “Pembuatan Faktur Baru”
  - ▣ File → Save dengan nama UFlsiPenjualan.pas
- Susun Form seperti pada gambar di slide berikutnya

# Pengolahan Data Penjualan

103

**Pembuatan Faktur Baru**

Kode:

Nama:

Alamat:

Kota:

No Telepon:

KodeBarang	Qty	Nama Barang	Harga Jual	Sub Total

Disusun Oleh : Andri Heryandi, S.T.

# Pengolahan Data Penjualan

104

Properti-properti penting komponen di FIsiPenjualan

1. Edit, dengan property :
  - Name : Ekode
  - Text : <kosongkan>
2. Edit, dengan property :
  - Name : ENama
  - Text : <kosongkan>
3. Edit, dengan property :
  - Name : EAlamat
  - Text : <kosongkan>
4. Edit, dengan property :
  - Name : EKota
  - Text : <kosongkan>
5. Edit, dengan property :
  - Name : ENoTelepon
  - Text : <kosongkan>
6. BitBtn, dengan property :
  - Name : TCari
  - Caption : Cari
5. DBGrid, dengan property :
  - Name : DBGrid1
  - DataSource : DM.DsTempltemJual
6. BitBtn, dengan property :
  - Name : TTambah
  - Caption : Tambah Item
7. BitBtn, dengan property :
  - Name : TEdit
  - Caption : Edit Item
8. BitBtn, dengan property :
  - Name : THapus
  - Caption : Hapus Item
9. BitBtn, dengan property :
  - Name : TSelesai
  - Caption : Selesai
10. BitBtn, dengan property :
  - Name : TDataPelanggan
  - Caption : Data Pelanggan

Disusun Oleh : Andri Heryendi, S.T.



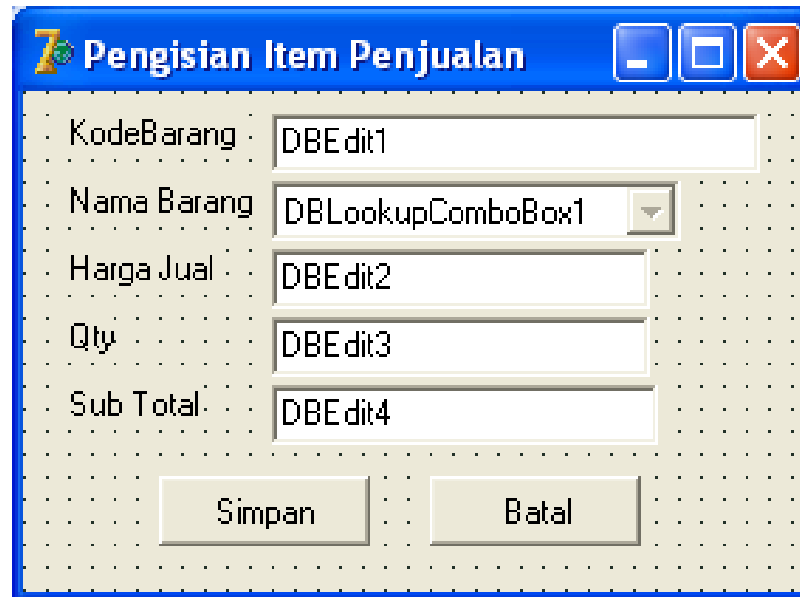
# Pengolahan Data Penjualan

105

- Buat sebuah form untuk Pengisian Item Penjualan
  - ▣ File → New → Form
  - ▣ Ganti Name menjadi FlsitemJual
  - ▣ Isi Caption dengan “Pengisian Item Jual”
  - ▣ File → Save dengan nama UFlsitemJual.pas
- Susun Form seperti pada gambar di slide berikutnya

# Pengolahan Data Penjualan

106



The screenshot shows a Windows application window titled "Pengisian Item Penjualan". The window has a blue title bar with standard minimize, maximize, and close buttons. The main area is a light gray grid. It contains five input fields, each with a label to its left: "KodeBarang" (text box labeled DBEdit1), "Nama Barang" (lookup combo box labeled DBLookupComboBox1), "Harga Jual" (text box labeled DBEdit2), "Qty" (text box labeled DBEdit3), and "Sub Total" (text box labeled DBEdit4). At the bottom, there are two buttons: "Simpan" (Save) and "Batal" (Cancel).

Field Label	Control Name
KodeBarang	DBEdit1
Nama Barang	DBLookupComboBox1
Harga Jual	DBEdit2
Qty	DBEdit3
Sub Total	DBEdit4

Buttons: Simpan, Batal

# Pengolahan Data Penjualan

107

Properti-properti penting komponen di FlsitemJual

1. DBEdit, dengan property :

- Name : DBEdit1
- DataSource : DM.DsTempltemJual
- DataField : KodeBarang

2. DBEdit, dengan property :

- Name : DBEdit2
- DataSource : DM.DsTempltemJual
- DataField : Harga Jual

3. DBEdit, dengan property :

- Name : DBEdit3
- DataSource : DM.DsTempltemJual
- DataField : Qty

4. DBEdit, dengan property :

- Name : DBEdit4
- DataSource : DM.DsTempltemJual
- DataField : Sub Total

5. DBLookupComboBox, dengan property :

- Name : DBLookupComboBox
- DataSource : DM.DsTempltemJual
- DataField : Nama Barang

Disusun Oleh : Andri Heryandi, S.T.

# Pengolahan Data Penjualan

108

- Isi event **OnClick** tombol **Penjualan Baru** yang ada di form **FPenjualan**

```
procedure TFPenjualan.TPenjualanBaruClick(Sender: TObject);
begin
    // Hapus Semua Data yang ada di Temporary Item Jual
    DM.TTempItemJual.First;
    While DM.TTempItemJual.Eof = false do
        DM.TTempItemJual.Delete;

    FISiPenjualan.EKode.Clear;
    FISiPenjualan.ENama.Clear;
    FISiPenjualan.EAlamat.Clear;
    FISiPenjualan.EKota.Clear;
    FISiPenjualan.ENoTelepon.Clear;

    FISiPenjualan.Showmodal; // Tampilkan Form Isi Penjualan
end;
```

# Pengolahan Data Penjualan

109

## □ Isi event OnClick tombol Cari yang ada di form FlsiPenjualan

```
procedure TFIsiPenjualan.TCariClick(Sender: TObject);
begin
    if DM.TPelanggan.Locate('Kode',EKode.Text,[])=true then
    begin
        ENama.Text:=DM.TPelanggan.FieldName('Nama').AsString;
        EAlamat.Text:=DM.TPelanggan.FieldName('Alamat').AsString;
        EKota.Text:=DM.TPelanggan.FieldName('Kota').AsString;
        ENoTelepon.Text:=DM.TPelanggan.FieldName('NoTelepon').AsString;
    end
    else
    begin
        Showmessage('Pelanggan Tidak Ditemukan');
        ENama.Clear;
        EAlamat.Clear;
        EKota.Clear;
        ENoTelepon.Clear;
    end;
end;
```

# Pengolahan Data Penjualan

110

- Isi event OnClick tombol Data Pelanggan yang ada di form FIsiPenjualan

```
procedure TFIsiPenjualan.TDataPelangganClick(Sender: TObject);  
begin  
    FPelanggan.Showmodal;  
end;
```

# Pengolahan Data Penjualan

111

- Isi event OnClick tombol Tambah Item yang ada di form FIsiPenjualan

```
procedure TFIsiPenjualan.TTambahClick(Sender: TObject);  
begin  
    DM.TTempItemJual.Append;  
    FIsiItemJual.Showmodal;  
end;
```

# Pengolahan Data Penjualan

112

- Isi event OnClick tombol Edit Item yang ada di form FIsiPenjualan

```
procedure TFIsiPenjualan.TEditClick(Sender: TObject);  
begin  
    DM.TTempItemJual.Edit;  
    FIsiItemJual.Showmodal;  
end;
```



# Pengolahan Data Penjualan

113

## □ Isi event OnClick tombol Hapus Item yang ada di form FlsiPenjualan

```
procedure TFIsiPenjualan.THapusClick(Sender: TObject);
var
    pesan:string;
begin
    pesan:='Anda yakin item berikut : '+#13+#13+
        'Kode Barang : '+DM.TTempItemJualKodeBarang.value+#13+
        'Nama Barang : '+DM.TTempItemJualNamaBarang.value+#13+#13+
        'Akan dihapus ?';
    if messagedlg(pesan,mtconfirmation,[mbyes,mbno],0)=mryes then
        begin
            DM.TTempItemJual.Delete;
        end;
    end;
```

# Pengolahan Data Penjualan

114

## □ Isi event OnClick tombol Selesai yang ada di form FlsiPenjualan

```
procedure TFIsiPenjualan.TSelesaiClick(Sender: TObject);
var
    Total:Currency;
begin
    if MessageDlg('Transaksi selesai?',mtconfirmation,[mbyes,mbno],0)=mryes then
    begin
        if DM.TPelanggan.Locate('Kode',EKode.Text,[])=false then
        begin
            Showmessage('Kode Pelanggan harus diisi dan terdaftar di Pelanggan');
            EKode.SetFocus;
            Exit;
        end;
        // Di sini, dipastikan bahwa pelanggan diisi dengan benar
        DM.TPenjualan.Append;
        DM.TPenjualanWaktuTransaksi.Value:=Now;
        DM.TPenjualanTotal.value:=0;// untuk sementara 0
        DM.TPenjualanKodePelanggan.value:=StrToInt(EKode.Text);
        DM.TPenjualan.Post; // Simpan sementara, agar nomor penjualan dapat
        dihasilkan otomatis
    end;
```

# Pengolahan Data Penjualan

115

```
DM.TTempItemJual.First;
Total:=0;
while DM.TTempItemJual.Eof = false do
begin
    // Simpan Temp Item Jual ke Detail Penjualan satu per satu
    DM.TDetailPenjualan.Append;
    DM.TDetailPenjualanNomorPenjualan.value:=DM.TPenjualanNomor.Value;
    // ini sebenarnya tidak usah karena sudah menggunakan master/detail
    DM.TDetailPenjualanKodeBarang.value:=DM.TTempItemJualKodeBarang.value;
    DM.TDetailPenjualanHarga.Value:=DM.TTempItemJualHargaJual.value;
    DM.TDetailPenjualanQty.value:=DM.TTempItemJualQty.value;
    DM.TDetailPenjualan.Post;

    // Hitung total
    Total:=Total+DM.TTempItemJualSubTotal.value;
```

# Pengolahan Data Penjualan

116

```
        //Update Stok di tabel barang
        if
DM.TBarang.Locate('Kode',DM.TDetailPenjualanKodeBarang.value,[])=true then
        begin
            DM.TBarang.Edit;
            DM.TBarangStok.value:=DM.TBarangStok.value -
DM.TDetailPenjualanQty.value;
            DM.TBarang.Post;
        end;

        DM.TTempItemJual.Next;
    end;

    // Update Total penjualan
    DM.TPenjualan.Edit;
    DM.TPenjualanTotal.value:=Total;
    DM.TPenjualan.Post;
    close;
end;
end;
```

# Pengolahan Data Penjualan

117

## □ Isi event OnClick pada Tombol Simpan yang ada di FlsiltemJual

```
procedure TFIsiItemJual.TSimpanClick(Sender: TObject);
begin
    if DM.TTempItemJualKodeBarang.IsNull then
    begin
        Showmessage('Kode Barang tidak boleh dikosongkan');
        DBEdit1.Setfocus;
        Exit;
    end;
    if DM.TTempItemJualNamaBarang.IsNull then
    begin
        Showmessage('Kode Barang tidak ditemukan');
        DBEdit1.Setfocus;
        Exit;
    end;
    if DM.TTempItemJualQty.IsNull then
    begin
        Showmessage('Qty tidak boleh dikosongkan');
        DBEdit3.Setfocus;
        Exit;
    end;
    if DM.TTempItemJualQty.value<=0 then
    begin
        Showmessage('Qty harus lebih besar dari 0');
        DBEdit3.Setfocus;
        Exit;
    end;

    DM.TTempItemJual.Post;
    Close;
end;
```

Disusun Oleh : Andri Heryandi, S.T.

# Pengolahan Data Penjualan

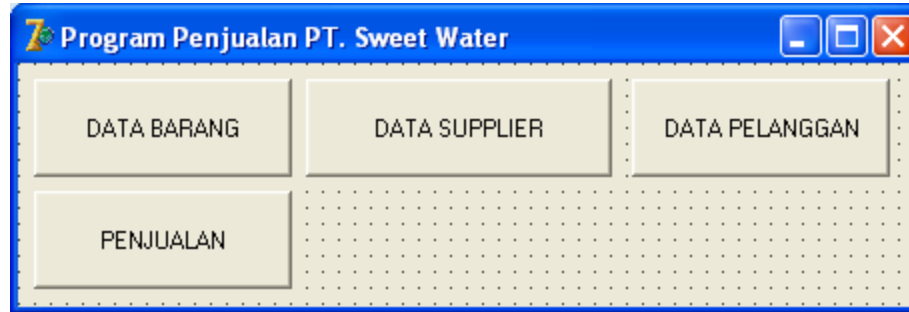
118

## □ Isi event OnClick pada Tombol Cancel yang ada di FIsiltemJual

```
procedure TFIsiItemJual.TBatalClick(Sender: TObject);
begin
    if MessageDlg('Pengisian data mau dibatalkan ?',mtconfirmation,[mbyes,mbno],0)=mryes then
    begin
        DM.TTempItemJual.Cancel;
        Close;
    end;
end;
```

# Pengolahan Data Penjualan

119



Isi event OnClick tombol TDataBarang yang ada di Futama dengan perintah di bawah ini :

```
procedure TFutama.TDataPenjualanClick(Sender: TObject);  
begin  
    FPenjualan.Showmodal;  
end;
```

# SELESAI (untuk sementara)

120

Keep connected to  
[kuliahdelphi@yahoo.com](mailto:kuliahdelphi@yahoo.com)  
<http://delphi.heryandi.net>

Disusun Oleh : Andri Heryandi, S.T.