

Pertemuan ke 12

Contoh Kasus [Lanjutan]



Pemrograman 2

Dosen : Eko Budi Setiawan, S.Kom

Universitas Komputer Indonesia

Sistem informasi penjualan barang

Tipe – Tipe Field

☐ *Field Data (Data Field)*

Tipe field yang secara fisik disimpan dalam file data. Jadi field ini dibuat ketika kita membuat struktur tabelnya. Untuk membuat Field Data di Delphi, double klik tabel yang akan dibuat fieldnya, kemudian klik kanan di Field Editor, pilih Add All Field atau Add Fields

☐ *Field Lookup (Lookup Field)*

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)

☐ *Field Perhitungan (Calculated Field)*

Tipe field yang isi datanya merupakan hasil perhitungan atau operasi tertentu (Misalnya field Keuntungan yang diambil berdasarkan perhitungan HargaJual dikurangi HargaBeli)



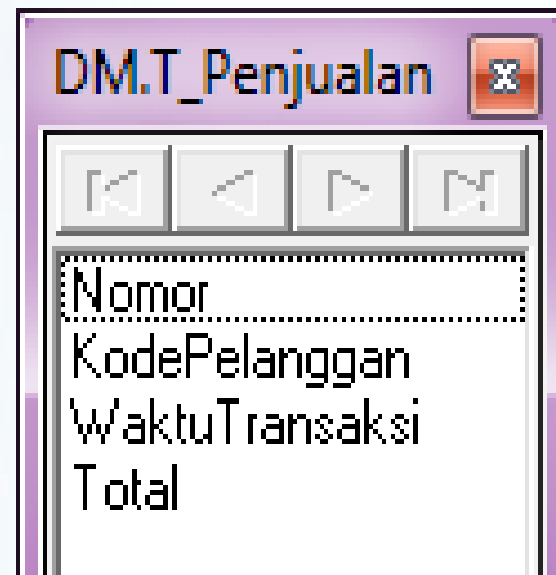
Membuat Field Lookup

- ❑ 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 komponen 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)



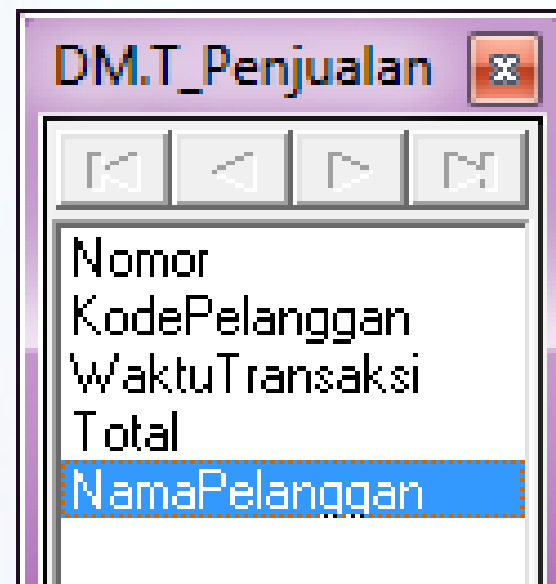
Contoh Field Lookup



DM.T_Penjualan

Nomor
KodePelanggan
WaktuTransaksi
Total

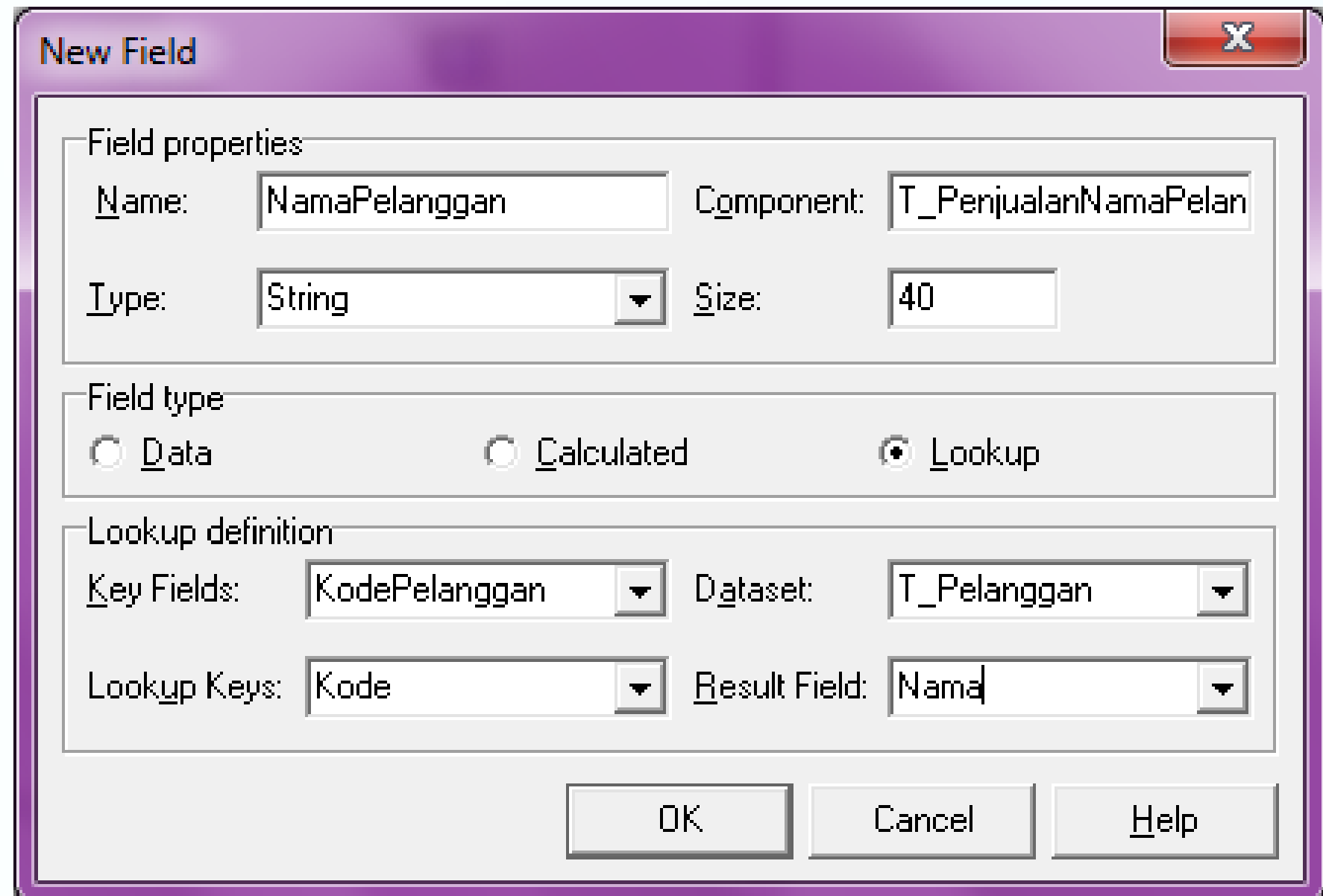
Tabel T_Penjualan



DM.T_Penjualan

Nomor
KodePelanggan
WaktuTransaksi
Total
NamaPelanggan

Tabel T_Penjualan
Sesudah ditambah
Field Baru



New Field

Field properties

Name:>NamaPelanggan Component: T_Penjualan>NamaPelan

Type: String Size: 40

Field type

☐ Data ☐ Calculated ☒ Lookup

Lookup definition

Key Fields: KodePelanggan Dataset: T_Pelanggan

Lookup Keys: Kode Result Field:>Nama

OK Cancel Help

Pengaturan Field Baru



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 komponen 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



Contoh Field Perhitungan (Calculated Field)

DM.T_Barang

Kode
Nama
HargaBeli
HargaJual
Stok
StokMinimal
Gambar

Tabel T_Barang

DM.T_Barang

Kode
Nama
HargaBeli
HargaJual
Stok
StokMinimal
Gambar
Keuntungan

Tabel T_Barang

Sesudah ditambah Field Baru

New Field

Field properties
Name: Keuntungan Component: T_BarangKeuntungan
Type: Currency Size: 0

Field type
☐ Data ☒ Calculated ☐ Lookup

Lookup definition
Key Fields: Dataset:
Lookup Keys: Result Field:

OK Cancel Help

Pengaturan Field Baru



Membuat Field Perhitungan

- ❑ Double Klik di T_Barang
- ❑ Klik Kanan di Field Editor, pilih New Fields kemudian isi propertiesnya

Name : Keuntungan
Component : T_BarangKeuntungan
Type : Currency
Size :
Field Type : Calculated

- ❑ Cari Event OnCalcField di T_Barang. Isi dengan rumus perhitungannya

```
procedure TDM.T_BarangCalcFields(DataSet: TDataSet);  
begin  
    T_BarangKeuntungan.Value:=T_BarangHargaJual.Value-T_BarangHargaBeli.Value;  
end;
```



Membuat Field

☐ Buatlah Field Nama Pelanggan (T_Penjualan) dan Field Keuntungan (T_Barang) sesuai dengan yang ada di halaman 4 & 6

☐ Buatlah field-field berikut di T_Penjualan

☐ Field Alamat Pelanggan

Name : Alamat Pelanggan

Type/Size : String/250

Field Type : Lookup

Key Field : KodePelanggan

DataSet : T_Pelanggan

Lookup Key : Kode

Result Field : Alamat

☐ Field Kota

Name : Kota

Type/Size : String/20

Field Type : Lookup

Key Field : KodePelanggan

DataSet : T_Pelanggan

Lookup Key : Kode

Result Field : Kota



Membuat Field

☐ Buatlah Field berikut di T_DetailPenjualan

☐ Field Nama Barang

Name	:	Nama Barang
Type/Size	:	String/50
Field Type	:	Lookup
Key Field	:	KodeBarang
DataSet	:	T_Barang
Lookup Key	:	Kode
Result Field	:	Nama

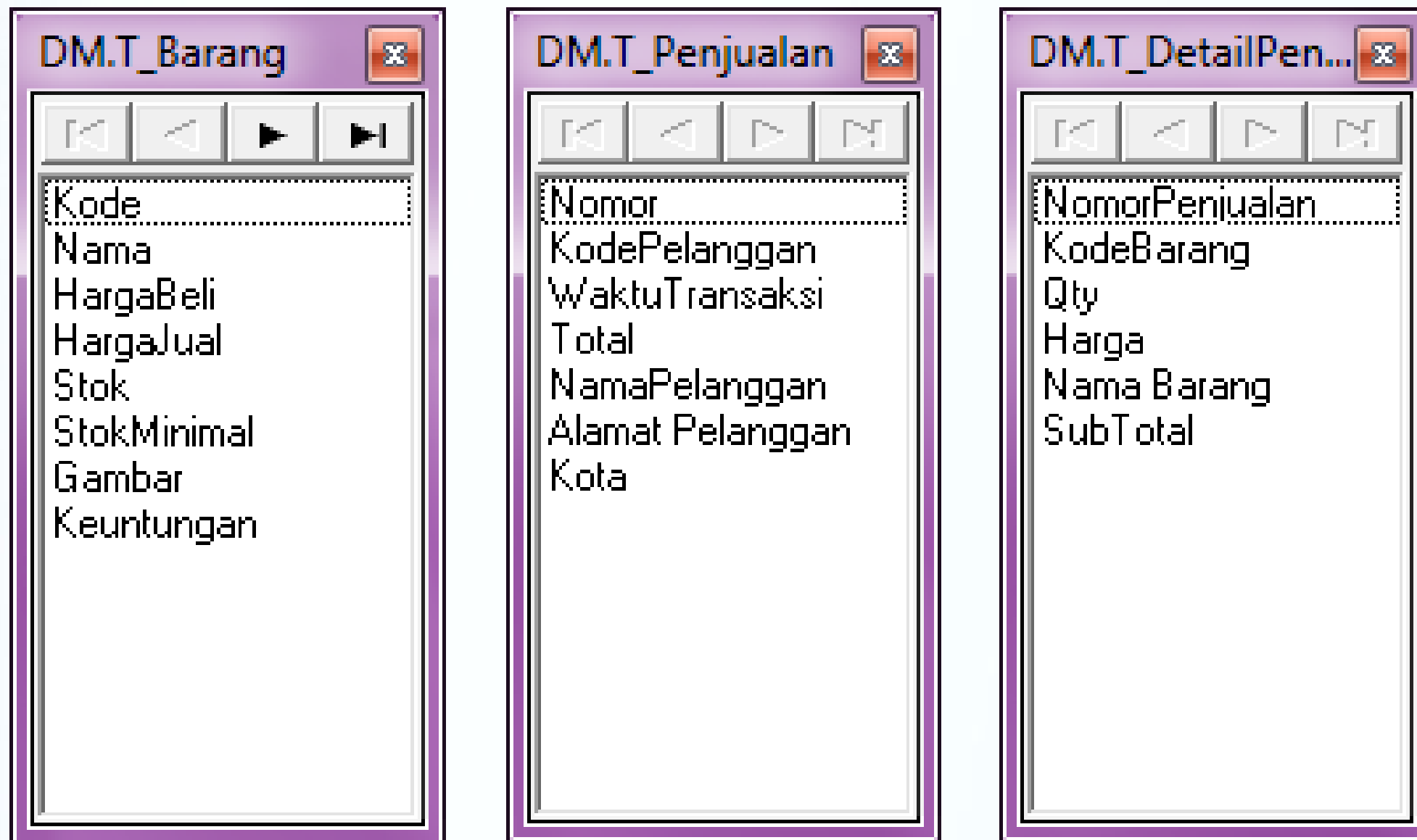
☐ Field Subtotal

Name	:	SubTotal
Type/Size	:	Currency
Field Type	:	Calculated

```
procedure TDM.T_DetailPenjualanCalcFields(DataSet: TDataSet);  
begin  
    T_DetailPenjualanSubTotal.Value:=T_DetailPenjualanQty.Value*T_DetailPenjualanHarga.Value;  
end;
```



Membuat Field



The image displays three database window screenshots, each with a title bar and a list of fields. The first window, 'DM.T_Barang', lists fields: Kode, Nama, HargaBeli, HargaJual, Stok, StokMinimal, Gambar, and Keuntungan. The second window, 'DM.T_Penjualan', lists fields: Nomor, KodePelanggan, WaktuTransaksi, Total, NamaPelanggan, Alamat Pelanggan, and Kota. The third window, 'DM.T_DetailPen...', lists fields: NomorPenjualan, KodeBarang, Qty, Harga, Nama Barang, and SubTotal. Each window has a toolbar with navigation buttons (back, forward, etc.) and a close button (X).

Field Name
Kode
Nama
HargaBeli
HargaJual
Stok
StokMinimal
Gambar
Keuntungan

Field Name
Nomor
KodePelanggan
WaktuTransaksi
Total
NamaPelanggan
Alamat Pelanggan
Kota

Field Name
NomorPenjualan
KodeBarang
Qty
Harga
Nama Barang
SubTotal

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



Relasi Master / Detail

- ❑ Relasi Master/Detail adalah suatu relasi antara 2 buah tabel yang mempunyai hubungan master/detail. Contoh :
 - ❑ 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 (satu ke banyak)
- ❑ Jika sebuah data di tabel master dipilih, maka tabel detailnya hanya menampilkan data sesuai dengan data yang ditunjuk ditabel master. Contoh jika tabel master dipilih Nomor Faktur 1, maka ditabel detail hanya akan menampilkan daftar detail penjualan nomor faktur 1 saja



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.



Contoh Relasi antar T_Penjualan dan T_DetailPenjualan

- ☐ Buat relasi master detail antara tabel T_Penjualan sebagai tabel master dan tabel T_DetailPenjualan sebagai tabel detail
- ☐ Aktifkan Data Modul
- ☐ Klik tabel yang menjadi detail (T_DetailPenjualan)
- ☐ Isi properti MasterSource : T_Penjualan
- ☐ Di Properti Masterfield klik tombol [...]
- ☐ Di pilihan Detail Field isi dengan field NomorPenjualan
- ☐ Di Pilihan Master Field isi dengan Field Nomor
- ☐ Klik tombol Add
- ☐ Klik OK



Contoh Relasi Master Detail

The dialog box is titled "Field Link Designer". It has a tab labeled "Available Indexes" with a dropdown menu showing "Primary". Below this, there are two main sections: "Detail Fields" and "Master Fields". The "Detail Fields" section contains a list box with "NomorPenjualan" and "KodeBarang". The "Master Fields" section contains a list box with "Nomor", "KodePelanggan", "WaktuTransaksi", "Total", and "NamaPelanggan". An "Add" button is located between these two sections. At the bottom, there is a "Joined Fields" section with an empty text box, a "Delete" button, and a "Clear" button. The bottom of the dialog has "OK", "Cancel", and "Help" buttons.

The dialog box is titled "Field Link Designer". It has a tab labeled "Available Indexes" with a dropdown menu showing "Primary". Below this, there are two main sections: "Detail Fields" and "Master Fields". The "Detail Fields" section contains a list box with "KodeBarang". The "Master Fields" section contains a list box with "KodePelanggan", "WaktuTransaksi", "Total", "NamaPelanggan", and "Alamat Pelanggan". An "Add" button is located between these two sections. At the bottom, there is a "Joined Fields" section with a text box containing "NomorPenjualan -> Nomor", a "Delete" button, and a "Clear" button. The bottom of the dialog has "OK", "Cancel", and "Help" buttons.

- ❑ Dengan perngaturan master/detail ini, maka setiap ada perpindahan record di tabel master (T_Penjualan), maka isi dari tabel detailnya hanya akan menampilkan data dengan kode penjualan yang sedang ditunjuk di T_Penjualan



Pengolahan Data Penjualan

- ❑ 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
- ❑ Buat tabel untuk temporary, simpan dengan nama tabel : TemplItemJual.DB

Field Name	Type	Size	Key
KodeBarang	A	13	*
Qty	I		

- ❑ Kemudian Tempatkan Komponen Table dan Datasource di DataModule untuk TemplItemJual.DB dengan nama tabel T_TemplItemJual dan DS_TemplItemJual. Jangan lupa menggunakan Add All Field di T_TemplItemJual

Buat Field – Field Lanjutan

❑ Buat Field-field berikut ditable TT_TempltemJual

❑ Field Nama Barang

Name : Nama Barang
Type/Size : String/50
Field Type : Lookup
Key Field : KodeBarang
DataSet : T_Barang
LookupKey : Kode
Result Field : Nama

The 'New Field' dialog box is shown with the following settings:

- Field properties:**
 - Name: Nama Barang
 - Component: mpltemJualNamaBarang
 - Type: string
 - Size: 50
- Field type:**
 - ☐ Data
 - ☐ Calculated
 - ☒ Lookup
- Lookup definition:**
 - Key Fields: KodeBarang
 - Dataset: T_Barang
 - Lookup Keys: Kode
 - Result Field: Nama

Buttons: OK, Cancel, Help

❑ Field Nama Barang

Name : Harga Jual
Type/Size : Currency
Field Type : Lookup
Key Field : KodeBarang
DataSet : T_Barang
LookupKey : Kode
Result Field : HargaJual

The 'New Field' dialog box is shown with the following settings:

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

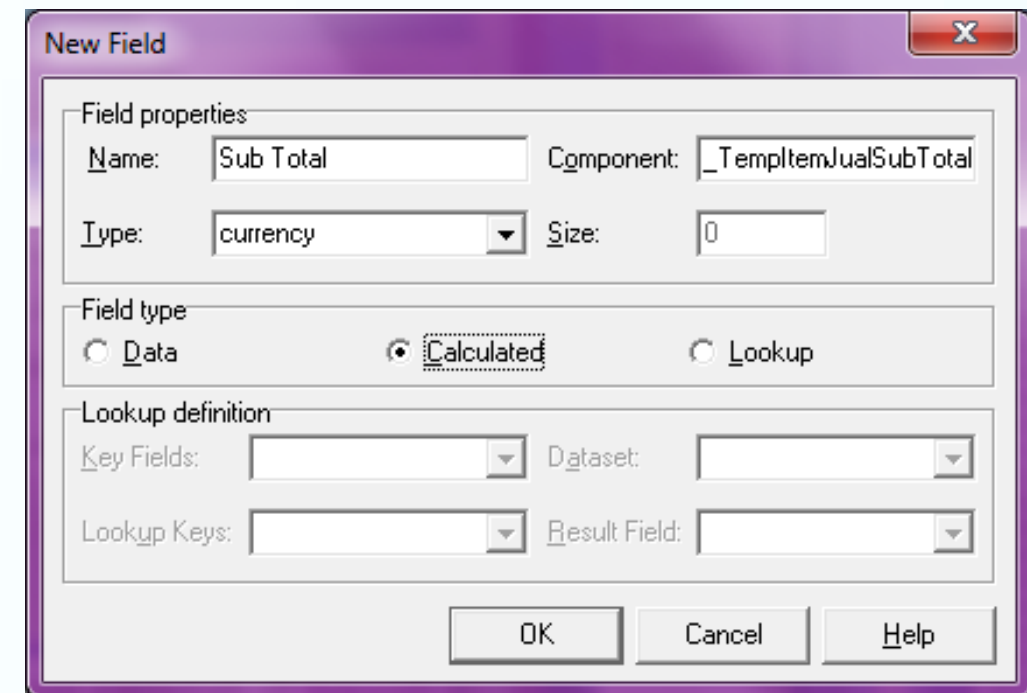
Buttons: OK, Cancel, Help

Buat Field - Field Lanjutan

- ☐ Buat Field-field ditable TT_TemplItemJual

- ☐ Field Nama Barang

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



Isi Event OnCalcField di T_TemplItemJual

```
procedure TDM.T_TemplItemJualCalcFields(DataSet: TDataSet);  
begin  
  T_TemplItemJualSubTotal.Value:=T_TemplItemJualQty.Value*T_TemplItemJualHargaJual.Value;  
end;
```

- ☐ Buat Sebuah Form untuk pengisian Penjualan

- ☐ File → New → Form

- ☐ Ganti Name menjadi FIsiPenjualan

- ☐ Isi Caption dengan “Pembuatan Faktur Baru”

- ☐ File → Save dengan nama UlsiPenjualan.Pas

- ☐ Susun Form seperti pada gambar di slide berikutnya

Pengolahan Data Penjualan (FIsiPenjualan)

Pembuatan Faktur Baru

Kode **Cari** **Data Pelanggan**

Nama

Alamat

Kota

No Telepon

KodeBarang	Qty	Nama Barang	Harga Jual	Sub Total
▶				

Tambah **Edit** **Hapus** **Selesai**



Properties Form IsiPenjualan (FIsiPenjualan)

1. Edit, dengan property

Name : Ekode

Text :

2. Edit

Name : ENama

Text :

3. Edit

Name : EAlamat

Text :

4. Edit

Name : EKota

Text :

5. Edit

Name : ENoTelepon

Text :

6. Button

Name : B_Cari

Caption : Cari

7. DBGrid

Name : DBGrid1

DataSource : DM.DS_TemplItemJual

8. Button

Name : B_Tambah

Caption : Tambah

9. Button

Name : B_Edit

Caption : Edit

10. Button

Name : B_Hapus

Caption : Hapus

11. Button

Name : B_Selesai

Caption : Selesai

12. Button

Name : B_DataPelanggan

Caption : Data Pelanggan



Form Item Penjualan (FlsiItemJual)

- ☐ Buat Sebuah Form untuk pengisian Item Penjualan
 - ☐ File → New → Form
 - ☐ Ganti Name menjadi FlsiItemJual
 - ☐ Isi Caption dengan “Pembuatan Item Jual”
 - ☐ File → Save dengan nama UlsiltemJual.Pas
- ☐ Susun Form seperti pada gambar di slide berikutnya



The image shows a screenshot of a Windows application window titled "Pembuatan Item Jual". The window has a purple title bar with standard minimize, maximize, and close buttons. The main area of the window is a grid with a dotted background. It contains five labels on the left and corresponding input controls on the right: "KodeBarang" with a text box labeled "DBEdit1", "Nama Barang" with a dropdown menu labeled "DBComboBox1", "Harga Jual" with a text box labeled "DBEdit2", "Qty" with a text box labeled "DBEdit3", and "Sub Total" with a text box labeled "DBEdit4". At the bottom of the grid, there are two buttons: "Simpan" and "Batal".

Properties Form IsiPenjualan (FIsiPenjualan)

1. DBEdit, dengan property
Name : DBEdit1
DataSource : DM.DS_TemplItemJual
DataField : KodeBarang
2. DBEdit
Name : DBEdit2
DataSource : DM.DS_TemplItemJual
DataField : Harga Jual
3. DBEdit
Name : DBEdit3
DataSource : DM.DS_TemplItemJual
DataField : Qty
4. DBEdit
Name : DBEdit4
DataSource : DM.DS_TemplItemJual
DataField : SubTotal
5. DBEdit
Name : DBLookupComboBox
DataSource : DM.DS_TemplItemJual
DataField : NamaBarang



Event OnClick Tombol Penjualan Baru di Form FPenjualan

```
procedure TFPenjualan.B_PenjualanBaruClick(Sender:
TObject);
begin
    //Hapus semua data yang ada di Temporary Item Jual
    DM.T_TempItemJual.First;
    While DM.T_TempItemJual.Eof=false do
        DM.T_TempItemJual.Delete;
    FIsiPenjualan.EKode.Clear;
    FIsiPenjualan.ENama.Clear;
    FIsiPenjualan.EAlamat.Clear;
    FIsiPenjualan.EKota.Clear;
    FIsiPenjualan.ENoTelepon.Clear;
    FIsiPenjualan.ShowModal; //tampilkan form isi penjualan
end;
```



Event OnClick Tombol Cari di Fisi Penjualan

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



Event OnClick Lain di FIsiPenjualan

- ❑ Event onClick tombol Data Pelanggan di Form FIsiPenjualan

```
procedure TFIsiPenjualan.B_DataPelangganClick(Sender: TObject);  
begin  
    FPelanggan.ShowModal;  
end;
```

- ❑ Event onClick tombol Tambah di Form FIsiPenjualan

```
procedure TFIsiPenjualan.B_TambahClick(Sender: TObject);  
begin  
    DM.T_TempItemJual.Append;  
    FIsiItemJual.ShowModal;  
end;
```

- ❑ Event onClick tombol Edit di Form FIsiPenjualan

```
procedure TFIsiPenjualan.B_EditClick(Sender: TObject);  
begin  
    DM.T_TempItemJual.Edit;  
    FIsiItemJual.ShowModal;  
end;
```



Event OnClick Lain di FIsiPenjualan

- ❑ Event onClick tombol Hapus di Form FIsiPenjualan

```
procedure TFIsiPenjualan.B_HapusClick(Sender: TObject);
var
  pesan : string;
begin
  pesan := 'Anda yakin item berikut :' + #13+#13+
    'Kode Barang : '+DM.T_TempItemJualKodeBarang.Value+#13+
    'Nama Barang : '+DM.T_TempItemJualNamaBarang.value+#13+#13+
    'Akan dihapus?';
  if MessageDlg(pesan,mtConfirmation,[mbyes,mbno],0)=mrYes then
  begin
    Dm.T_TempItemJual.Delete;
  end;
end;
```



Event OnClick Lain di FIsiPenjualan

❑ Event onClick tombol Selesai di Form FIsiPenjualan

```
procedure TFIsiPenjualan.B_SelesaiClick(Sender: TObject);
var
  Total:Currency;
begin
  if MessageDlg('Transaksi Selesai ?',mtConfirmation,[mbYes,mbNo],0)=mrYes then
  begin
    if DM.T_Pelanggan.Locate('Kode',EKode.Text,[])=false then
    begin
      ShowMessage('Kode Pelanggan harus diisi dan terdaftar di Pelanggan');
      EKode.SetFocus;
      Exit;
    end;
    DM.T_Penjualan.Append;
    DM.T_PenjualanWaktuTransaksi.Value:=Now;
    DM.T_PenjualanTotal.Value:=0;
    DM.T_PenjualanKodePelanggan.Value:=StrToInt(EKode.Text);
    DM.T_Penjualan.Post;
    DM.T_TempItemJual.First;
    Total:=0;
    while DM.T_TempItemJual.Eof=false do
    begin
      DM.T_DetailPenjualan.Append;
```



Event OnClick Lain di FIsiPenjualan

```
DM.T_DetailPenjualanNomorPenjualan.Value:=DM.T_PenjualanNomor.Value;
DM.T_DetailPenjualanKodeBarang.Value:=DM.T_TempItemJualKodeBarang.Value;
DM.T_DetailPenjualanQty.Value:=DM.T_TempItemJualQty.Value;
DM.T_DetailPenjualan.Post;
Total:=Total+DM.T_TempItemJualSubTotal.Value;
    //update stok di tabel barang
if
DM.T_Barang.Locate('Kode',DM.T_DetailPenjualanKodeBarang.Value,[])= true then
begin
    DM.T_Barang.Edit;
    DM.T_BarangStok.Value:=DM.T_BarangStok.Value-
    DM.T_DetailPenjualanQty.Value;
    DM.T_Barang.Post;
end;
DM.T_TempItemJual.Next;
end;
Dm.T_Penjualan.Edit;
DM.T_PenjualanTotal.Value:=Total;
DM.T_Penjualan.Post;
Close
end;
```



Event OnClick Tombol Simpan di FIsiItemJual

```
procedure TFIsiItemJual.B_SimpanClick(Sender: TObject);
begin
  if DM.T_TempItemJualKodeBarang.IsNull then
  begin
    ShowMessage('Kode Barang tidak boleh kosong');
    DBEdit1.SetFocus;
    exit;
  end;
  if DM.T_TempItemJualNamaBarang.IsNull then
  begin
    ShowMessage('Kode Barang tidak ditemukan');
    DBEdit1.SetFocus;
    exit;
  end;
  if DM.T_TempItemJualQty.IsNull then
  begin
    ShowMessage('Qty tidak boleh kosong');
    DBEdit3.SetFocus;
    Exit;
  end;
```



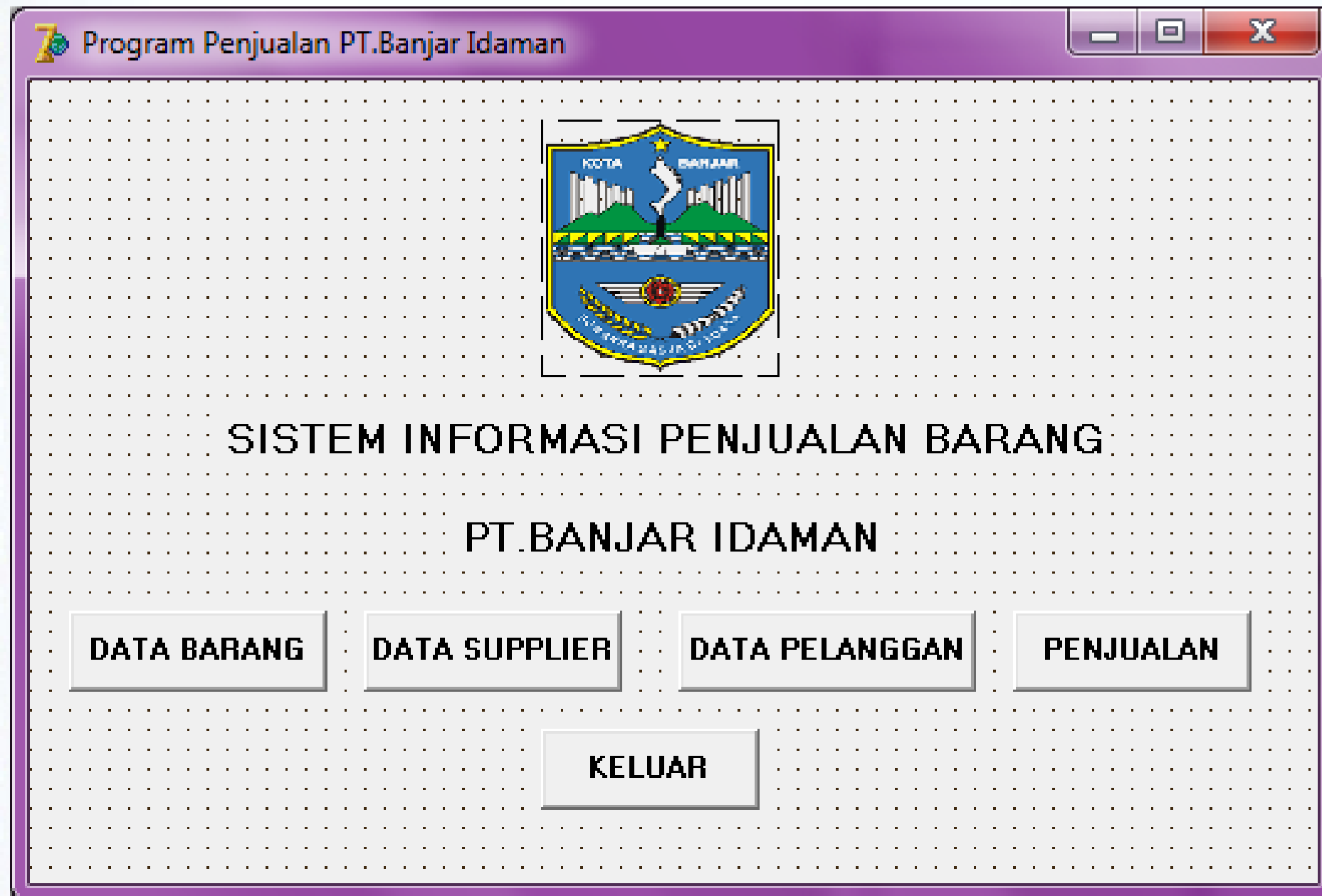
Event OnClick Tombol Simpan di FIsiItemJual

```
if DM.T_TempItemJualQty.Value<=0 then
begin
    ShowMessage('Qty harus lebih besar dari 0');
    DBEdit3.SetFocus;
    Exit;
end;
DM.T_TempItemJual.Post;
Close;
end;

procedure TFIsiItemJual.B_BatalClick(Sender: TObject);
begin
    if MessageDlg('Pengisian data mau dibatalkan
?',mtConfirmation,[mbYes,mbNo],0)=mrYes THEN
        begin
            DM.T_TempItemJual.Cancel;
            close;
        end;
    end;
end;
```



FUtama



```
procedure TFUtama.B_PenjualanClick(Sender: TObject);  
begin  
    FPenjualan.ShowModal;  
end;
```

