

# PEMROGRAMAN DATABASE : MEMBUAT FORM PENANGANAN TRANSAKSI (MULTI-TABEL)

*Oleh :*

*Teguh Wahyono*

*tegoeh@uksw.edu*

*Dipublikasikan dan didedikasikan  
untuk perkembangan pendidikan di Indonesia melalui*

## MateriKuliah.Com

### ***Lisensi Pemakaian Artikel:***

Seluruh artikel di **MateriKuliah.Com** dapat digunakan, dimodifikasi dan disebarakan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut Penulis. Hak Atas Kekayaan Intelektual setiap artikel di **MateriKuliah.Com** adalah milik Penulis masing-masing, dan mereka bersedia membagikan karya mereka semata-mata untuk perkembangan pendidikan di Indonesia. **MateriKuliah.Com** sangat berterima kasih untuk setiap artikel yang sudah Penulis kirimkan.

# MEMBUAT FORM

## PENANGANAN TRANSAKSI (MULTI-TABEL)

Pada bagian sebelumnya (tentang *Koneksi database menggunakan VB dengan ActiveX Data Object dan Koneksi database menggunakan VB dengan ADODB Connection*) sudah dibahas tentang bagaimana melakukan koneksi database baik dengan ADO Data Control maupun koneksi menggunakan modul ADODB. Tetapi pada kasus-kasus yang telah dibahas, form-form yang dibuat masih hanya menggunakan tabel tunggal. Misalnya form barang hanya berhubungan dengan satu tabel saja yaitu tabel barang. Demikian juga dengan form pelanggan hanya berhubungan dengan satu tabel pelanggan saja.

Pada bagian ini Anda akan diajak untuk mempelajari bagaimana pembuatan form yang memanfaatkan beberapa tabel di dalamnya. Sebagai contoh bahwa di dalam form transaksi penjualan barang (yang nanti akan dibahas cara pembuatannya di bagian ini), di dalamnya akan melibatkan beberapa tabel yang saling berelasi yaitu tabel Barang, tabel Pelanggan, tabel Penjualan dan beberapa tabel penting lainnya. File yang akan digunakan masih berhubungan dengan project yang dibuat pada Modul 5 (koneksi dengan modul ADODB).

### Rancangan Form Transaksi

Form transaksi yang akan dibuat pada bagian ini adalah form yang digunakan input nota penjualan seperti yang terlihat pada gambar di bawah.

Kode Barang	Nama Barang	Harga	Satuan	Jumlah Beli	Total
003	Pensil	Buah	2500	10	25000
005	Tempe	Buah	100	40	4000
001	Buku Tulis	Buah	3000	9	27000
002	Disket	Box	25000	10	250000

Gambar 6.1. Database yang akan digunakan

Pada gambar tersebut terlihat bahwa form input data penjualan melibatkan beberapa tabel database yaitu diantaranya tabel pelanggan, tabel barang dan tabel penjualan itu sendiri.

Selanjutnya untuk membuat form seperti di atas juga diperlukan beberapa tabel tambahan. Salah satunya adalah tabel yang digunakan untuk menyimpan data-data kasir di bawah ini.

Tabel Kasir

Field	Type	Size
Kode Kasir	Character	5
Nama Kasir	Character	30
Alamat	Charater	40

Tambahkan tabel di atas pada database penjualan, yang sebelumnya sudah dibuat dengan MS Access. Selain tabel kasir, Anda juga perlu membuat satu tabel temporary (tabel yang digunakan dalam kondisi

tertentu saja dan bukan tabel permanen) untuk menampung data item barang yang dibeli di dalam grid daftar barang.

Penjualantemp : Table			
	Field Name	Data Type	Script
	Nonota	Text	
	Tanggaljual	Date/Time	
	Kodepelanggan	Text	
	Kodekasir	Text	
	Kodebarang	Text	
	Jumlahbarang	Number	
	Total	Number	
	TotalBayar	Text	

Field Properties

General

Lookup

Field Size

10

Format

Input Mask

Caption

Default Value

Validation Rule

Validation Text

Required

No

Allow Zero Length

Yes

Indexed

No

Unicode Compression

Yes

IME Mode

No Control

IME Sentence Mode

None

Gambar 6.2. Database yang akan digunakan

## Membuat Form Transaksi

Selanjutnya akan dibuat form transaksi penjualan barang seperti pada rancangan di atas dengan menggunakan interface Visual Basic dan dikoneksikan dengan beberapa tabel yang berhubungan dengan penjualan barang.

### Bagian Induk Transaksi

Yang dimaksud bagian induk transaksi adalah bagian-bagian yang berisi field-field *parent table*. Sebagai contoh, misalnya jika sebuah nota transaksi penjualan barang, maka field-field *parent table*-nya adalah **Nomor Nota**, **Penjualan**, **Kode Pelanggan**, **Kode Kasir** dan **Kode Barang** yang diambil.

Sebagai langkah awal, buatlah satu form sederhana seperti di bawah ini.

The screenshot shows a Visual Basic form titled "Form1" with a grid layout. On the left, there are labels for "Nomor Nota", "Tanggal", "Kode Pelanggan", "Nama Pelanggan", "Kode Kasir", and "Nama Kasir". To the right of these labels are input controls: a text box for "Nomor Nota", a date picker for "Tanggal", a combo box for "Kode Pelanggan", a text box for "Nama Pelanggan", a combo box for "Kode Kasir", and a text box for "Nama Kasir". On the right side of the form, there is a list of control names with arrows pointing to the corresponding controls on the form: "Name : Text1" points to the "Nomor Nota" text box, "Name : DTPicker1" points to the "Tanggal" date picker, "Name : Combo1" points to the "Kode Pelanggan" combo box, "Name : Text2" points to the "Nama Pelanggan" text box, "Name : ComboKasir" points to the "Kode Kasir" combo box, and "Name : Text3" points to the "Nama Kasir" text box.

Gambar 6.3. Isi Modul

Aturlah *properties* masing-masing obyek dengan *name* seperti pada gambar di atas. Berikut adalah penjelasan dari setiap obyek tersebut.

1. Obyek **Text1**.

Digunakan untuk mengisi Nomor Nota. Dibuat menggunakan TextBox.

2. Obyek **DTPicker**.

Digunakan untuk mengisi Tanggal. Obyek ini dibuat dengan menggunakan Tool DTPicker yang dapat diambil dari deretan ToolBox seperti di bawah ini.



Gambar 5.4. Isi Modul

Jika menggunakan DTPicker, maka nantinya untuk menginputkan tanggal cukup memilih waktu yang diinginkan dengan menggunakan mouse seperti di bawah ini.



Gambar 6.4. Isi Modul

3. Obyek **ComboPelanggan**.

Digunakan untuk mengisi Kode Pelanggan. Obyek ini dibuat menggunakan Combo Box.

4. Obyek **Text2**.

Digunakan untuk menampilkan nama pelanggan. Obyek ini dibuat menggunakan TextBox.

5. Obyek **ComboKasir**.

Digunakan untuk mengisi Nomor Nota. Obyek ini dibuat menggunakan ComboBox.

6. Obyek **Text3**.

Digunakan untuk menampilkan nama kasir. Obyek ini dibuat menggunakan ComboBox.

Setelah membuat obyek-obyek tersebut di atas, mulailah dengan mengaktifkan database yang akan digunakan pada form ini dengan mengisi program pengaktifannya pada prosedur **Form Load**. Double klik pada form dan isikan skrip seperti di bawah ini.

```
Private Sub Form_Load()  
    'Mengaktifkan Prosedur Buka Database  
    bukaDB  
  
    Text1.Text = " "  
    Combokasir.Text = " "  
    Combobarang.Text = " "  
    Text2.Text = " "  
    Text3.Text = " "  
  
    'Mengaktifkan Tabel Pelanggan  
    rcplg.Open "Pelanggan", myDB, adStateOpen, adLockOptimistic, adCmdTable
```

```
'Mengisi Combo Pelanggan dengan Data Kode pelanggan
rcplg.MoveFirst
Do While Not rcplg.EOF
    combopelanggan.AddItem rcplg.Fields(0)
    rcplg.MoveNext
Loop

'Mengaktifkan Tabel Kasir
rcksr.Open "Kasir", myDB, adStateOpen, adLockOptimistic, adCmdTable

'Mengisi Combo Pelanggan dengan Data Kode Kasir
rcksr.MoveFirst
Do While Not rcksr.EOF
    Combokasir.AddItem rcksr.Fields(0)
    rcksr.MoveNext
Loop

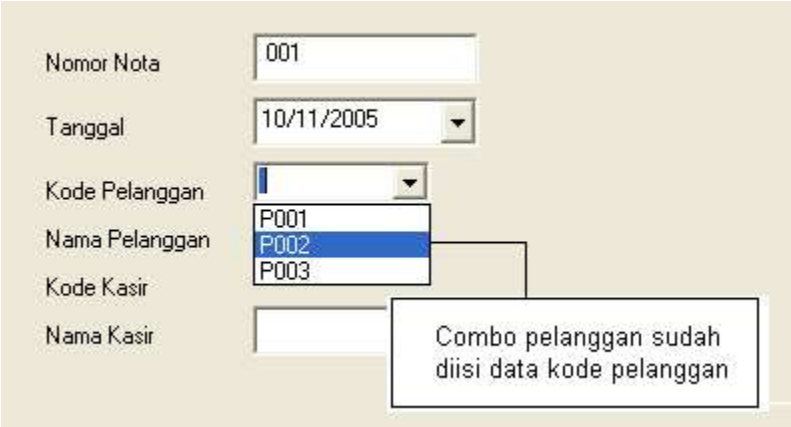
'Mengaktifkan tabel Penjualan dan Temporary-nya
rctrantemp.Open "Penjualantemp", myDB, adStateOpen, adLockOptimistic, adCmdTable
rctran.Open "Penjualan", myDB, adStateOpen, adLockOptimistic, adCmdTable

End Sub
```

Dari skrip program di atas bisa dilihat bahwa beberapa kotak combo telah diisi dengan data-data yang diinginkan. Sebagai contoh, untuk mengisi kotak Combo Pelanggan dengan Kode Pelanggan dibuat skrip seperti di bawah ini.

```
rcplg.MoveFirst
Do While Not rcplg.EOF
    combopelanggan.AddItem rcplg.Fields(0)
    rcplg.MoveNext
Loop
```

Maksud dari skrip di atas adalah bahwa Combo Pelanggan akan diisi dengan sejumlah data yang diambil dari “Field ke 0” tabel pelanggan. Field ke 0 yang dimaksud adalah field kode pelanggan. Jika program dijalankan, maka akan muncul kotak kombo seperti berikut.



Gambar 6.5. Isi Modul

Selanjutnya agar nama pelanggan secara otomatis muncul pada teks di bawah kode pelanggan tersebut, maka pada prosedur Click di kotak Combo Pelanggan diisi dengan program seperti di bawah ini.

```
Private Sub combopelanggan_Click()
rcplg.MoveFirst
If combopelanggan.Text = "" Then Exit Sub
Do While Not rcplg.EOF
    If rcplg.Fields(0).Value = combopelanggan.Text Then
        Text2.Text = rcplg.Fields(1).Value
        Exit Do
    End If
    rcplg.MoveNext
Loop
End Sub
```

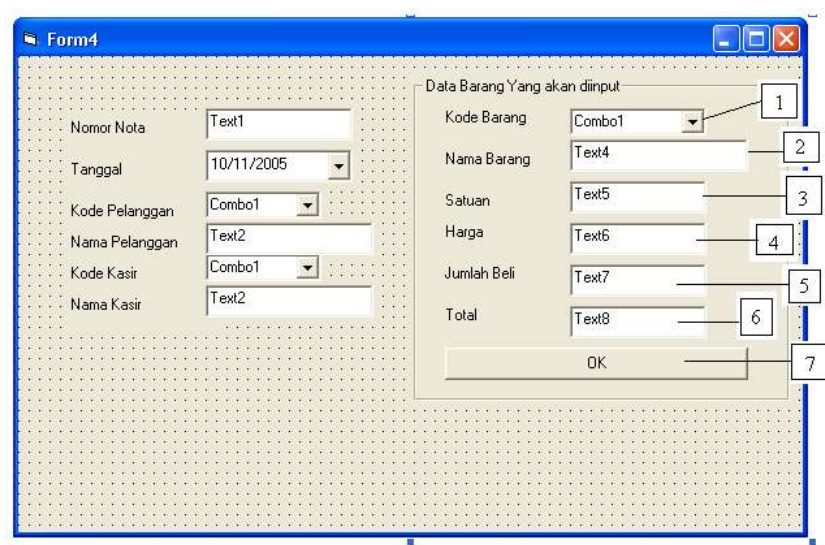
Sementara itu pada prosedur Click di kotak Combo Kasir diisi dengan program seperti di bawah ini.

```
Private Sub Combokasir_Click()  
rcksr.MoveFirst  
If Combokasir.Text = "" Then Exit Sub  
Do While Not rcksr.EOF  
    If rcksr.Fields(0).Value = Combokasir.Text Then  
        Text3.Text = rcksr.Fields(1).Value  
        Exit Do  
    End If  
    rcksr.MoveNext  
Loop  
End Sub
```

Dengan skrip seperti di atas, maka ketika dipilih kode kasir, nama kasir juga akan muncul pada text box di bawahnya.

Bagian Item Transaksi

Yang dimaksud item transaksi adalah daftar barang yang di beli oleh pelanggan. Untuk membuat item transaksi tersebut, maka buatlah dahulu bagian form sebagai berikut :



Gambar 6.6.

Obyek-obyek tambahan di atas (nomor 1 sampai 7) dapat dijelaskan sebagai berikut :

- 1. Obyek 1 -> Name : **ComboBarang**.  
Digunakan untuk memilih **Kode Barang** yang akan dibeli.
- 2. Obyek 2 -> Name : **Text4**.  
Digunakan untuk menampilkan **nama barang** sesuai dengan kode yang dipilih pada ComboBarang. Perlu dicatat bahwa properties **Enabled** pada obyek ini diisi dengan **False**.
- 3. Obyek 3 -> Name : **Text5**..  
Digunakan untuk menampilkan **satuan barang** sesuai dengan kode yang dipilih pada ComboBarang. Properties **Enabled** pada obyek ini juga diisi dengan **False**
- 4. Obyek 4 -> Name : **Text6**.  
Digunakan untuk menampilkan **harga barang** sesuai dengan kode yang dipilih pada ComboBarang. Properties **Enabled** obyek ini diisi dengan **False**



5. Obyek 5 -> Name : **Text7**.

Digunakan untuk mengisikan **jumlah beli** yaitu jumlah barang yang akan dibeli oleh pelanggan. Properties **Enabled** diisi dengan **True**

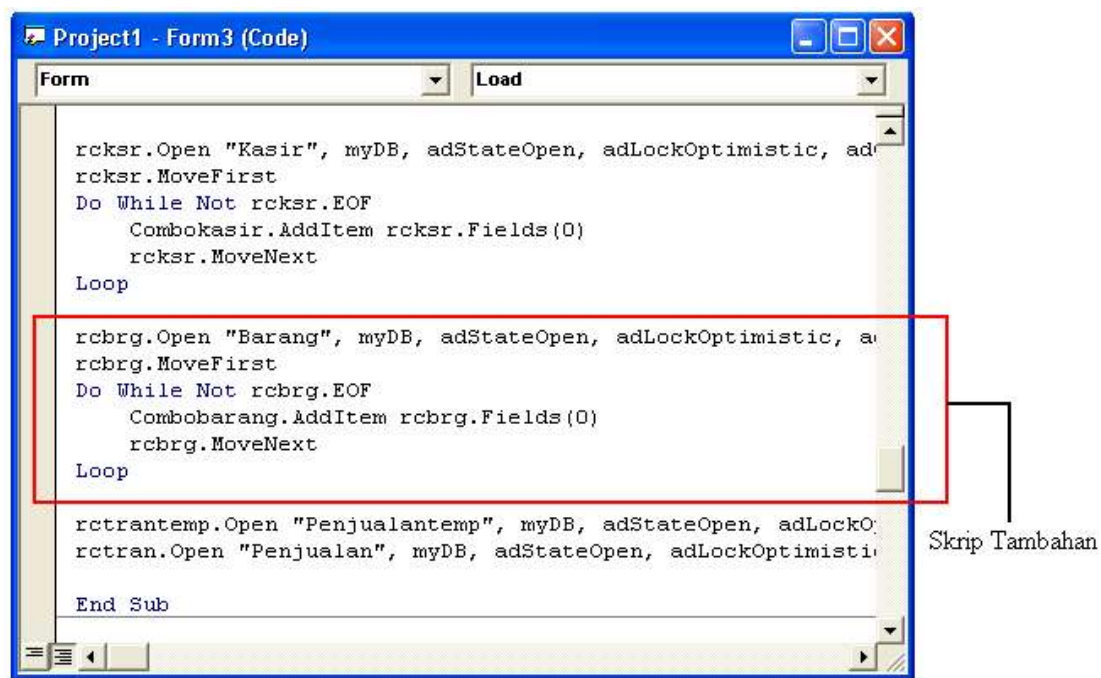
6. Obyek 6 -> Name : **Text8**.

Digunakan untuk menampilkan nilai **total beli** yaitu hasil perkalian antara jumlah beli dengan harga barang. Properties **Enabled** diisi dengan **False**.

7. Obyek 7 -> Name : **ComandOK**.

Digunakan untuk memastikan bahwa barang tersebut adalah barang barang yang dibeli pelanggan dan akan di catat di dalam database .

Agar combo barang terisi dengan daftar kode barang yang ada dalam database, maka tambahkan kode program di dalam **Form\_Load** sebagai berikut :



**Gambar 6.7. Skrip Tambahan**

Lebih jelasnya, bagian yang di “kotak” pada gambar di atas adalah seperti berikut :

```
rcbrg.Open "Barang", myDB, adStateOpen, adLockOptimistic, adCmdTable
rcbrg.MoveFirst
Do While Not rcbrg.EOF
    Combobarang.AddItem rcbrg.Fields(0)
    rcbrg.MoveNext
Loop
```

Hasilnya adalah bahwa ketika di klik combobarang, maka akan muncul daftar kode barang yang ada pada tabel database sebagai berikut :

Gambar 6.8. Skrip Tambahan

Selanjutnya akan dibuat aksi bahwa jika dipilih kode barang tertentu, maka pada teks-teks di bawahnya akan muncul nama, satuan dan harga dari kode barang yang dipilih pada combobarang tersebut. Untuk itu, double klik **combobarang** tersebut dan isikan di dalamnya (pada procedure *Click*) dengan skrip berikut.

```
Private Sub Combobarang_Click()  
rcbrg.MoveFirst  
  
`Jika combo barang kosong, maka tidak akan muncul identitas barang.  
If Combobarang.Text = "" Then Exit Sub  
  
`Jika combo diisi, maka akan muncul identitas barang.  
Do While Not rcbrg.EOF  
If rcbrg.Fields(0).Value=Combobarang.Text Then  
    Text4.Text = rcbrg.Fields(1).Value  
    Text5.Text = rcbrg.Fields(2).Value  
    Text6.Text = rcbrg.Fields(3).Value  
    Exit Do  
End If  
rcbrg.MoveNext  
Loop  
End Sub
```

Setelah diisi skrip di atas, maka setelah kode barang dipilih pada Combobarang, akan muncul identitas barang di bawahnya. Perhatikan gambar di bawah ini.

Gambar 6.9. Skrip Tambahan

TextBox **Total** akan diisi dengan perkalian dari harga dengan jumlah beli. Maka double klik **Text7** dan pada procedure **LostFocus**, isikan dengan program seperti di bawah ini :



```
Private Sub Text7_LostFocus()  
    harga = Val(Text6.Text)  
    jumbeli = Val(Text7.Text)  
    total = harga * jumbeli  
    Text8.Text = total  
End Sub
```

Program tersebut akan membuat aksi bahwa setelah Anda mengisi jumlah beli, maka otomatis total beli akan terisi dengan perkalian antara jumlah beli dengan harga barang.

A screenshot of a Windows form titled "Data Barang Yang akan diinput". The form has a light beige background and a thin border. It contains several input fields: "Kode Barang" with a dropdown menu showing "004", "Nama Barang" with a text box containing "Tempe", "Satuan" with a text box containing "Bungkus", "Harga" with a text box containing "1000", "Jumlah Beli" with a text box containing "10", and "Total" with a text box containing "10000". At the bottom of the form is a button labeled "OK".

Gambar 6.10. Skrip Tambahan

Pada gambar di atas terlihat bahwa setelah jumlah beli diisi dan ditekan **Tab** pada *keyboard*, bagian **Total** otomatis akan terhitung.

DataGrid untuk Daftar Barang yang dibeli

Langkah berikutnya, kita akan membuat grid sebagai area untuk memuat daftar barang yang akan dibeli. Buatlah grid sebagai berikut.

A screenshot of a Visual Basic IDE window titled "rm4 (Form)". The IDE shows a form with a light beige background. On the left side of the form is a DataGrid. On the right side is a data entry form titled "Data Barang Yang akan diinput" with fields for Kode Barang, Nama Barang, Satuan, Harga, Jumlah Beli, and Total, and an OK button. Two callout boxes are present: one pointing to the DataGrid with the text "DataGrid Dibuat dengan MHSFlexGrid", and another pointing to the DataGrid with the text "Data Grid sebagai tempat data barang".

Gambar 6.11. Data Grid

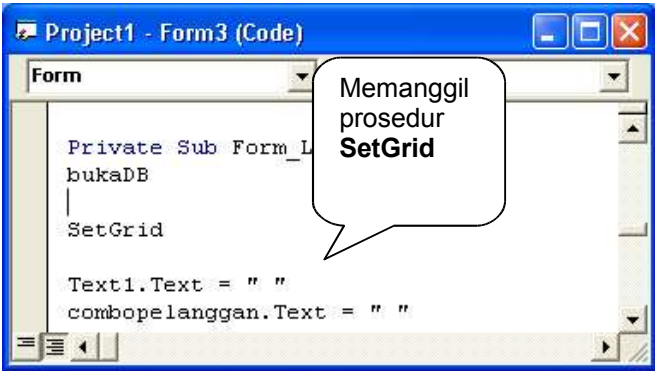
Pada *datagrid* tersebut, aturlah propertiesnya menjadi sebagai berikut :

Name : Grid1, Enabled : True

Buatlah pula sebuah *procedure* untuk mengatur bentuk grid tersebut, sesuai jumlah kolom yang diinginkan seperti di bawah ini.

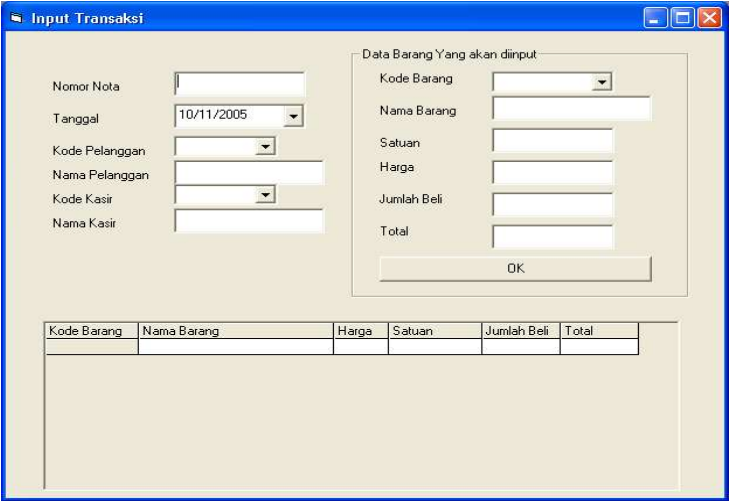
```
Private Sub SetGrid()  
`Menentukan jumlah kolom grid  
Grid1.Clear  
Grid1.Rows = 2  
Grid1.Cols = 6  
  
`Menentukan lebar masing-masing kolom  
`dalam satuan piksel  
Grid1.ColWidth(0) = 1200  
Grid1.ColWidth(1) = 2500  
Grid1.ColWidth(2) = 700  
Grid1.ColWidth(3) = 1200  
Grid1.ColWidth(4) = 1000  
Grid1.ColWidth(5) = 1000  
  
`Kolom 1 diisi dengan Kode Barang  
Grid1.Row = 0  
Grid1.Col = 0  
Grid1.Text = "Kode Barang"  
  
`Kolom 2 diisi dengan Nama Barang  
Grid1.Row = 0  
Grid1.Col = 1  
Grid1.Text = "Nama Barang"  
  
`Kolom 3 diisi dengan harga  
Grid1.Row = 0  
Grid1.Col = 2  
Grid1.Text = "Harga"  
  
`Kolom 4 diisi dengan Satuan  
Grid1.Row = 0  
Grid1.Col = 3  
Grid1.Text = "Satuan"  
  
`Kolom 5 diisi dengan Jumlah Beli  
Grid1.Row = 0  
Grid1.Col = 4  
Grid1.Text = "Jumlah Beli"  
  
`Kolom 6 diisi dengan Total  
Grid1.Row = 0  
Grid1.Col = 5  
Grid1.Text = "Total"  
End Sub
```

Untuk mengaktifkan pengaturan grid di atas, maka panggilah prosedur tersebut pada saat **Form Load**.



Gambar 6.12. Memanggil prosedur SetGrid di FormLoad

Jika prosedur setgrid dipanggil pada **Form Load**, maka jika dijalankan grid tersebut akan terbagi menjadi kolom-kolom seperti di bawah ini.



**Gambar 6.13. Grid terbagi menjadi kolom sesuai keinginan**

Yang harus dilakukan selanjutnya adalah mengisi tombol OK dengan prosedur seperti dibawah ini.

```
Private Sub CommandOK_Click()  
  
    'Jika jumlah beli belum diisi,  
    'Maka akan muncul pesan kesalahan  
    If Text7.Text = " " Then  
        MsgBox "Data jumlah beli belum diisi"  
        Exit Sub  
    End If  
  
    'Mengubah text7 (jumlah beli) menjadi integer  
    'supaya bisa diinput ke tabel database  
    jumbeli = Val(Text7.Text)  
  
    'Memasukkan data pada teks ke dalam Grid  
    Grid1.AddItem Combobarang.Text & Chr(9) & _  
        Text4.Text & Chr(9) & _  
        Text5.Text & Chr(9) & Text6.Text & Chr(9) & _  
        jumbeli & Chr(9) & Text8.Text  
  
    'Memasukkan data ke tabel penjualan temporary  
    rctrantemp.AddNew  
    rctrantemp!noNota = Text1.Text  
    rctrantemp!tanggaljual = DTPicker1.Value  
    rctrantemp!KodePelanggan = combopelanggan.Text  
    rctrantemp!KodeKasir = Combokasir.Text  
    rctrantemp!KodeBarang = Combobarang.Text  
    rctrantemp!Jumlahbarang = Val(Text7.Text)  
    rctrantemp!total = Val(Text8.Text)  
    rctrantemp.Update  
  
    'Setelah data dimasukkan, maka textbox di atas  
    'di bersihkan  
    Combobarang.Text = " "  
    Text4.Text = " "  
    Text5.Text = " "  
    Text6.Text = " "  
    Text7.Text = " "  
    Text8.Text = " "  
  
End Sub
```

Setelah tombol **OK** diisi program seperti di atas, maka penekanan tombol tersebut akan membuat data yang diinput masuk ke dalam grid.

Perhatikan gambar di bawah ini.

Kode Barang	Nama Barang	Harga	Satuan	Jumlah Beli	Total
001	Buku Tulis	Buah	3000	10	30000
002	Disket	Box	25000	20	500000

Gambar 6.14. Grid terisi data yang diinput

Tombol-tombol Penyimpanan Data

Langkah terakhir dari pembuatan form transaksi pada bagian ini adalah membuat tombol penyimpanan data dari grid, masuk ke dalam **tabel Penjualan**.

Buatlah tombol-tombol dari obyek **Command Button** seperti di bawah ini.

Gambar 6.15. Tombol Penyimpanan dan Batal

Pada gambar di atas terlihat tiga tombol baru pada form sebagai berikut :

- 1. Tombol **Simpan**  
Digunakan untuk penyimpanan data dari grid, masuk ke dalam **tabel Penjualan**. Propertiesnya adalah sebagai berikut.  
Name : CommandSimpan  
Caption : Simpan
- 2. Tombol **Batal**  
Digunakan untuk pembatalan transaksi. Propertiesnya adalah sebagai berikut.  
Name : CommandBatal  
Caption : Batal
- 3. Tombol **Keluar**  
Digunakan untuk keluar dari form. Propertiesnya adalah sebagai berikut.  
Name : CommandKeluar  
Caption : Keluar

Pada tombol **simpan**, isikan dengan prosedur sebagai berikut :

---

```
Private Sub CommandSimpan_Click()  
    rctrantemp.MoveFirst  
    Do While Not rctrantemp.EOF  
        rctran.AddNew  
        rctran!noNota = rctrantemp.Fields(0)  
        rctran!tanggaljual = rctrantemp.Fields(1)  
        rctran!KodePelanggan = rctrantemp.Fields(2)  
        rctran!KodeKasir = rctrantemp.Fields(3)  
        rctran!KodeBarang = rctrantemp.Fields(4)  
        rctran!Jumlahbarang = rctrantemp.Fields(5)  
        rctran!total = rctrantemp.Fields(6)  
        rctran.Update  
    rctrantemp.MoveNext  
    MsgBox "Data telah tersimpan"  
    SetGrid  
Loop
```

---

Pada tombol **batal**, isikan dengan prosedur sebagai berikut :

---

```
Private Sub CommandBatal_Click()  
    rctrantemp.MoveFirst  
    Do While Not rctrantemp.EOF  
        rctrantemp.Delete  
        rctrantemp.MoveNext  
    Loop  
    MsgBox "Data telah dibatalkan"  
    SetGrid  
End Sub
```

---

Pada tombol **Keluar**, isikan dengan prosedur sebagai berikut :

---

```
Private Sub Keluar_Click()  
    'Membersihkan isi tabel temporary  
    If rctrantemp.RecordCount > 0 Then  
        rctrantemp.MoveFirst  
        Do While Not rctrantemp.EOF  
            rctrantemp.Delete  
            rctrantemp.MoveNext  
        Loop  
    End If  
  
    'Keluar dari form ini  
    Unload Me  
End Sub
```

---

Dengan selesainya prosedur di atas, maka secara prinsip pembuatan form ini telah terselesaikan. Tetapi bukan berarti bahwa form tersebut sudah sempurna sebagai sebuah form transaksi. Validasi lanjut masih diperlukan dan tentunya tidak sekedar sarana memasukkan data saja.

# Tugas

*Buatlah form di atas, kumpulkan tanggal 18 Nopember 2005*