

# PEMROGRAMAN VISUAL FOXPRO



Microsoft Visual FoxPro atau sering disebut Visual FoxPro adalah salah satu aplikasi pengolah database yang menerapkan manajemen database relasional yang biasa disebut dengan **RDBMS (Relational Database Management System)** yang berorientasi objek.

## Pengertian Dasar *Object-Oriented Programming*

Bagi pemula dalam pemrograman *object-oriented* harus betul-betul memahami lima buah konsep, yaitu: *class*, *object*, *property*, *event*, dan *method/procedure*.

**Class** : adalah kumpulan instruksi/informasi yang menunjukkan bagaimana sebuah *object* harus bekerja.

**Object** : adalah sebuah bentuk konkret dari suatu *class*. Contoh *object* dalam Visual FoxPro adalah: *command button*, *textbox*, dll.

Setiap *object* memiliki:

- **Property (karakteristik)**, misalnya: warna *object* (*forecolor/backcolor*), tulisan yang ditampilkan *object* tertentu (*caption*), serta apakah *object* tersebut tampak atau tidak waktu program dijalankan (*visible*).
- **Event**, yaitu kejadian-kejadian yang terjadi atas *object* tertentu. Misalnya, waktu sebuah *object* dibuat/diciptakan (*init*), *object* tersebut diaktifkan (*activate*) dan waktu *object* itu di-click dengan mouse (*click event*)
- **Method/procedure** adalah program khusus (bisa dibuat sendiri) yang terkait dengan suatu *object* tertentu.

1. Pada Visual Foxpro terdapat beberapa istilah, diantaranya :

- 📁 Project Manager
- 📁 Database
- 📁 Tabel : Tabel Bebas dan Tabel yang termasuk dalam Database
- 📁 View
- 📁 Connection

2. Cara Mengaktifkan Visual Foxpro

Ada beberapa cara untuk mengaktifkan Visual Foxpro, seperti :

- a. Dari Taskbar : Start, Program, Microsoft Visual Studio, kemudian pilih Microsoft Visual Foxpro.
- b. Windows Explorer.
- c. Shortcut Microsoft Visual Foxpro.

Kemudian akan muncul layar menu utama Visual Foxpro yang terdiri dari :

- Baris Judul (*Title Bar*)
- Baris Menu (*Menu Bar*)

- Toolbar
- Jendela Command (*Command Window*)
- Baris Status (*Status Bar*)

### 3. Mengenal Jendela Window

Selain melalui baris menu, pemakai juga dapat melakukan instruksi secara langsung melalui jendela command (*Command Windows*). Cara mengaktifkannya :

- ✓ Click menu Window
- ✓ Click Command Window pada toolbar atau dengan menekan **Ctrl + F2**.

Instruksi – instruksi dalam mengoperasikan Visual Foxpro melalui jendela command. Seperti :

? : digunakan untuk menampilkan/mencetak data pada layar.

Misal : ?"Robi"

?Date() : digunakan untuk menampilkan tanggal sistem pada layar.

?Time() : digunakan untuk menampilkan waktu sistem pada layar.

Membuat tabel bebas dari jendela command :

- Mengatur default folder penyimpanan data dengan syntax

#### **SET DEFAULT TO NamaFolder**

Dimana : *NamaFolder* adalah nama folder yang akan digunakan untuk menyimpan data, penulisannya lengkap dengan nama drive dan path, jika nama folder ada karakter space maka penulisan nama folder harus diapit dengan tanda kutip ("")

Misalnya :

SET DEFAULT TO D:\DATAFOXPRO

Atau

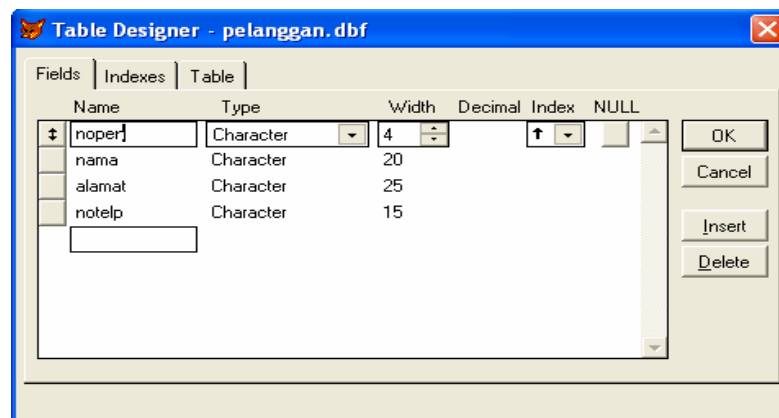
SET DEFAULT TO "D:\DATA FOXPRO"

- Membuat tabel bebas yaitu tabel yang tidak berada di dalam database.

Syntax : **CREATE NamaTabel**

Dimana : *NamaTabel* adalah Nama tabel yang akan dibuat, dianjurkan tanpa karakter space

Misalnya : CREATE PELANGGAN



Gambar 1. Jendela Table Designer

- Lalu isi nama-nama field, tipe, width dan index yang diperlukan sesuai dengan struktur tabel yang akan dibuat misalnya seperti tabel berikut.

Struktur Tabel Pelanggan

Field Name	Type	Width	Decimal	Index
Nopen	Character	4		↑ Ascending
Nama	Character	20		
Alamat	Character	25		
Notelp	Character	15		

### Operasi – operasi pada Tabel :

- Tampilkan tabel dengan :
  - ❖ Browse
  - ❖ List
  - ❖ List Off
  - ❖ List fields nama\_field. Misal : List Fields nama
  - ❖ Display (menampilkan tabel pada layar)
- Modifikasi tabel/ mengubah struktur tabel  
BU : Modify Structure atau Modi Stru
- Tambah Record  
BU : Append atau Appe
- Hapus Record  
BU : Delete [scope] nomor\_record  
Ket : scope : - All : Semua record  
Misal : Delete all atau Dele all  
- Next : dari record yang aktif sapai record yang keN.  
Misal : Delete next record 4 atau Dele next reco 4  
- Record : Record ke N.  
Misal : Delete record 2 atau Dele reco 2
- Menyembunyikan record yang sudah dihapus.  
BU : Set delete on atau Set dele on
- Menampilkan record yang sudah dihapus.  
BU : Set delete off atau Set dele off
- Mengaktifkan kembali record yang sudah dihapus.  
BU : Recall [scope] nomor\_record  
Misal : Recall record 2 atau Reca reco 2
- Menghilangkan record yang sudah dihapus secara permanen.  
BU : Pack
- Mengaktifkan tabel  
BU : Use nama\_tabel  
Misal : Use pelanggan

- Mengubah / mengganti / mengisi data  
 BU : Replace [scope] field with statement  
 Misal : Replace nama with "Digdoyo" :      mengubah record awal pada field nama dengan nama Digdoyo.  
  
 Goto 2  
 Replace Kota with "Palangkaraya" :      mengubah record kedua pada field Kota dengan Kota Palangkaraya.  
  
 Replace All Kota with "Jakarta" :      mengubah semua record pada field Kota menjadi Jakarta.
- Menutup semua tabel yang aktif  
 BU : Close all
- Keluar dari Visual Foxpro  
 BU : Quit

### Latihan 1

1. Buatlah sebuah tabel dengan nama **Siswa.Dbfc**. Berikut struktur tabelnya

Name	Type	Width	Keterangan
Nis	Character	8	Index (Reguler),(Ascending)
Nama	Character	25	
Alamat	Character	35	
Kota	Character	10	

2. Kemudian input datanya sebagai berikut:

Nis	Nama	Alamat	Kota
20060201	RIZA WULAN	Jl. Mawar no.32	Jakarta
20060202	DESI SARI	Jl. Melati no.12	Bandung
20060203	NONENG	Jl. Kamboja no.1	Denpasar
20060204	BAGUS	Jl. Utan Kayu	Jakarta

Melalui command window berikan perintah-perintah berikut :

- ❖ List
- ❖ List Off
- ❖ List fields Nama, Kota
- ❖ Display
- ❖ Browse

Perhatikan apa yang dihasilkan dilayar visual foxpro

# POINTER



Pointer adalah penunjuk record yang berbentuk tanda anak panah kecil di depan sebuah record yang berfungsi menunjukkan record yang sedang aktif sekarang. Umumnya pada saat tabel dibuka, pointer berada pada record pertama.

Operasi yang berkaitan dengan keberadaan pointer diantaranya :

1. Recno()

BU : ?Recno()

Misal : ?Recno()

Browse atau display

2. Goto

BU : Goto nomor\_record atau Go nomor\_record

Misal : Go 2

Browse

3. Pindah ke record pertama (BOF/Begining of File)

BU : Go Top

Misal : Go top

Brow

4. Pindah ke record terakhir (EOF/ End of File)

BU : Go Bottom

Misal : Go bottom

Brow

5. SKIP = digunakan untuk pindah dari record yang sedang aktif ke record berikutnya.

BU : Skip

?Recno()

Brow

Misal : Go top

Skip

?Recno()

Skip 2

?Recno()

Skip-1

?Recno()

## PENCARIAN DATA



Dalam pengolahan data tidak dapat lepas dengan aktifitas pencarian yaitu kegiatan membandingkan data yang disimpan dalam tabel database dengan data yang diinputkan atau data yang diinginkan. Pencarian dilakukan untuk memenuhi atau memberikan informasi dari hasil pengolahan data yang sesuai dengan keinginan pemakainya. Perintah pencarian dalam Visual FoxPro ada dua macam yaitu :

### 1. Locate

Berfungsi untuk meletakkan pointer record ke record yang dicari.

BU : Locate for nama\_field = statement

Misal : Locate for Kota = "Jakarta"

?Found()

Browse

Dapat juga digunakan perintah **?Found()** untuk mengetahui apakah yang dicari ketemu atau tidak.

### 2. Seek

Pencarian record tertentu dapat dilakukan dengan cepat seandainya file indeks digunakan. Namun tentu saja data untuk pencarian record harus sama dengan isi field yang ada.

BU : Seek nama\_record\_yang\_dicari

Misal : Use pelanggan

Set order to tagindex

Seek "1112"

?Found()

?Nopen,nama,alamat

Browse

Dengan perintah Locate maka pencarian akan dilakukan dari awal record sampai record yang dicari ditemukan sedangkan pencarian menggunakan perintah Seek pencarian dilakukan dengan cara selalu membagi dua jumlah record yang ada dan membandingkan data yang dicari dengan data yang ada dalam tabel. Oleh karena itu pencarian data dengan perintah Seek akan dilakukan lebih cepat jika dibandingkan dengan perintah Locate dan pencarian data dengan perintah Seek memerlukan sebuah file index.

## BEKERJA DENGAN FORM



Form adalah interface yang digunakan sebagai interaksi antara pengguna dengan program aplikasi. Dengan kata lain membuat *form* adalah merancang format *interface* atau format dokumen atau bentuk tampilan dimana data dikomunikasikan dalam bentuk yang dapat dimengerti oleh pengguna.

Pada *Visual FoxPro*, *form* dapat dibuat dengan dua cara yaitu dengan *form designer* (secara manual) dan *Form wizard* (dipandu komputer). Agar dapat memahami inti pembuatan *form* secara keseluruhan lebih baik kita menggunakan *Form Designer*. Sedangkan *wizard* digunakan untuk membuat program *entry* data yang sederhana, dan kita hanya perlu menjawab serangkaian pertanyaan yang akan diberikan oleh *Form Wizard* lalu *Visual FoxPro* akan membuatkan *Form* tersebut secara otomatis.

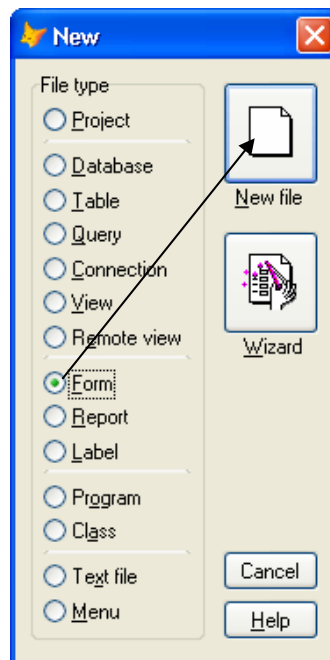
### Membuat Form Baru

Langkah-langkah awal membuat *form* dari *Form Designer*

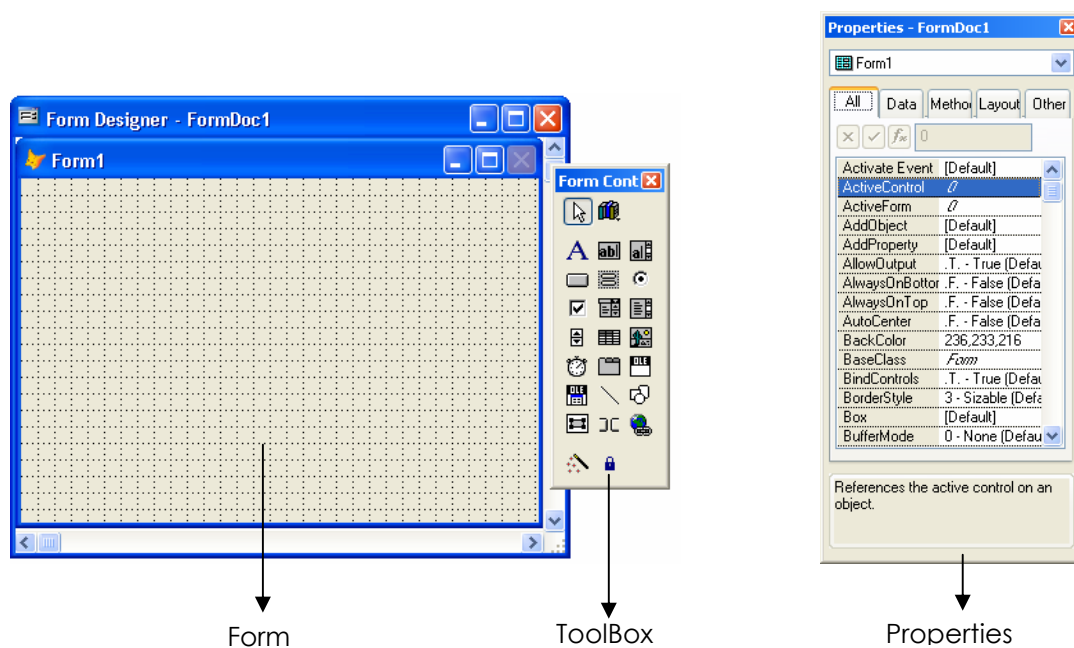
1. Pada *project manager*, click *tab documents*
2. Click *icon forms*
3. Click tombol "New..."

Kotak dialog *New* akan muncul, pilih *Form*.

4. Click tombol *New Form* untuk membuat *Form* menggunakan *Form Designer*, kemudian akan ditampilkan *Form Designer*, *form control* dan *properties*.



Gambar 2. Dialog New



Gambar 3. Jendela Form – Form Control Toolbox – Jendela Properties

## Beberapa Property Umum pada Form

### 1. Name (nama)

Property ini sangat penting, merupakan nama *object* itu sendiri dan dipakai lebih lanjut untuk mengamati *object* yang bersangkutan. Penamaan *object* memiliki beberapa konvensi, misalnya: untuk *form* diawali *frm*, untuk *command button* diawali dengan *cmd*, untuk *textbox* diawali *txt*, *dsb*.

### 2. Caption (judul)

Property ini mengandung teks yang akan ditampilkan seandainya *object* tersebut memiliki judul. *Caption* biasanya harus ditentukan untuk *form* dan untuk *command button* serta halaman-halaman dari *pageframe*.

### 3. Visible (terlihat)

Apabila *visible* ditentukan *.T. (true)*, maka *object* itu akan terlihat waktu dijalankan sedangkan bila ditentukan *.F. (false)*, maka *object* itu tidak terlihat waktu program dijalankan.

### 4. Enabled

Apabila *enabled* ditentukan *.T. (true)* maka *object* yang bersangkutan dapat dimanipulasi (*di-click*, diubah isinya, *dsb*) sedangkan jika *enabled* adalah *.F.* maka hal tersebut tidak bisa dilakukan.

### 5. Value (nilai awal)

Secara umum, merupakan nilai awal/teks awal yang tertulis pada sebuah *object*. Kadang-kadang nilai/teks awal tersebut dapat diganti dan kadang-kadang tidak, tergantung jenis *object*nya.

### 6. Alignment (perataan teks)

Menentukan apakah teks ditampilkan rata kanan, rata kiri atau rata tengah.



## Beberapa *object* Umum dalam Visual FoxPro



### 1. FORM

*Form* adalah *object* utama yang otomatis dibuat oleh *Visual FoxPro* pada waktu pengetikan perintah *MODIFY FORM*. *FORM* dapat mengandung semua *object* yang lain, kecuali *formset* yang lebih luas karena sebuah *formset* dapat mengandung beberapa *form*.

#### Property khusus:

*MaxButton*, *MinButton*, *ControlBox*: menentukan apakah tombol *maximize*, *minimize*, dan *control MenuBox* ditampilkan atau tidak.

*WindowState*: menentukan ukuran *form* tersebut waktu program dijalankan.



### 2. LABEL

Sesuai dengan namanya, *object* yang biasanya menampilkan teks tertentu seperti: judul, keterangan, dan lain-lain. Label tidak memiliki *data source*, tidak dapat diedit secara langsung, tidak dapat di-*click* atau ditab.



### 3. TEXT BOX

*Textbox* dipergunakan untuk mengisi input yang tidak dapat ditentukan sebelumnya. Jika *Value* tidak diberi nilai awal, maka isi *textbox* dianggap *character*, sedangkan jika diberi nilai awal berupa angka, maka angka dianggap *numeric*.

#### Property khusus:

- a. *ControlSource*: menentukan *field* dan *table* mana yang merupakan sumber data *textbox* yang bersangkutan.
- b. *InputMask*: menentukan karakter apa yang dapat diketikkan ke dalam *textbox* tersebut.
  - X → semua karakter dapat diisikan
  - 9 → angka dan tanda (+/-) dapat diisikan
  - # → angka, tanda, dan *blank* dapat diisikan
  - Y → membatasi pengisian hanya bisa Y dan N
  - \$ dan £\$ → menambahkan tanda *currency*
  - \* → tempat kosong akan diisi tanda bintang/asterik
  - . → menunjukkan jumlah desimal yang dikehendaki
  - , → membubuhkan pemisah ribuan
- c. *Format*: menentukan bagaimana karakter yang merupakan *value* dari *textbox* itu akan ditampilkan.
  - A → hanya menerima huruf (spasi & tanda baca tidak diterima)
  - D → menggunakan *format date* dengan *SET DATE*
  - E → mengedit data jenis *Date* sebagai *British*
  - K → menyeleksi *textbox* waktu kursor dipindahkan ke *textbox*
  - L → mengisi semua tempat kosong dengan nol
  - M → memberikan pilihan dalam *InputMask*
  - ! → mengubah huruf kecil menjadi huruf besar

- d. *PasswordChar*: menentukan karakter apa yang akan ditampilkan waktu user mengisi *password*.

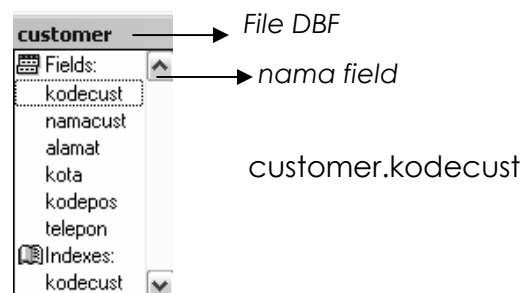


#### 4. EDIT BOX

Edit Box dipakai untuk mengisi/mengedit/menampilkan field memo atau field character yang sangat panjang.

##### Property khusus:

- a. *ControlSource*. Isikan nama table (file DBF) dan nama field yang terkait dengan edit box tersebut. Misalnya:



- b. *ReadOnly*: menentukan apakah user diperbolehkan mengubah teks yang ada di *text box*.  
 c. *Scrollbars*: apakah *textbox* akan diberi *scroll bar* atau tidak.



#### 5. COMMAND BUTTON

Merupakan *object* yang paling penting dan paling sering digunakan, sebab biasanya dipakai untuk memerintahkan kepada program untuk melakukan suatu proses.

**Event khusus:** *click*



#### 6. COMMAND GROUP

Kadang-kadang memberikan kemudahan, yaitu membuat sekelompok *command button* sekaligus dimana *procedure* untuk *click event-nya* dapat dibuat sekaligus pula pada *command button group*. Contohnya:

RELEASE THIS FORM (untuk keluar dari form tersebut)

**Event khusus:** *click*

##### Property khusus:

*Button count*, menyatakan berapa tombol yang terdapat di dalam *group*



#### 7. OPTION GROUP

*Option Button Group* merupakan sebuah *container* yang mengandung beberapa *Option Buttons*: *Object* ini biasanya untuk memaksa user agar hanya dapat memilih satu dari sejumlah pilihan.

##### Property khusus:

- a. *ButtonCount*: menentukan berapa pilihan di dalam *group*  
 b. *Value*: menentukan *option* mana yang telah terpilih (*option pertama*=1, *kedua*=2, dst)



## 8. CHECKBOX

Berbeda dengan *Option Group* biasanya sebuah program aplikasi menampilkan sekelompok *check box* di mana *user* dapat memilih salah satu atau beberapa dari pilihan yang diajukan. Namun demikian, masing-masing *checkbox* hanya dapat berstatus *.T.* atau *.F.* atau *NULL*. Apabila sebuah *checkbox* dikaitkan dengan sebuah *field logical*, maka jika sebuah record adalah *.T.* akan diberi tanda *X*, sedangkan jika *.F.* akan dikosongkan.

### Property khusus:

- a. *Value*: menyatakan status dari pada *checkbox*
- b. *Control Source*: keterkaitan *checkbox* dengan sebuah *field* tertentu.



## 9. COMBO BOX

*Combo box* merupakan kombinasi daripada sebuah *list box* dan *text box*. Ada dua jenis *combo box*: *Dropdown Combo* dan *Dropdown List*.

### Property khusus:

- a. *Control Source*: menentukan tabel (file *DBF*) yang merupakan sumber dari mana pilihan diperoleh.
- b. *RowSourceType*: menentukan jenis input yang diharapkan.
  - 0 - None
  - 1 - Value (pilihan ditulis langsung diproperti ini)
  - 2 - Table (pilihan diperoleh dari Table (file *DBF*))
  - 3 - SQL Statement (pilihan diperoleh dari rumus SQL query)
  - 4 - QPR (pilihan diperoleh dari file *QPR*)
  - 5 - Array (pilihan diperoleh dari variable array)
  - 6 - Fields (pilihan diperoleh dari fields tertentu)
- c. *RowSource*: menentukan secara spesifik sumber dari pilihan-pilihan yang ada. Hal ini harus sesuai dengan *RowSourceType*.
- d. *Style*: memilih antara *Dropdown Combo* dan *Dropdown List*.



## 10. LIST BOX

Mirip dengan *ComboBox*, sebuah *List Box* biasa dipakai untuk menampilkan sebuah daftar pilihan.

### Property Khusus:

- a. *ColumnCount*: menentukan jumlah kolom di dalam *List Box*.
- b. *ColumnWidths*: menentukan lebar masing-masing kolom di dalam *Listbox*. Contoh penulisan: 25,100,50 artinya kolom pertama lebarnya 25, kolom kedua 100, dan ketiga lebarnya 50.
- c. *ControlSource*: menunjukkan dimana nilai yang terkandung di dalam *ListBox* disimpan.
- d. *RowSource & RowSourceType*: lihat penjelasan *ComboBox*.

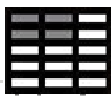


### 11. SPINNER

*Spinner* biasanya dipakai untuk membuat pilihan *numeric* atas sederetan nilai. Pengisian *spinner* dapat langsung diketikkan, dan dapat juga dengan meng-*click mouse* pada tanda panah di sebelah kanan *spinner*.

**Property khusus:**

- KeyboardHighValue&KeyboardLowValue*: menentukan nilai terbesar dan terkecil yang dapat diisi melalui *keyboard*.
- SpinnerHighValue&SpinnerLowValue*: menentukan nilai terbesar dan terkecil yang dapat diisi dengan meng-*click mouse*.
- Interval*: berapa nilai yang 'diloncat' setiap kali *user* menekan tombol *mouse*.



### 12. GRID

Seperti yang terlihat pada gambar diatas, sebuah *Grid* dapat mengandung beberapa *Columns* dan *Headers*. Masing-masing *grid*, *column* dan *header* merupakan *object* tersendiri.

**Property khusus:**

- ColoumnCount*: menentukan berapa banyak *column* yang terdapat di dalam *grid*. (Jika diisi -1, maka jumlah *column* otomatis sama banyaknya dengan jumlah *field* dari tabel (*file DBF*) yang terasosiasi dengan *Grid* tersebut).
- Recordsource & RecordSourceType*: hal ini sama dengan *Row Source* dan *RowSourceType* pada *combo box*.

CATATAN: untuk mengubah lebar setiap kolom. Letakkan *mouse cursor* pada batas diantara dua *header* sampai bentuk *cursor* berubah, lalu *click and drag*.

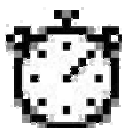


### 13. IMAGE

Dipakai untuk menampilkan sebuah gambar (*file .BMP*) pada *form*.

**Property khusus:**

*Picture*: disini disebutkan nama *file* gambar (.BMP) yang harus ditampilkan.



### 14. TIMER

*Timer* adalah suatu objek yang khas, karena pada saat *runtime* (program dijalankan), objek ini tidak akan terlihat. Yang terasa hanyalah pengaruhnya atas jalannya program. *Timer* akan melakukan sebuah *procedure* tertentu secara berulang-ulang setiap kali sejumlah waktu telah berlalu.

**Event khusus :** *Timer*. Ini adalah *event* yang harus dibuatkan *procedurennya*, dimana kita memberikan perintah-perintah kepada program, apa yang harus dilakukan *timer* setiap kali interval waktu itu telah lewat.

**Property khusus:**

*Interval*: lamanya objek *Timer* harus menunggu sebelum melakukan suatu proses. Satuan interval yang dipergunakan adalah *milisecond*. Misalnya, jika interval ditentukan 500, maka berarti *Timer* akan mengulang kegiatannya setiap setengah detik sekali.



**15. PAGEFRAME**

**Page frame** merupakan sebuah *container* yang mengandung beberapa objek lain, yaitu *page*.

**Property khusus:**

*Pagecount*: menentukan jumlah *pages* yang terdapat di dalam *Page frame*.

1. Latihan Membuat Form 1

Buatlah form dengan layout seperti berikut:

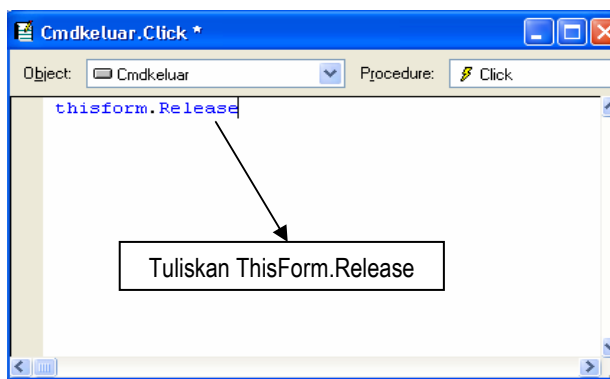


Gambar 4. Form Latihan Pertama

Ubahlah Properties Sebagai berikut:

Objek	Properties	Setting
Form1	Caption	Form Latihan 1
	Name	Frmlatihan1
Label1	Caption	Selamat Belajar Visual Foxpro
Command1	Caption	\<Keluar
	Name	cmdKeluar

Kemudian Click dua kali pada command keluar sehingga ditampilkan jendela code seperti berikut :



Gambar 5. Jendela Source Code

Untuk menjalankan progrm, click menu Program >DO atau Click icon (**Ctrl+E**). Kemudian pada saat dijalankan visual fox pro akan menampilkan konfirmasi sebagai berikut. **Click Yes** untuk menyimpannya.

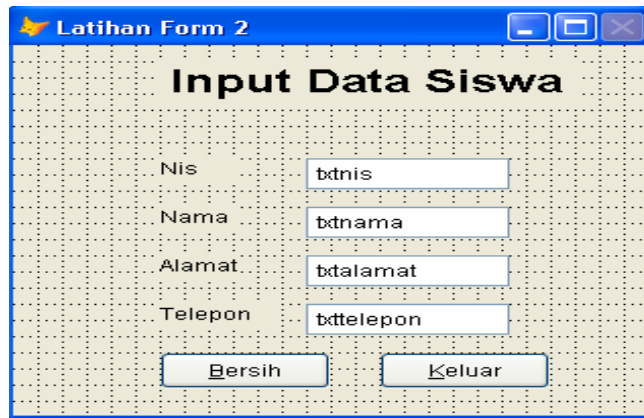


Gambar 6. Dialog Konfirmasi Penyimpanan Form

Click **Yes** untuk menyimpan pembuatan form dan menjalankan form setelah disimpan. Jika form belum pernah disimpan maka akan ditampilkan kotak dialog **Save** untuk menyimpan form tersebut, berilah nama form tersebut kemudian click Save.

2. Latihan Membuat Form 2

Buatlah form dengan layout seperti berikut:



Gambar 7. Form Latihan Kedua

Ubahlah Properties menjadi seperti berikut:

Objek	Properties	Setting
Form1	Caption	Form Latihan 2
	Name	Frmlatihan2
Labe11	Caption	Input Data Siswa
	Aligment	Center
	Fontbold	True
	FontSize	36
Labe12	Caption	Nis
Labe13	Caption	Nama
Labe14	Caption	Alamat
Labe15	Caption	Telepon
Text1	Name	Txtnis
Text2	Name	Txtnama
Text3	Name	Txtalamat
Text4	Name	Txttelepon
Command1	Caption	\<Bersih
	Name	Cmdbersih
Command2	Caption	\<Keluar
	Name	CmdKeluar

Lengkapi dengan kode seperti berikut sesuai dengan object dan procedure-nya :

```
Object : cmdBersih      Procedure : Click
```

```
ThisForm.txtnis.Value=""
ThisForm.txtnama.Value=""
ThisForm.txtalamat.Value=""
ThisForm.txttelepon.Value=""
ThisForm.txtnis.SetFocus()
```

```
Object : cmdKeluar     Procedure : Click
```

```
ThisForm.Release
```

**Operator Aritmatika**

Dalam pemrograman database menggunakan visual foxpro juga tidak terlepas dengan perhitungan-perhitungan aritmatika yang tentunya juga memerlukan operand dan operatornya. Pada umumnya operator yang digunakan untuk operasi aritmatika adalah sama dengan operator aritmatika pada bahasa pemrograman yang lainnya. Operator disini juga digunakan untuk melakukan proses perhitungan matematika yang meliputi perpangkatan, perkalian, pembagian, penjumlahan, pengurangan dan pengelompokan seperti tabel berikut :

Tabel Operator Aritmatika

Operator	Keterangan	Contoh
^	Pangkat	5^2=25
*	Kali	5*2=10
/	Bagi	4/2=2
+	Tambah	5+2=7
-	Kurang	5-2=3
( )	Pengelompokan	(3+2) * 4

3. Latihan Membuat Form 3

Buatlah form dengan layout seperti berikut:



Gambar 8. Form Latihan Ketiga

Ubahlah Properties menjadi seperti berikut:

Objek	Properties	Setting
Form1	Caption	Form Latihan 3
	Name	Frmlatihan3
Labe11	Caption	Pertambahan
	Aligment	Center
	Fontbold	True
	FontSize	14
Labe12	Caption	Nilai 1
Labe13	Caption	Nilai 2
Labe14	Caption	Hasil
Text1	Name	Txtnilai1
Text2	Name	Txtnilai2
Text3	Name	Txthasil
	Maxlength	10
	InputMask	99999999
	Enabled	True
Command1	Caption	\<Hitung
	Name	Cmdhitung
Command2	Caption	\<Keluar
	Name	CmdKeluar

Lengkapi dengan kode program yang sesuai dalam object dan procedure-nya seperti berikut :

```
Object : cmdHitung      Procedure : Click
```

```
Nilai1=VAL(ThisForm.txtnilai1.Value)
Nilai2=VAL(ThisForm.txtnilai2.Value)
hasil=nilai1+nilai2
ThisForm.txthasil.Value=hasil
```

```
Object : cmdKeluar      Procedure : Click
```

```
ThisForm.Release
```

4. Latihan Membuat Form 4

Buatlah form dengan layout seperti berikut:



Gambar 9. Form Latihan Keempat

Seperti cara yang sama sebelumnya, lengkapilah command button yang ada dengan kode program yang sesuai.



# FUNGSI LOGIKA



Agar program aplikasi yang dibangun menjadi semakin baik maka perlu menambah intruksi atau perintah tambahan ke dalam aplikasi tersebut melalui objek-objeknya. Untuk itu kita perlu mengetahui fungsi logika program yang diperlukan. Berikut adalah beberapa fungsi logika yang tersedia.

## A. Logika If – Endif

Logika If – Endif digunakan untuk mengambil suatu keputusan dari beberapa kondisi.

Format :

```

If [Ekspresi] Then
    Statement 1
Else
    Statement 2
Endif
  
```

## B. Logika Do Case – EndCase

Logika **Do Case – EndCase** sama seperti logika If – Endif, digunakan untuk mengambil suatu keputusan dari beberapa kondisi. Biasa digunakan jika kondisinya lebih banyak.

Format:

```

Do Case
    Case [Ekspresi 1]
        Statement 1
    Case [Ekspresi 2]
        .....
        .....
    Otherwise
        [Ekspresi N]
EndCase
  
```

1. Latihan Membuat Form 5

Buatlah form dengan layout seperti berikut:

Gambar 10. Form Latihan Kelima

Ubahlah Properties menjadi seperti berikut:

Objek	Properties	Setting
Form1	Caption	Form Latihan 5
	Name	Frmlatihan5
Labe11	Caption	Logika - If
	Aligment	Center
	Fontbold	True
	FontSize	14
Labe12	Caption	Nis
Labe13	Caption	Nilai
Labe14	Caption	Keterangan
Text1	Name	Txtnis
Text2	Name	Txtnilai
Text3	Name	Txtketerangan
Command1	Caption	\<Proses
	Name	Cmdproses
Command2	Caption	\<Bersih
	Name	CmdBersih
Command3	Caption	\<Keluar
	Name	CmdKeluar

Lengkapi dengan kode program seperti berikut :

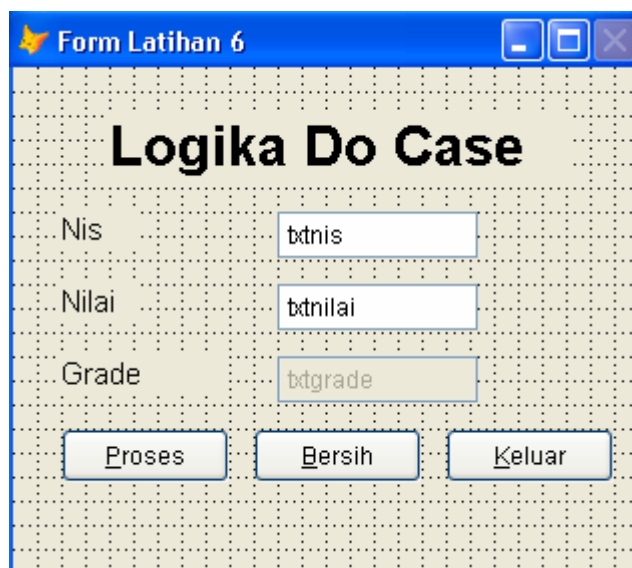
Obejct: CmdProses                      Procedure: Click

```
IF VAL(ThisForm.txtnilai.Value)>65
    ThisForm.txtketerangan.Value="Lulus"
ELSE
    ThisForm.txtketerangan.Value="Gagal"
ENDIF
```

Untuk kode program **bersih** dan **keluar** buatlah sendiri dengan cara yang sama seperti sebelumnya.

2. Latihan Membuat Form 6

Buatlah form dengan layout seperti berikut:



Gambar 11. Form Latihan Keenam

Ubahlah Properties seperti berikut:

Objek	Properties	Setting
Form1	Caption	Form Latihan 6
	Name	Frmlatihan6
Labe11	Caption	Logika Do Case
	Aligment	Center
	Fontbold	True
	FontSize	14
Labe12	Caption	Nis
Labe13	Caption	Nilai
Labe14	Caption	Grade
Text1	Name	Txtnis
Text2	Name	Txtnilai
Text3	Name	Txtgrade
Command1	Caption	\<Proses
	Name	Cmdproses
Command2	Caption	\<Bersih
	Name	CmdBersih
Command3	Caption	\<Keluar
	Name	CmdKeluar

Lengkapi dengan kode program seperti berikut :

Obejct: CmdProses	Procedure: Click
-------------------	------------------

```
DO case
    CASE VAL(ThisForm.txtnilai.Value) > 89
        ThisForm.txtgrade.Value = "A"
    CASE VAL(ThisForm.txtnilai.Value) > 79
        ThisForm.txtgrade.Value = "B"
    CASE VAL(ThisForm.txtnilai.Value) > 69
        ThisForm.txtgrade.Value = "C"
    CASE VAL(ThisForm.txtnilai.Value) > 59
        ThisForm.txtgrade.Value = "D"
    OTHERWISE
        ThisForm.txtgrade.Value = "E"
ENDCASE
```

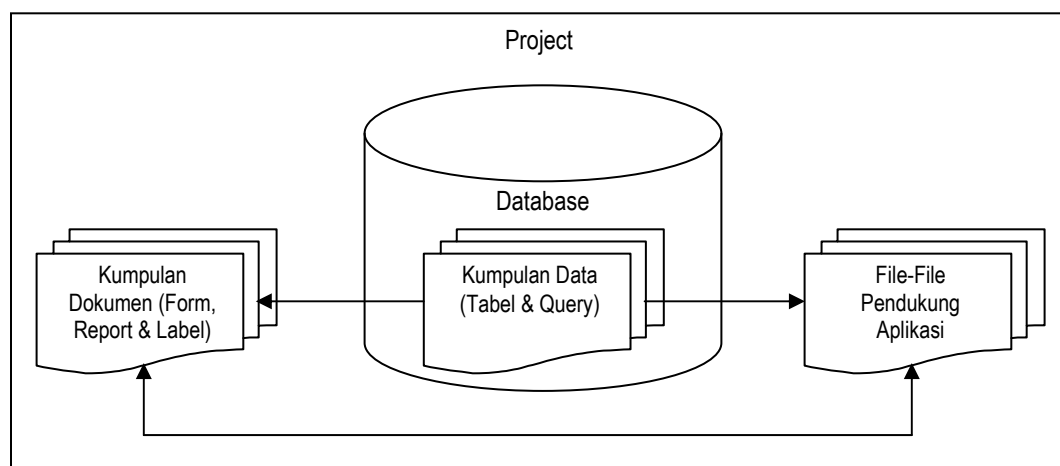
Untuk kode program **bersih** dan **keluar** buatlah sendiri.

# BEKERJA DENGAN PROJECT MANAGER



Pembuatan program aplikasi umumnya dibuat dalam satu file project yaitu file yang digunakan untuk mengorganisir dan mengatur file-file yang diperlukan dalam sebuah program aplikasi. Hal ini dikarenakan dalam sebuah program aplikasi tidak hanya terdiri dari satu file saja melainkan terdiri dari banyak file yang berkaitan dalam aplikasi yang dibuat, untuk itu diperlukan tempat sebagai pengelola file-file yang diperlukan. Sedangkan database diperlukan guna mengorganisir dan mengatur file-file data dalam bentuk tabel-tabel dan relasi antar table yang bersesuaian.

Gambaran secara umum dalam sebuah program aplikasi dapat dilihat dalam ilustrasi gambar berikut :



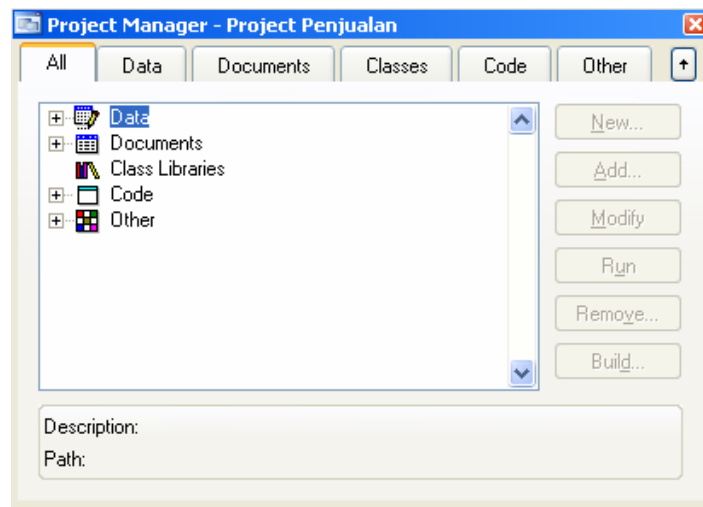
Gambar 12. Ilustrasi File dalam Program Aplikasi

Project Manager pada Visual FoxPro merupakan kumpulan dari file Data, Documents, Classes, Code dan Other yang keseluruhan objek disimpan dalam file dengan ekstensi **.pjx**.

Sebelum membuat project manager buat dahulu sebuah folder baru pad folder anda masing-masing dengan nama folder **penjualan**. Jika sudah selesai, berilah perintah **SET DEFAULT TO PENJUALAN** melalui command windows untuk menetapkan folder default operasi agar data dan proses yang dilakukan tidak salah tempat.

## 1. Membuat Project Manager

- Click Menu File>New>Pilih Project>Click New File
- Kemudian pada kotak dialog Create di bagian *Enter project file* isikan nama file project misalnya **Penjualan**, simpan dengan click tombol **Save** sehingga kotak dialog project manager ditampilkan seperti berikut :



Gambar 13. Jendela Project Manager



Pada tab **All** berisi semua kelompok file yang ada, tab **Data**, **Documents**, **Classes**, **Code** dan **Other** berisi kelompok file seperti yang telah digambarkan sebelumnya. Sedangkan tombol-tombol operasi yang ada diantaranya **New** untuk membuat data atau file baru sesuai pilihan kelompok file yang terpilih dalam tab All atau masing-masing tab yang ada, **Add** untuk menambahkan data atau file yang sudah ada yang mungkin telah dibuat secara terpisah sebelumnya, **Modify** untuk memperbaiki atau mengedit data atau kelompok file yang dipilih, **Run** untuk mengaktifkan atau menjalankan (running) file yang ada dalam kelompok file, **Remove** untuk memutuskan atau membuang kelompok file yang terpilih dari project manager, jika pilihan remove dipilih maka akan ditampilkan jendela konfirmasi untuk menghapus file atau hanya memutuskan file dari project manager, **Build** untuk mengcompile project yang sedang dibuat menjadi file executable (file exe) sehingga dapat dijalankan tanpa memanggil program Visual FoxPro terlebih dahulu.

## 2. Membuat Database dan Tabel Pada Project Manager

- a. Pada Project Manager Click **Tab Data**
- b. Kemudian Pilih **Database**, Click **New**> Click **New Database**. Simpan dengan nama **dbPenjualan** sehingga ditampilkan jendela database dan toolbox database designer seperti berikut



Gambar 14. Jendela Database Designer

- c. Membuat Tabel Pada **dbPenjualan**. Click icon  pada toolbox database designer atau tutup jendela database designer kemudian Click tanda + pada  , Kemudian Pilih **Tabels** pada **dbPenjualan** Click New>Click New Table. Kemudian simpan dengan nama tabel **Barang**. Pada table designer isi fields namanya seperti berikut:

#### Struktur Tabel Barang

Name	Type	Width	Keterangan
Kdbarang	Character	6	Index (Primary),(Ascending)
Nmbarang	Character	25	
Harga	Numeric	10	

Kemudian buatlah **Tabel Customer**, **Tabel Transaksi**, **Tabel Faktur**, dan **Tabel Tmp**. Berikut struktur masing-masing tabel tersebut :

#### Struktur Tabel Customer

Name	Type	Width	Keterangan
Kdcustomer	Character	6	Index (Primary),(Ascending)
Nmcustomer	Character	25	
Alamat	Character	35	
Notelp	Character	15	

Untuk **Tabel Barang** dan **Tabel Customer** masukan datanya sebanyak 10 (sepuluh) masing-masing tabel.

#### Struktur Tabel Faktur

Name	Type	Width	Keterangan
Nofaktur	Character	6	Index (Primary),(Ascending)
Tanggal	Date	8	
Kdcustomer	Character	6	Index (Regular),(Ascending)

#### Struktur Tabel Transaksi

Name	Type	Width	Keterangan
Nofaktur	Character	6	Index (Primary),(Ascending)
Kdbarang	Character	6	Index (Regular),(Ascending)
Qty	Numeric	6	
Jumlah	Numeric	10	

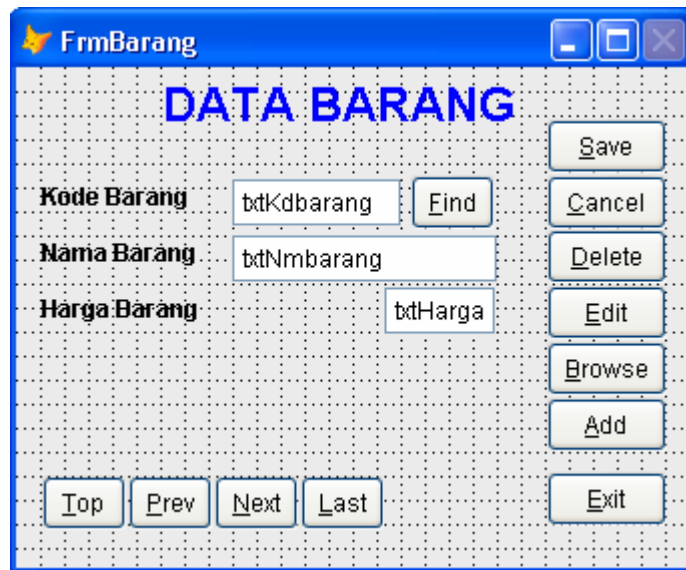
#### Struktur Tabel Tmp

Name	Type	Width	Keterangan
Kdbarang	Character	6	Tidak diberikan index
Nmbarang	Character	25	
Harga	Numeric	10	
Qty	Numeric	6	
Jumlah	Numeric	10	

**3. Membuat Form Pada Project Manager**

**Membuat Form Barang**

- a. pada Project Manager click **Tab Document**
- b. Kemudian pilih **Forms** click New>click New Form.
- c. Buatlah form barang sebagai berikut:



Gambar 15. Form Latihan Ketujuh

Ubahlah Properties menjadi seperti berikut:

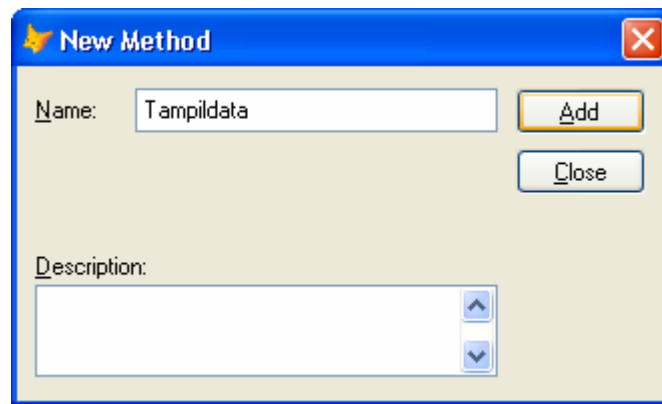
Objek	Properties	Setting	Objek	Properties	Setting
Form1	Caption	Data Barang	Command5	Caption	\<Edit
	Name	Frmbarang		Name	Cmdedit
Labe11	Caption	Data Barang	Command6	Caption	\<Browse
	Aligment	Center		Name	CmdBrowse
	FontSize	14	Command7	Caption	\<Add
Labe12	Caption	Kode Barang		Name	CmdAdd
Labe13	Caption	Nama Barang	Command8	Caption	\<Exit
Labe14	Caption	Harga		Name	CmdExit
Text1	Name	TxtKdBarang	Command9	Caption	\<Top
Text2	Name	TxtNmBarang		Name	CmdTop
Text3	Name	TxtHarga	Command10	Caption	\<Prev
Command1	Caption	\<Find		Name	CmdPrev
	Name	CmdFind	Command11	Caption	\<Next
Command2	Caption	\<Save		Name	CmdNext
	Name	CmdSave	Command12	Caption	\<Last
Command3	Caption	\<Cancel		Name	CmdLast
	Name	CmdCancel			
Command4	Caption	\<Delete			
	Name	CmdDelete			

**Membuat Procedure Baru**

Prosedur dalam program dapat ditambah sesuai dengan keperluan. Manfaatnya adalah untuk memudahkan penulisan terhadap program yang akan digunakan secara berulang-ulang

pada sebuah form dan pemberian nama pada procedure baru tidak boleh mengandung spasi. Berikut langkah membuat procedure baru :

Click **Menu Form > New Method**, kemudian tuliskan nama procedure yang akan dibuat pada kotak isian Name kemudian click tombol **Add** (Lihat gambar berikut ini).



Gambar 16. Dialog New Method

Buatlah procedure baru sebanyak 5 (lima) buah procedur masing-masing dengan nama : **AturObject, KondisiAwal, Kosong, NonAktif dan TampilData.**

#### Membuat Kode Program pada Form Barang

Lengkapilah form Barang dengan kode program seperti berikut (Sesuaikan Nama Object dan Nama Procedurnya

Obejct: Form	Procedure: Load
--------------	-----------------

```
SET DELETED ON
SELECT 1
USE Barang
SET ORDER TO Tag KdBarang
```

Obejct: Form	Procedure: Init
--------------	-----------------

```
PUBLIC Isi
Isi = .T.
ThisForm.TxtKdBarang.Enabled = .T.
ThisForm.txtNmBarang.Enabled = .F.
ThisForm.txtHarga.Enabled = .F.

ThisForm.txtKdBarang.Value = ""
ThisForm.txtNmBarang.Value = ""
ThisForm.txtHarga.Value = ""

ThisForm.cmdSave.Enabled= .F.
ThisForm.cmdFind.Visible= .T.
ThisForm.cmdDelete.Enabled= .T.
ThisForm.cmdEdit.Enabled= .F.
ThisForm.cmdAdd.Enabled= .F.

ThisForm.txtKdBarang.SetFocus()
```



Obejct: Frmbarang	Procedure: AturObject
-------------------	-----------------------

```
PARAMETERS Kondisi
ThisForm.txtKdbarang.Enabled = .Not. Kondisi
ThisForm.txtNmBarang.Enabled = Kondisi
ThisForm.txtHarga.Enabled = Kondisi
```

Obejct: Frmbarang	Procedure: Kosong
-------------------	-------------------

```
ThisForm.txtKdbarang.Value = ""
ThisForm.txtNmbarang.Value = ""
ThisForm.txtHarga.Value = ""
```

Obejct: Frmbarang	Procedure: KondisiAwal
-------------------	------------------------

```
ThisForm.AturObject(.F.)
ThisForm.Kosong
ThisForm.cmdFind.Visible = .T.
ThisForm.cmdSave.Enabled = .F.
ThisForm.cmdFind.Enabled = .T.
ThisForm.txtKdBarang.SetFocus()
```

Obejct: Frmbarang	Procedure: Tampildata
-------------------	-----------------------

```
SELECT 1
ThisForm.txtKdBarang.Value = KdBarang
ThisForm.txtNmBarang.Value = NmBarang
ThisForm.txtHarga.Value = Harga
```

Obejct: Frmbarang	Procedure: Nonaktif
-------------------	---------------------

```
ThisForm.txtKdbarang.Enabled = .F.
ThisForm.txtNmBarang.Enabled = .F.
ThisForm.txtHarga.Enabled = .F.
```

Obejct: CmdTop	Procedure: Click
----------------	------------------

```
ThisForm.txtKdBarang.Enabled = .F.
ThisForm.cmdDelete.Enabled = .T.
ThisForm.cmdEdit.Enabled = .T.
ThisForm.cmdAdd.Enabled = .T.
SELECT 1
GO TOP
ThisForm.TampilData
ThisForm.Refresh
```

Obejct:CmdLast	Procedure: Click
----------------	------------------

```
ThisForm.txtKdbarang.Enabled= .F.  
ThisForm.cmddelete.Enabled= .T.  
ThisForm.cmdedit.Enabled= .T.  
ThisForm.cmdadd.Enabled= .T.  
SELECT 1  
GO BOTTOM  
ThisForm.TampilData  
ThisForm.Refresh
```

Obejct:CmdPrev	Procedure: Click
----------------	------------------

```
ThisForm.txtKdBarang.Enabled = .F.  
ThisForm.cmdDelete.Enabled = .T.  
ThisForm.cmdEdit.Enabled = .T.  
ThisForm.cmdAdd.Enabled = .T.  
SKIP -1  
IF BOF() Then  
    GO TOP  
    MESSAGEBOX("Sudah di Awal Record",48,"Perhatian")  
ENDIF  
ThisForm.TampilData  
ThisForm.Refresh
```

Obejct:CmdNext	Procedure: Click
----------------	------------------

```
ThisForm.txtKdBarang.Enabled = .F.  
ThisForm.cmdDelete.Enabled = .T.  
ThisForm.cmdEdit.Enabled = .T.  
ThisForm.cmdAdd.Enabled = .T.  
SKIP  
IF EOF() Then  
    GO BOTTOM  
    MESSAGEBOX("Sudah di Akhir Record",48,"Perhatian")  
ENDIF  
ThisForm.TampilData  
ThisForm.Refresh
```

Obejct:Cmdfind	Procedure: Click
----------------	------------------

```
IF EMPTY(ThisForm.txtKdbarang.Value) Then
    MESSAGEBOX("Ups..., Kode Barang Harus Diisi!",48,"Validasi")
    ThisForm.txtKdBarang.SetFocus()
    RETURN
ELSE
    SELECT 1
    SEEK(TRIM(ThisForm.txtKdbarang.Value))
    IF FOUND() Then
        ThisForm.Nonaktif
        ThisForm.TampilData
        ThisForm.cmdFind.Enabled = .T.
    ENDIF
ENDIF
ThisForm.txtNmBarang.Value=""
ThisForm.txtHarga.Value=""
ThisForm.cmdSave.Enabled= .F.
ThisForm.cmdFind.Visible= .T.
ThisForm.txtKdBarang.SetFocus
ThisForm.AturObject(.T.)
ThisForm.cmdFind.Visible = .F.
ThisForm.cmdSave.Enabled = .T.
ThisForm.cmdCancel.Enabled = .T.
ThisForm.cmdDelete.Enabled = .F.
ThisForm.cmdAdd.Enabled = .F.
ThisForm.txtNmBarang.SetFocus()
```

Obejct:CmdSave	Procedure: Click
----------------	------------------

```
IF EMPTY(ThisForm.txtNmBarang.Value) Then
    MESSAGEBOX("Upss..., Nama Barang Harus Diisi!",48,"Validasi")
    ThisForm.txtNmBarang.SetFocus()
    RETURN
ENDIF
IF EMPTY(ThisForm.txtHarga.Value) Then
    MESSAGEBOX("Upss..., Harga Barang Harus Diisi!",48,"Validasi")
    ThisForm.txtHrgbarang.SetFocus()
    RETURN
ENDIF
```

```

IF Isi = .T. Then
    APPEND BLANK
    Replace KdBarang WITH ThisForm.txtKdBarang.Text
    Replace NmBarang WITH ThisForm.txtNmBarang.Text
    Replace Harga WITH VAL(ThisForm.txtHarga.Text)
ELSE
    Replace KdBarang WITH ThisForm.txtKdBarang.Value
    Replace NmBarang WITH ThisForm.txtNmBarang.Value
    Replace Harga WITH ThisForm.txtHarga.Value
ENDIF
ThisForm.KondisiAwal
ThisForm.Refresh

```

Obejct:CmdCancel	Procedure: Click
------------------	------------------

```

ThisForm.KondisiAwal

```

Obejct:CmdDelete	Procedure: Click
------------------	------------------

```

IF RECCOUNT() = 0 Then
    RETURN
ELSE
    Pesan = MESSAGEBOX("Benarkah Data Tersebut Akan Dihapus?",36,"Konfirmasi")
    IF Pesan=6 Then
        DELETE
        PACK
    ENDIF
ENDIF
ThisForm.TampilData
ThisForm.Refresh

```

Obejct:CmdEdit	Procedure: Click
----------------	------------------

```

Isi = .F.
ThisForm.txtKdBarang.Enabled = .F.
ThisForm.txtNmBarang.Enabled = .T.
ThisForm.txtHarga.Enabled = .T.
ThisForm.cmdAdd.Enabled = .F.
ThisForm.cmdFind.Enabled = .F.
ThisForm.cmdSave.Enabled = .T.
ThisForm.txtNmBarang.SetFocus
ThisForm.Refresh

```

Obejct:CmdAdd Procedure: Click

```
isi=.T.
ThisForm.txtKdBarang.Enabled= .T.
ThisForm.txtNmBarang.Enabled= .F.
ThisForm.txtHarga.Enabled= .F.
ThisForm.kosong

ThisForm.cmdfind.Enabled= .T.
ThisForm.txtKdBarang.SetFocus
ThisForm.Refresh
```

Obejct:Cmdexit Procedure: Click

```
nTanya = MESSAGEBOX("Apakah Anda Yakin Akan Keluar?",4+48,"Perhatian!")
IF nTanya = 6 Then
    ThisForm.Release
ENDIF
```

Obejct:CmdBrowse Procedure: Click

```
Browse
```

### Membuat Form Customer

- Pada Project Manager Click **Tab Document**
- Kemudian pilih **Forms** click New>click New Form.
- Buatlah form **Customer** dengan layout seperti berikut:

The screenshot shows a Windows application window titled "Input Data Customer". The window has a blue title bar with standard minimize, maximize, and close buttons. The main area has a light gray grid background. At the top, the text "DATA CUSTOMER" is displayed in large blue letters. Below this, there are four rows of data entry fields:
 

- "Kode Customer" with a text box labeled "btkdcust" and a "Find" button to its right.
- "Nama Customer" with a text box labeled "btxnmcust".
- "Alamat Customer" with a text box labeled "btxalmt".
- "Telpon" with a text box labeled "btxtelp".

 To the right of these fields is a vertical column of buttons: "Save", "Cancel", "Delete", "Edit", "Browse", "Add", and "Exit". At the bottom of the form, there are four buttons: "Top", "Prev", "Next", and "Last".

Gambar 17. Form Latihan Kedelapan

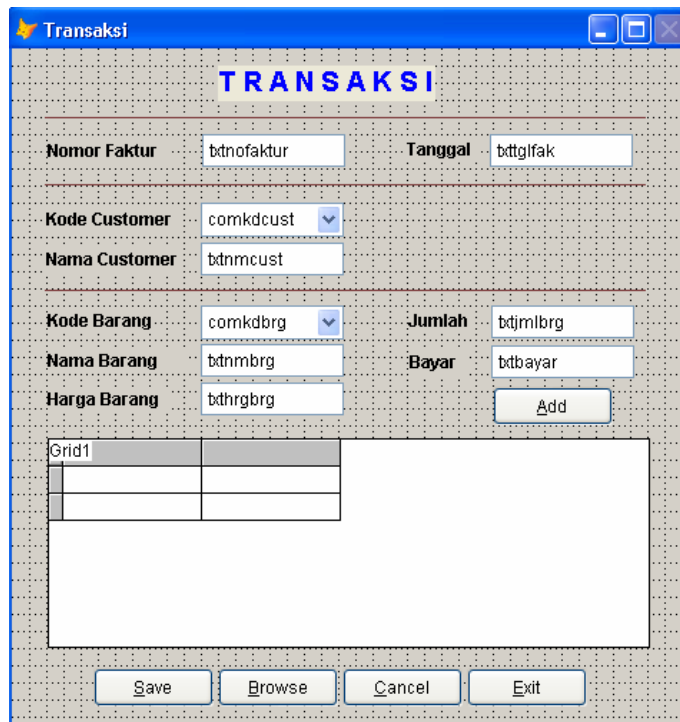
Ubahlah Properties menjadi seperti berikut:

Objek	Properties	Setting	Objek	Properties	Setting
Form1	Caption	Data Cusatomer	Command4	Caption	\<Delete
	Name	FrmCustomer		Command5	Name
Labe11	Caption	Data Customer	Command6		Caption
	Aligment	Center		Command7	Name
	FontSize	14	Command8		Caption
Labe12	Caption	Kode Customer		Command9	Name
Labe13	Caption	Nama Customer	Command10		Caption
Labe14	Caption	Alamat		Command11	Name
Labe15	Caption	Telepon	Command12		Caption
Text1	Name	Txtkdcust		Command11	Name
Text2	Name	Txtnmcust	Command10		Caption
Text3	Name	Txtalmt		Command11	Name
Text4	Name	txttelp	Command12		Caption
Command1	Caption	\<Find		Command11	Name
	Name	CmdFind	Command12		Caption
Command2	Caption	\<Save		Command11	Name
	Name	CmdSave	Command12		Caption
Command3	Caption	\<Cancel		Command12	Name
	Name	CmdCancel			

Selanjutnya lengkapi dengan **kode program** dan **procedure baru** sama seperti pada **form data barang**.

**Membuat Form Transaksi**

- a. Pada Project Manager click **Tab Document**
- b. Kemudian pilih **Forms** click New>click New Form.
- c. Buatlah form **Transaksi** dengan layout seperti berikut:



Gambar 18. Form Latihan Kesembilan

Ubahlah Properties Sebagai berikut:

Objek	Properties	Setting	Objek	Properties	Setting
Form1	Caption	Transaksi	Text1	Name	Txtnofaktur
	Name	FrmTransaksi	Text2	Name	Txttglfak
Label1	Caption	Transaksi	Text3	Name	Txtnmcust
	Aligment	Center	Text4	Name	txtnmbrg
	FontSize	14	Text5	Name	Txthrgbrg
Label2	Caption	No. Faktur	Text6	Name	Txtjmlbrg
Label3	Caption	Tanggal	Text7	Name	Txtbayar
Label4	Caption	Kode Customer	Grid1	RecordSource	Tmp
Label5	Caption	Nama Customer		RecordSourceType	2-Alias
Label6	Caption	Kode Barang	Command1	Caption	\<Add
Label7	Caption	Nama Barang		Name	CmdAdd
Label8	Caption	Jumlah	Command2	Caption	\<Save
Label9	Caption	Bayar		Name	CmdSave
Combo1	Name	comkdcust	Command3	Caption	\<Browse
	ControlSource	Customer		Name	CmdBrowse
	RowSource	Customer.kdcustomer	Command4	Caption	\<Cancel
	RowSourceType	2-Alias		Name	CmdCancel
Combo2	Name	Comkdbrg	Command5	Caption	\<Exit
	ControlSource	Barang		Name	CmdExit
	RowSource	Barang.Kdbarang			
	RowSourceType	2-Alias			

Untuk **Combo1** dan **Combo2** pada propertisnya diubah pada **ControlSource** untuk menghubungkan dengan tabel dan **Rowsource** dihubungkan dengan filed pada tabel dan **RowSourceType** dengan **2 - Alias** typenya. Sedang dengan **Grid1** diubah pada propertiesnya **RecordSource** dengan sebuah nama tabel, **RowSourceType** dengan **2 - Alias** typenya.

**Melengkapi Form Trasaksi dengan Kode Program**

Sebelumnya buatlah procedure baru dengan nama **Bersih** dan **KondisiAwal**.

Obejct: Form	Procedure: Load
--------------	-----------------

```

SELECT 1
USE Barang
SET ORDER TO tag KdBarang

SELECT 2
USE customer
SET ORDER TO tag kdcustomer

SELECT 3
USE Faktur
SET ORDER TO tag nofaktur

SELECT 4
USE Tmp.dbf
    
```

```
SELECT 5
USE Transaksi
SET RELATION TO KdBarang INTO Barang
SET RELATION TO nofaktur INTO Faktur Additive
```

```
SELECT 3
SET RELATION TO KdCustomer INTO Customer Additive
```

```
SELECT 4
DELETE ALL
PACK
```

Obejct: Form	Procedure: Bersih
--------------	-------------------

```
ThisForm.txtnofaktur.Value=""
ThisForm.comkdcust.Value=""
ThisForm.txtnmcust.Value=""
ThisForm.comkdbrg.Value=""
ThisForm.txtnmbrg.Value=""
ThisForm.txthrgbrg.Value=""
ThisForm.txtjmlbrg.Value=""
ThisForm.txtbayar.Value=""
```

Obejct: Form	Procedure: Kondisiawal
--------------	------------------------

```
ThisForm.Bersih
ThisForm.txtnofaktur.SetFocus()
ThisForm.txtnofaktur.Value=""
ThisForm.comkdcust.Enabled= .F.
ThisForm.txtnmcust.Enabled= .F.
ThisForm.comkdbrg.Enabled= .F.
ThisForm.txtnmbrg.Enabled= .F.
ThisForm.txthrgbrg.Enabled= .F.
ThisForm.txtjmlbrg.Enabled= .F.
ThisForm.txtbayar.Enabled= .F.
ThisForm.cmdsave.Enabled= .F.
ThisForm.cmdcancel.Enabled= .F.
```

Obejct: Form	Procedure: Activate
--------------	---------------------

```
ThisForm.KondisiAwal
mtgl=DATE()
ThisForm.txttgifak.Value = mtgl
```



Obejct:Comkdcust	Procedure: Click
------------------	------------------

```
SELECT 2
SEEK ThisForm.comkdcust.Value
IF FOUND() Then
    ThisForm.txtnmcust.Value=nmcustomer
    ThisForm.comkdbrg.Enabled= .T.
    ThisForm.comkdbrg.SetFocus
ELSE
    MESSAGEBOX("Kode tersebut tidak ada",48,"Informasi")
ENDIF
```

Obejct:Comkdbrg	Procedure: Click
-----------------	------------------

```
SELECT 1
SEEK ThisForm.comkdbrg.Value
IF FOUND() Then
    ThisForm.txtnmbrg.Value=NmBarang
    ThisForm.txthrgbrg.Value=hrgbarang
    ThisForm.txtjmlbrg.Enabled= .T.
    ThisForm.txtjmlbrg.SetFocus()
ELSE
    MESSAGEBOX("Kode tersebut tidak ada ",48,"Informasi")
ENDIF
```

Obejct:CmdAdd	Procedure: Click
---------------	------------------

```
SELECT 4
APPEND BLANK
Replace KdBarang WITH ThisForm.comkdbrg.Value
Replace NmBarang WITH ThisForm.txtnmbrg.Value
Replace Harga WITH ThisForm.txthrgbrg.Value
Replace qty WITH VAL(ThisForm.txtjmlbrg.Value)
Replace bayar WITH ThisForm.txtbayar.Value

ThisForm.Refresh
ThisForm.comkdbrg.Value=""
ThisForm.comkdbrg.SetFocus
ThisForm.txtnmbrg.Value=""
ThisForm.txthrgbrg.Value=""
ThisForm.txtjmlbrg.Value=""
ThisForm.txtbayar.Value=""
ThisForm.cmdadd.Enabled= .F.
ThisForm.cmdsave.Enabled= .T.
```

Obejct:txtnofaktur	Procedure: Lostfocus
--------------------	----------------------

```
SELECT 3
IF EMPTY(ThisForm.txtnofaktur.Value)
    RETURN
ELSE
    IF SEEK(TRIM(ThisForm.txtnofaktur.Value))
        MESSAGEBOX("Nomor tersebut sudah ada",48,"Pesan")
        ThisForm.txtnofaktur.Value=""
        ThisForm.txtnofaktur.SetFocus()
    ELSE
        ThisForm.comkdcust.Enabled= .T.
        ThisForm.comkdcust.SetFocus
    ENDIF
ENDIF
```

Obejct:Cmddsave	Procedure: Click
-----------------	------------------

```
SELECT 3
APPEND BLANK
Replace nofaktur WITH ThisForm.txtnofaktur.Value
Replace tanggal WITH ThisForm.txttglfak.Value
Replace kdcustomer WITH ThisForm.comkdcust.Value
SELECT 4
SCAN
    SELECT 5
    APPEND BLANK
    Replace nofaktur WITH ThisForm.txtnofaktur.Value
    Replace KdBarang WITH tmp.KdBarang
    Replace qty WITH tmp.qty
    Replace bayar WITH tmp.bayar
    SELECT 4
ENDSCAN
SELECT 4
DELETE ALL
PACK
ThisForm.Grid1.RecordSource="tmp"
ThisForm.Grid1.Column1.width=100
ThisForm.Grid1.Column2.width=100
ThisForm.Grid1.Column3.width=100
```

```
ThisForm.Grid1.column4.width=30
ThisForm.Grid1.column5.width=100
ThisForm.txtnofaktur.Value=""
ThisForm.comkdcust.Value=""
ThisForm.comkdcust.Enabled= .F.
ThisForm.comkdbrg.Value=""
ThisForm.comkdbrg.Enabled= .F.
ThisForm.txtnmcust.Value=""
ThisForm.txtnmbrg.Value=""
ThisForm.txthrgbrg.Value=""
ThisForm.txtjmlbrg.Value=""
ThisForm.txtbayar.Value=""
ThisForm.txtjmlbrg.Enabled= .F.
ThisForm.txtnofaktur.SetFocus()
```

Obejct:CmdBrowse	Procedure: Click
------------------	------------------

```
SELECT 5
BROWSE
```

Obejct:Cmdcancel	Procedure: Click
------------------	------------------

```
ThisForm.Bersih
SELECT 4
DELETE ALL
PACK
ThisForm.grid1.RecordSource="tmp"
ThisForm.grid1.column1.width=80
ThisForm.grid1.column2.width=150
ThisForm.grid1.column3.width=50
ThisForm.grid1.Column4.width=40
ThisForm.grid1.column5.width=80
ThisForm.txtnofaktur.Value=""
ThisForm.comkdcust.Value=""
ThisForm.txtnmcust.Value=""
ThisForm.txtnofaktur.SetFocus()
```

Obejct:CmdExit	Procedure: Click
----------------	------------------

```
ThisForm.Release()
```

## BEKERJA DENGAN REPORT



*Report* atau Laporan adalah kegiatan menampilkan dan meringkas data dalam format cetakan baik ke printer maupun cetakan ke layar. Dua komponen utama dari laporan adalah data dan rancangan. Sumber data untuk laporan bisa tabel, *query*, atau pun *view*.

Dalam *report designer*, kita bisa memasukkan obyek-obyek berupa *textbox*, *label*, *line*, *rectangle*, *rounded rectangle*, dan *picture/ole bound control*. Obyek-obyek tersebut bisa kita masukkan ke dalam beberapa bagian *report* yang berbeda. Pada *report* yang tidak memiliki data *grouping*, *report* dibagi ke dalam tiga bagian, *pageheader*, *detail*, dan *page footer*. *Page header* dan *page footer* hanya akan ditampilkan satu kali (maksimal) dalam satu halaman. Bagian *detail* akan dicetak berulang-ulang selama datanya masih ada (misalnya dari tabel). Pada *report* yang memiliki data *grouping*, ada dua bagian tambahan, yaitu *group header* dan *group footer*.

### Variable

Di dalam *report*, kita dapat membuat satu atau beberapa buah variabel yang perlu ditampilkan dalam *report* sebagai hasil dari perhitungan tertentu, baik perhitungan fisik tabel (misalnya *COUNT*) maupun perhitungan matematis (misalnya *SUM*).

Pembuatan variabel dalam *report* dapat dilakukan dengan membuka dialog *variables* (pilih menu *reports, variables*). Berikut beberapa komponen dalam dialog *variables*:

Komponen	Keterangan
<i>Variables</i>	Tuliskan nama-nama variabel yang akan kita buat dalam <i>list</i> ini.
<i>Value to Store</i>	Tuliskan variabel, nama <i>field</i> , atau apapun yang akan dimasukkan ke dalam variabel.
<i>Calculate</i>	Beberapa perlakuan (penghitungan) yang dilakukan pada <i>value to store</i> sebelum disimpan pada variabel. Pilihannya adalah: <ul style="list-style-type: none"> <li>• <i>Nothing</i></li> <li>• <i>Count</i></li> <li>• <i>Sum</i></li> <li>• <i>Average</i></li> <li>• <i>Lowest</i></li> <li>• <i>Highest</i></li> <li>• <i>Std. Deviation</i></li> <li>• <i>Variation</i></li> </ul>
<i>Initial Value</i>	Nilai awal yang disimpan pada variabel.
<i>Release after report</i>	Pilihan untuk me-release variabel setelah <i>report</i> selesai ditampilkan
<i>Reset at</i>	Pilihan waktu pengulangan proses <i>Calculate</i> . Ketika pengulangan dilakukan, nilai variabel akan berubah menjadi <i>initial value</i> .

### Data Grouping

Seringkali kita perlu untuk mengelompokkan beberapa data dalam tabel yang ditampilkan dalam *report* berdasarkan kategori tertentu, pengelompokkan ini dalam *Visual FoxPro* dapat kita lakukan dengan menggunakan fasilitas *Data Grouping*.

Untuk membuat *data grouping*, pilih menu *report, data Grouping*. Pada *dialog box data grouping*, ada beberapa komponen yang perlu diisi.

Komponen	Keterangan
<i>Group Expression</i>	Dalam kotak ini, kita isikan dasar pengelompokkan.
<i>Group Properties</i>	Pilihan-pilihan preferensi perlakuan terhadap grup.

## A. Membuat Report

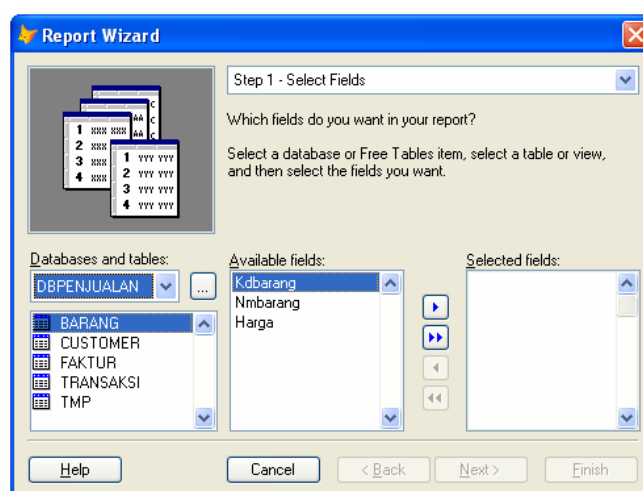
Dalam pembuatan report dapat bersumber dari satu tabel atau dari beberapa tabel bergantung keperluan informasi yang akan diberikan, langkah pembuatannya juga tidak banyak berbeda. Selain itu pembuatan report dapat dikelompokkan atau tidak dikelompokkan. Berikut adalah langkah-langkah dalam pembuatan report.

### 1. Report Tidak Dikelompokkan

#### a. Report Dari Satu Tabel

Untuk membuat report dari satu tabel dapat dilakukan dengan langkah-langkah berikut:

- 1) Pada project manager click tab **Documents**
- 2) Click kelompok **Reports**
- 3) Click tombol **New**
- 4) Pilih **Report Wizard** pada kotak dialog New Report yang ditampilkan
- 5) Pilih **Report wizard** lagi pada kotak dialog wizard selection kemudian click **OK** sehingga akan ditampilkan kotak dialog seperti berikut :



Gambar 19. Jendela Report Wizard

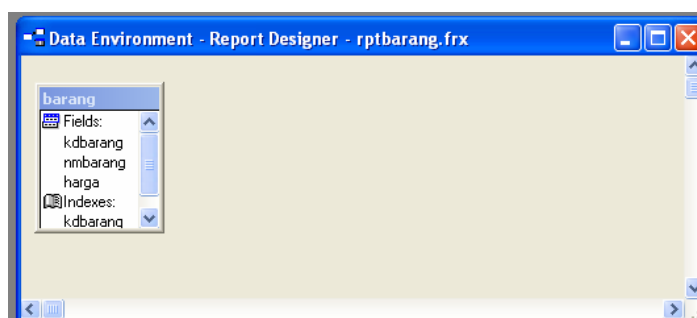
- 6) Pada kotak dropdown **Databases and tables**, tentukan/pilih database yang akan dibuatkan report kemudian pilih nama tabel sumber datanya pada daftar nama-nama tabel dibawah kotak dropdown
- 7) Pada bagian **Available fields**, pilih nama field yang akan ditampilkan dalam report kemudian klik icon ► atau ►► jika ingin mengambil semua field yang akan ditampilkan dalam report. Selanjutnya click tombol **Next**
- 8) Pada Step 2 (*Group Record*) click tombol **Next** (karena laporan tidak dikelompokan)
- 9) Selanjutnya tentukan style laporannya dan jika sudah, click tombol **Next**
- 10) Pada Step 4 (*Define Report Layout*) pilihlah orientasi pencetakan **Portrait** atau **Landscape** selanjutnya click **Next**
- 11) Langkah selanjutnya menentukan pengurutan data yang dilaporkan dengan memilih nama field pada bagian **Available field or index tag**. Pemilihan field ini digunakan untuk dasar pengurutan datanya baik secara ascending atau descending. Jika sudah click tombol **Next** lagi
- 12) Dikotak dialog selanjutnya tuliskan Judul Reportnya pada bagian kotak isian **Type a title for you report**. Pilih tombol radio **Save and print report**
- 13) Click tombol **Finish** kemudian jika diminta untuk memasukan nama file maka tuliskan nama file report yang dibuat, selanjutnya click **Save**.

#### b. Report Dari Beberapa Tabel

Untuk membuat report dari beberapa tabel langkahnya sama seperti membuat report dari satu tabel namun tabel pertama yang dipilih adalah tabel transaksinya, sedangkan tabel-tabel lain sebagai tabel referensi ditambahkan setelah masuk ke dalam desain reportnya.

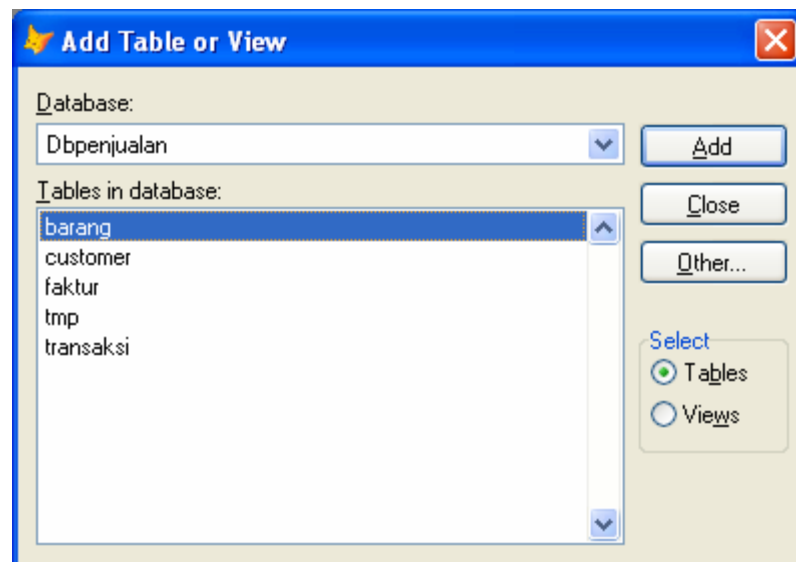
Langkah menambahkan tabel lain ke dalam report yang dibuat dari beberapa tabel adalah :

- 1) Buka desain report yang sudah dibuat
- 2) Click kanan pada area yang kosong kemudian pilih **Data Environment** sehingga ditampilkan kotak dialog data environment seperti gambar berikut :



Gambar 20. Jendela Data Environment

- 3) Click kanan pada area kosong jendela data environment kemudian pilih **Add...** sehingga dialog berikut ditampilkan.



Gambar 21. Jendela Add Table or View

- 4) Pilih/click nama tabel yang ada dalam daftar **Tables in database**, kemudian click tombol **Add**. Jika sudah selesai menambahkan tabel, clicklah tombol **Close**.
- 5) Selanjutnya relasikan tabel-tabel yang ada dengan cara drag pada nama field di tabel satu ke nama field di tabel lain sehingga terbentuk sebuah garis penghubung tabel.
- 6) Drag nama-nama field yang diperlukan ke area kerja report designer

## 2. Report Dikelompokkan

### a. Report Dari Satu Tabel

Untuk membuat report yang dikelompokkan dan dari satu tabel dapat dilakukan dengan langkah-langkah sama seperti tidak pembuatan report yang tidak dikelompokkan kecuali pada langkah 8 berikut langkah pembuatan lengkapnya :

- 1) Pada project manager click tab **Documents**
- 2) Click kelompok **Reports**
- 3) Click tombol **New**
- 4) Pilih **Report Wizard** pada kotak dialog New Report yang ditampilkan
- 5) Pilih **Report wizard** lagi pada kotak dialog wizard selection kemudian click **OK** sehingga akan ditampilkan kotak dialog seperti pada Gambar 19.
- 6) Pada kotak dropdown **Databases and tables**, tentukan/pilih database yang akan dibuatkan report kemudian pilih nama tabel sumber datanya pada daftar nama-nama tabel dibawah kotak dropdown
- 7) Pada bagian **Available fields**, pilih nama field yang akan ditampilkan dalam report kemudian click icon ► atau ►► jika ingin mengambil semua field yang akan ditampilkan dalam report. Selanjutnya click tombol **Next**
- 8) Pada Step 2 (*Group Record*) click kotak dropdown 1, 2 dan atau 3 kemudian pilih nama field yang digunakan sebagai pengelompokan. Jika diperlukan click tombol **Summary Options...** untuk menentukan field-field mana yang perlu diberikan penjumlahan atau

penentuan data statistiknya seperti average, count, min dan max. Kemudian click tombol **Next**

- 9) Selanjutnya tentukan style laporannya dan jika sudah, click tombol **Next**
- 10) Pada Step 4 (*Define Report Layout*) pilihlah orientasi pencetakan **Portrait** atau **Landscape** selanjutnya click **Next**
- 11) Langkah selanjutnya menentukan pengurutan data yang dilaporkan dengan memilih nama field pada bagian **Available field or index tag**. Pemilihan field ini digunakan untuk dasar pengurutan datanya baik secara ascending atau descending. Jika sudah click tombol **Next** lagi
- 12) Dikotak dialog selanjutnya tuliskan Judul Reportnya pada bagian kotak isian **Type a title for you report**. Pilih tombol radio **Save and print report**
- 13) Click tombol **Finish** kemudian jika diminta untuk memasukan nama file maka tuliskan nama file report yang dibuat, selanjutnya click **Save**.

**b. Report Dari Beberapa Tabel**

Untuk report dari beberapa tabel dengan pengelompokan juga dibuat dengan langkah yang sama seperti pembuatan report dari satu tabel dan dikombinasikan dengan report pengelompokan dari satu tabel.

**B. Latihan Membuat Report**

**Latihan Membuat Report dengan satu tabel**

Buatlah report daftar harga barang dari tabel Barang yang sudah dibuat sebelumnya. Desainlah layout laporan seperti dalam layout berikut :

**DAFTAR HARGA  
BARANG TOKO SERBA ADA "ANDA"**

No.	Kode Barang	Nama Barang	Harga Barang
99	xxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	999,999,999
99	xxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	999,999,999
99	xxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	999,999,999
99	xxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	999,999,999
99	xxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	999,999,999
99	xxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	999,999,999
99	xxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	999,999,999
99	xxxxxx	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	999,999,999
dst...	dst...	dst...	dst...

Dibuat Oleh  
Nama Anda Masing-Masing



**Latihan Membuat Report dengan beberapa tabel**

Untuk membuat report dengan beberapa tabel didasarkan pada tabel penjualan, sedangkan tabel yang lain sebagai tabel referensi. Buatlah report tersebut dengan layout berikut :

DAFTAR TRANSAKSI  
PENJUALAN  
PERIODE BULAN MARET 2011 – AGUSTUS 2011

<b>No. Faktur : 999999</b>			<b>Customer : xxxxxxxxxxxxxxxxxxxx</b>	
Kode Barang	Nama Barang	Jumlah	Harga Satuan	Jumlah Harga
xxxxxx	xxxxxxxxxxxxxxxxxxxx	999	999,999	999,999,999
xxxxxx	xxxxxxxxxxxxxxxxxxxx	999	999,999	999,999,999
xxxxxx	xxxxxxxxxxxxxxxxxxxx	999	999,999	999,999,999
xxxxxx	xxxxxxxxxxxxxxxxxxxx	999	999,999	999,999,999
xxxxxx	xxxxxxxxxxxxxxxxxxxx	999	999,999	999,999,999
<i>Total Harga</i>				999,999,999
<b>No. Faktur : 999999</b>			<b>Customer : xxxxxxxxxxxxxxxxxxxx</b>	
Kode Barang	Nama Barang	Jumlah	Harga Satuan	Jumlah Harga
xxxxxx	xxxxxxxxxxxxxxxxxxxx	999	999,999	999,999,999
xxxxxx	xxxxxxxxxxxxxxxxxxxx	999	999,999	999,999,999
xxxxxx	xxxxxxxxxxxxxxxxxxxx	999	999,999	999,999,999
<i>Total Harga</i>				999,999,999
<b>No. Faktur : 999999</b>			<b>Customer : xxxxxxxxxxxxxxxxxxxx</b>	
Kode Barang	Nama Barang	Jumlah	Harga Satuan	Jumlah Harga
xxxxxx	xxxxxxxxxxxxxxxxxxxx	999	999,999	999,999,999
xxxxxx	xxxxxxxxxxxxxxxxxxxx	999	999,999	999,999,999
dst...	dst...	dst...	dst...	dst...
<i>Total Harga</i>				999,999,999
<b>Total Harga Keseluruhan</b>				<b>999,999,999</b>

Dibuat Oleh  
Nama Anda Masing-Masing

# TAHAP AKHIR PEMBUATAN APLIKASI



Tahap akhir dalam pembuatan program aplikasi adalah tahapan penyelesaian agar program aplikasi dapat berjalan lebih maksimal dan dikontrol oleh sebuah perintah yang selalu dibaca pertama kali pada saat program dijalankan. Tahap akhir ini meliputi beberapa langkah, diantaranya adalah :

## A. Menetapkan Program Utama

Program utama dalam Visual FoxPro adalah program yang pertama kali dijalankan pada saat aplikasi dipanggil/dieksekusi. Program utama bisa ditetapkan menurut keinginan pembuat aplikasi, mungkin objek Form, objek Menu atau objek Code.

Umumnya objek yang dijadikan program utama adalah objek Code karena pada objek code tersebut dapat dituliskan/diberikan perintah-perintah pengaturan variabel lingkungan (*Setting Environment*). Agar sebuah objek dikenali sebagai program utama (*Main Program*) maka kita perlu mengubahnya menjadi program utama, ini dilakukan dengan cara berikut :

- Pada jendela project, pilihlah program yang akan dijadikan program utama
- Click kanan mouse pada item objek yang diinginkan
- Pada menu popup yang ditampilkan pilihlah **Set Main** sehingga tulisan nama item objek berubah menjadi berwarna hitam tebal

## B. Membuat Objek Code

Jika program utama ditetapkan berasal dari sebuah objek code maka terlebih dahulu harus dibuatkan objek dengan disertai perintah-perintah yang diperlukan dalam sebuah program aplikasi.

Untuk membuat objek code caranya hamper sama dengan membuat objek form atau yang objek lainnya dengan langkah-langkah berikut :

- Click objek **Code** pada jendela project
- Click item **Program**
- Click tombol perintah **New...** sehingga ditampilkan jendela kode program namun tidak memiliki bagian nama objek dan bagian nama procedure seperti dalam code program sebuah form
- Tuliskan semua perintah yang diperlukan sebagai pengaturan atau setting variabel environment
- Simpan dan berilah nama objek code tersebut dengan nama yang sesuai keinginan Anda. (File code program akan disimpan dengan ekstensi file **.PRG**)
- Jika objek code program akan dijadikan sebagai program utama (*Main Program*) lakukanlah langkah untuk membuatnya sebagai main program dengan cara yang sudah disampaikan sebelumnya (*dalam bagian A modul ini*).

Contoh perintah-perintah yang sering digunakan dalam objek code program yang digunakan sebagai main program diantaranya adalah :

Set Talk Off        && Untuk menonaktifkan echo perintah dalam layar/objek form  
Set Date Italian    && Untuk mengubah format tanggal seperti Negara Italia  
Set Century On     && Untuk mengaktifkan tampilan empat angka pd angka tahun  
Set Deleted On     && Untuk mengaktifkan perintah penghapusan record  
\_Screen.Visible = .F.        && Untuk menonaktifkan tampilan layer utama FoxPro  
Do Form frmUtama                && Untuk menjalankan form utama aplikasi  
Read Events        && Untuk menahan eksekusi yang ada sampai akhir program

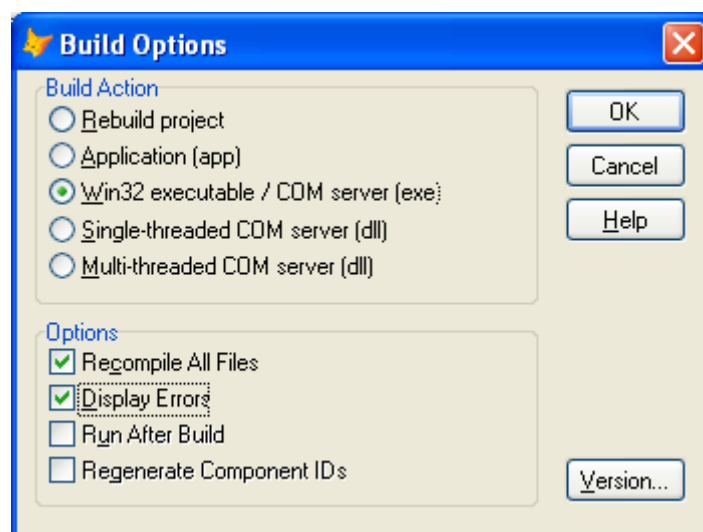
Contoh tersebut masih dapat dimodifikasi lagi atau ditambah perintah-perintah lainnya yang diperlukan dalam sebuah program aplikasi yang dibuat.

### C. Mengkompilasi Program

Agar program aplikasi yang dibuat dapat dijalankan langsung tanpa memanggil program Microsoft Visual FoxPro maka perlu di-*compile* (dikompilasi) terlebih dahulu.

Proses kompilasi akan menghasilkan sebuah file executable (**.EXE**) yaitu file yang dapat dijalankan langsung tanpa memanggil program Visual FoxPro. Proses kompilasi program dapat dilakukan dengan cara :

- Click tombol perintah **Build** pada jendela project sehingga ditampilkan kotak dialog Build Option seperti gambar berikut :



Gambar 22. Dialog Build Options

- Pada bagian Build Action pilihlah **Win32 executable / COM server (exe)** untuk membuat file hasil menjadi file executable (exe)

- Pada bagian Options pilihlah **Recompile All Files** untuk mengkompilasi semua file yang diperlukan dalam program aplikasi, dan **Display Errors** untuk menampilkan pesan kesalahan yang mungkin terjadi saat kompilasi sehingga kita mudah mencari kesalahan tersebut untuk dibetulkan
- Click tombol perintah **OK** sehingga ditampilkan kotak dialog **Save As** untuk menyimpan file hasil kompilasinya
- Tuliskan nama file hasil kompilasinya pada kotak dialog Save As tersebut (Nama file ini yang nantinya akan dijadikan sebagai nama file executablenya)
- Click tombol perintah **Save** untuk memulai proses kompilasi program dan tunggu sampai proses kompilasi selesai yang ditandai dengan pesan **Build executable complete** pada baris status
- Jika tidak terjadi kesalahan maka program berhasil dikompilasi dan jika ada kesalahan maka akan ditampilkan pesan kesalahannya pada jendela tersendiri dan harus diperbaiki terlebih dahulu kesalahan tersebut sebelum dikompilasi ulang.

Jika diperlukan lakukan pengujian untuk menjalankan file exe tersebut tanpa mengaktifkan program visual foxpro (tutup dulu program visual foxpro kemudian jalankan file exe yang terbentuk melalui window explorer)

## MEMBUAT FILE DISTRIBUSI (SETUP PROGRAM)



Jika kompilasi program menghasilkan file exe yang dapat dijalankan tanpa harus memanggil program Microsoft Visual FoxPro terlebih dahulu maka pembuatan file distribusi akan menghasilkan file **SETUP.EXE** yang digunakan untuk menginstal program aplikasi yang sudah dibuat ke Komputer lain yang tidak memiliki program Microsoft Visual FoxPro.

Sebab itulah file distribusi atau file setup program diperlukan untuk melakukan instalasi program aplikasi yang dibuat ke komputer lain sehingga program aplikasi dapat dijalankan tanpa harus memiliki program Microsoft Visual FoxPro.

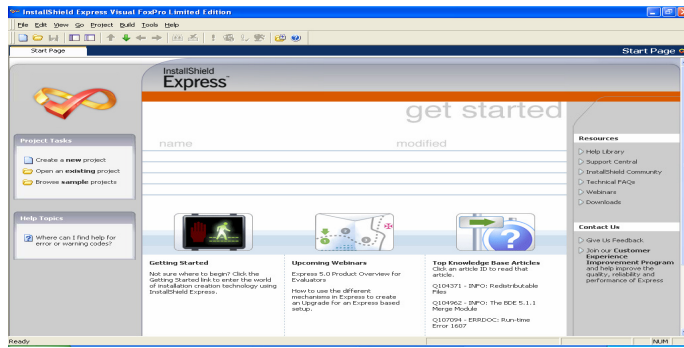
Pembuatan file distribusi ini memerlukan bantuan dari software **InstallShield Express for Visual FoxPro**. Jadi sebelumnya cek terlebih dahulu apakah program tersebut sudah terinstall dalam computer Anda dan jika belum ada tentunya harus diinstall terlebih dahulu software tersebut.

Jika sudah ada file distribusi bisa dibuat dengan langkah-langkah berikut :

- Buatlah folder baru dengan nama SUMBER (atau yang lainnya), kemudian copylah semua file yang diperlukan dalam aplikasi ke folder sumber tersebut. File-file yang diperlukan biasanya adalah file Database, file Tabel, file Index, file Report, file Executable dan file-file penunjang lainnya seperti file gambar, file icon dan sebagainya. Sedangkan file-file code program, menu utama dan file form tidak usah diikutkan karena sudah ada dalam file executable-nya.
- Buatlah folder baru dengan nama HASIL (atau yang lainnya) yang digunakan untuk menampung file hasil proses pembuatan file distribusi
- Jalankan program InstallShield Express for Visual FoxPro sehingga ditampilkan *splash screen* dan layar utama program seperti gambar berikut :

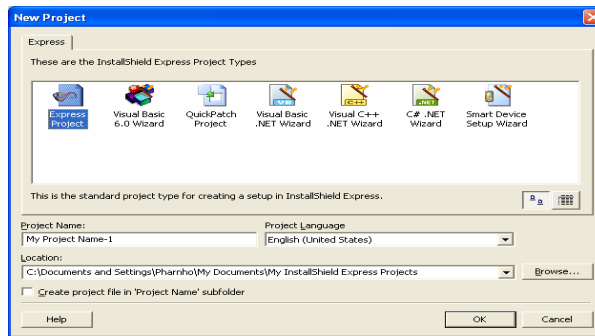


Gambar 23. Splash Screen InstallShield



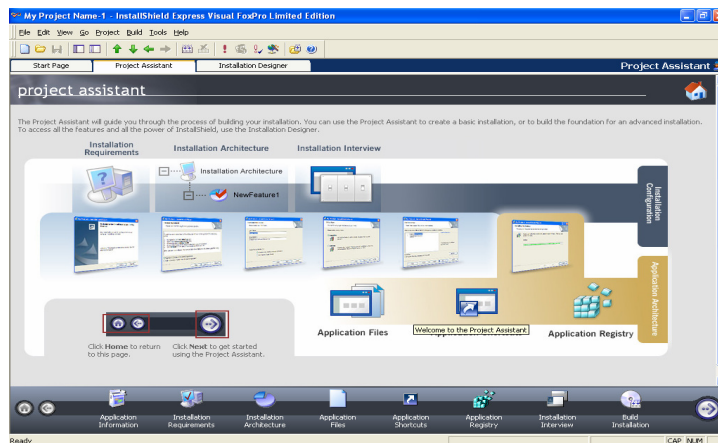
Gambar 24. Layar Utama Install Shield

- Dilayar utama pada bagian **Project Task** pilihlah **Create a new project** untuk membuat project installshield yang baru sehingga ditampilkan kotak dialog seperti gambar 23 (jika sudah pernah membuat project install shield, boleh memilih **Open an existing project** untuk menggunakannya kembali)



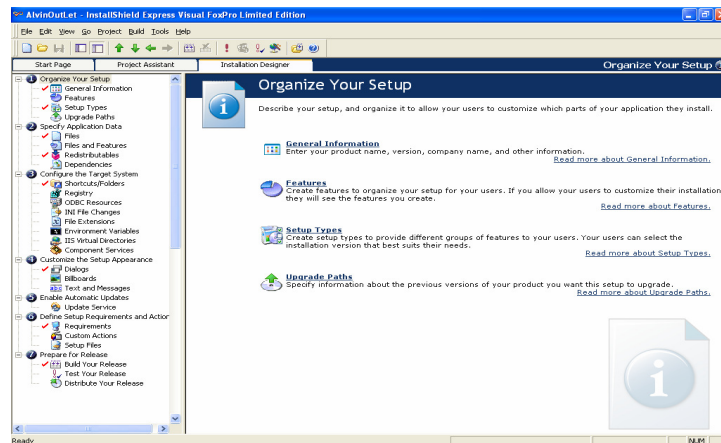
Gambar 25. Dialog New Project

- Pada bagian **Project Name** tuliskan Nama Project dari aplikasi yang Anda buat
- Pada bagian **Location** tuliskan nama folder hasil yang nantinya akan digunakan untuk menyimpan file hasil proses pembuatan file setup (dalam contoh ini nama foldernya adalah HASIL), click tombol **OK** sehingga layar wizard berikutnya ditampilkan seperti pada gambar berikut :



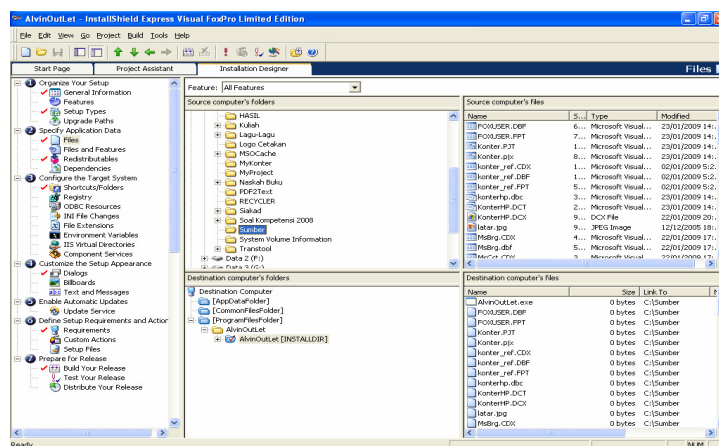
Gambar 26. Wizard Project Assistant

- Pilihlah tab **Installation Designer** sehingga layar wizard berubah menjadi seperti gambar berikut :



Gambar 27. Wizard pada tab Installation Designer

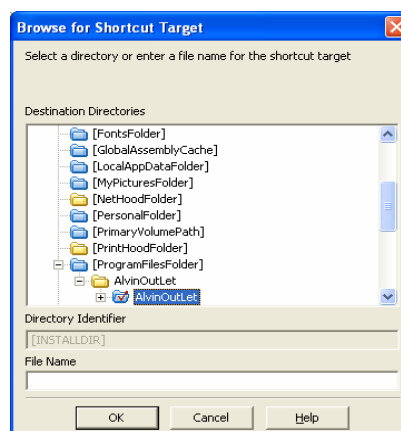
- Pada bagian **1. Organize Your Setup** pilihlah :
  - **General Information**, ubahlah data yang ada pada bagian :
    - Product Name* : untuk memberi nama aplikasi yang dibuat
    - Product Version* : untuk memberi nomor versi dari aplikasi yang dibuat (*optional*)
    - INSTALLDIR* : untuk menentukan folder yang digunakan sebagai tempat menyimpan file hasil setup aplikasinya
    - DATABASEDIR* : untuk menentukan folder database dan tabel-tabelnya jika ingin dibuat terpisah dengan folder programnya (*optional*)
    - Title* : untuk memberi judul program instalasi yang ditampilkan pada title bar saat melakukan instalasi program aplikasi
    - Author* : untuk menuliskan informasi pembuat program aplikasinya
  - **Setup Type**, aktifkan hanya pilihan **Typical** sedang Minimal dan Custom di nonaktifkan
- Pada bagian **2. Specify Application Data** pilihlah :
  - **File**, untuk menentukan file-file apa saja yang akan disertakan dalam program distribusi yang dibuat Lihat Gambar 28 berikut :



Gambar 28. Layar Pemilihan File yang Diperlukan

Cari dan click nama folder SUMBER pada bagian jendela *Source computer's folders*. Pilih dan blocklah file-file yang diperlukan pada bagian jendela *Source computer's files*. Click dan drag file yang sudah diblock ke nama folder installdir di bagian jendela *Destinations computer's folders*. Jika sudah perhatikan daftar nama-nama file yang ada pada bagian jendela *Destination computer's files* apakah semua file yang diperlukan dalam program aplikasi sudah masuk, jika ada yang belum ambil kembali dengan cara yang sama.

- **Redistributables**, untuk menyertakan file-file pendukung dan run time visual FoxPro. Pada bagian jendela *Installshield Objects/Merge Module* aktifkan atau centang semua pilihan yang diawali dengan kata-kata *Microsoft Visual FoxPro 9*.
- Pada bagian **3. Configure the Target System** pilihlah :
  - Shortcuts/Folder** untuk membuat shortcut program aplikasi pada start up menu dan desktop.
  - Untuk membuat shortcut menu pada Start – All Programs, click kanan pada *Program Menus* kemudian pilih *New Shortcut to Preexisting file* sehingga ditampilkan kotak dialog browse berikut :



Gambar 29. Dialog Browse for Shorcut Target

Pada bagian *File Name*, tuliskan nama file EXE dari program aplikasi yang sudah dibuat. Click tombol **OK** untuk mengakhiri pemilihan file untuk shortcut, kemudian gantilah nama *newshortcut* yang muncul dengan nama shortcut program aplikasi Anda.

- Untuk membuat shortcut menu pada layer Desktop, click kanan pada bagian *Desktop* kemudian lakukan hal yang sama seperti membuat shortcut pada bagian program menu.
- Pada bagian **4. Customize the Setup Appearance** pilihlah *Dialogs*. Pada jendela dialog aktifkan/clicklah elemen dialog yang diperlukan, misalnya *Instal Welcome*, *License Agreement*, *Customer Information*, *Destination folder*, *Ready to Install*, *Setup Progress* dan *Setup Complete Success*.
- Pada bagian **5. Enable Automatic Updates** boleh dilewati.
- Pada bagian **6. Define Setup Requirements and Action** pilihlah *Requirements*. Pada bagian *System Hardware Requirements* ubahlah nilai *Screen Resolution* sesuai dengan resolusi monitor



yang Anda gunakan pada saat mendesain program aplikasi. Ubah juga nilai *Color Depth* sesuai dengan kedalaman warna yang Anda gunakan pada saat mendesain program aplikasi. Pada bagian **OS Version** Jika diperlukan tentukan sistem operasi yang bias digunakan untuk menjalankan program aplikasi Anda.

- Pada bagian **7. Prepare For Release** pilihlah *Build Your Release* kemudian clicklah *SingleImage*. Jika diperlukan ubahlah nilai item *Generate Autorun.inf File* menjadi Yes. Click kanan *SingleImage* kemudian pilih *Build* untuk memulai proses pembuatan file distribusi (file setup) dan tunggu sampai proses selesai (*lihat jalannya proses pada bagian output yang ada di bawah layar*)
- Jika sudah selesai carilah file *setup.exe* hasil dari proses pembuatan tersebut pada folder *E:\HASIL>NamaAplikasi\Express\SingleImage\DiskImages\DISK1*. Perhatikan ada dua file dalam folder tersebut. Copylah file tersebut ke CD kemudian gunakan CD tersebut untuk menginstall program aplikasi Anda pada komputer yang lainnya.

## Daftar Referensi

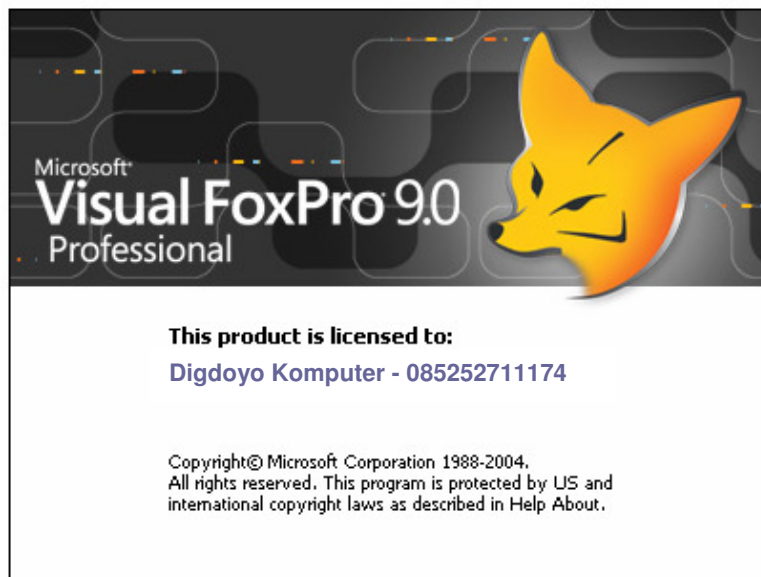
1. Tim Penerbit Andi, *Menggunakan Visual Foxpro 7.0*, Penerbit Andi, Yogyakarta 2002
2. Martin, Ir. Inge, *36 Jam Belajar Komputer Visual Foxpro 6.0*, PT. Elex Media Komputindo, Jakarta, 2001
3. Microsoft Visual FoxPro Help
4. InstallShield Help Library



# MODUL PRAKTIKUM

# Microsoft Visual FoxPro 9.0

( UNTUK KALANGAN SENDIRI )



Oleh  
**Suparno**

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
**(STMIK) PALANGKARAYA**  
2011

# KATA PENGANTAR



Microsoft Visual FoxPro 9.0 adalah bahasa pemrograman yang digunakan untuk membuat sebuah aplikasi yang berhubungan dengan pengolahan database. Kemampuannya dalam membuat aplikasi berbasis database dengan cepat dan mudah menjadikan bahasa pemrograman ini banyak diminati.

Modul Praktikum ini dimaksudkan untuk memberikan panduan dalam praktik membuat sebuah aplikasi untuk menerapkan perintah-perintah yang telah dipelajari dalam software Microsoft Visual FoxPro agar dalam pembuatan program aplikasi dapat dilakukan dengan efisien, efektif dan memiliki nilai user friendly sehingga orang lain yang menggunakan program aplikasi hasil buatan sendiri merasa lebih mudah dan dapat membantu pekerjaan yang ada secara optimal

Modul Praktikum ini disusun dengan tujuan untuk memberikan bantuan atau literatur tambahan bagi pembaca khususnya bagi Mahasiswa STMIK Palangkaraya yang menempuh matakuliah Pemrograman Database agar lebih mudah dalam berlatih membuat aplikasi dan dapat mengembangkannya sendiri sesuai dengan keperluan.

Tentunya modul praktikum ini masih banyak menyimpan kekurangan-kekurangan, karenanya saran dan kritik yang konstruktif dari pembaca atau pemakai sangat diharapkan untuk kelengkapan isinya agar selalu sesuai dengan perkembangan teknologi dalam bidang komputer khususnya dalam kelompok pemrograman database. Kritik dan saran dapat juga disampaikan pada penulis melalui email [endustong@yahoo.com](mailto:endustong@yahoo.com) atau melalui buku tamu pada [www.digdoyo3m.co.cc](http://www.digdoyo3m.co.cc) Akhirnya semoga buku ini dapat memberikan manfaat yang sesuai harapan kita semua.

Palangkaraya, Januari 2011

Penulis

# DAFTAR ISI

Kata Pengantar .....	i
Daftar Isi .....	ii
Pemrograman Visual FoxPro .....	1
Pointer .....	5
Pencarian Data .....	6
Bekerja dengan Form .....	7
Membuat Form Baru .....	7
Beberapa Property Umum pada Form .....	8
Beberapa Object Umum dalam Visual FoxPro .....	9
Latihan Membuat Form .....	13
Operator Aritmatika .....	15
Fungsi Logika .....	17
Bekerja dengan Project Manager .....	20
Membuat Project Manager .....	20
Membuat Database dan Tabel pada Project Manager .....	21
Membuat Form pada Project Manager .....	23
Membuat Procedure Baru .....	23
Membuat Kode Program .....	24
Bekerja dengan Report .....	36
Variabel .....	36
Data Grouping .....	37
Membuat Report .....	37
Report Tidak Dikelompokan .....	37
Report Dikelompokan .....	39
Latihan Membuat Report .....	40
Tahap Akhir Pembuatan Aplikasi .....	42
Menetapkan Program Utama .....	42
Membuat Object Code .....	42
Mengkompilasi Program .....	43
Membuat File Distribusi (Setup Program) .....	45
Daftar Referensi .....	49