

# Modul 1

## Dasar – Dasar Visual Foxpro

### 1.1. Pendahuluan

Microsoft Visual Foxpro merupakan perangkat lunak yang orientasinya adalah saplikasi bisnis, khususnya yang memakai database (pengolahan data) yang datanya cukup banyak.

Microsoft Visual Foxpro merupakan salah satu sistem manajemen database (*database management system*, DBMS) berorientasi obyek yang memungkinkan pembuatan aplikasi database yang tangguh dengan cepat dan mudah.

Kemudahan pembuatan aplikasi menggunakan Visual Foxpro antara lain karena pemrograman Visual Foxpro secara procedural dan sekaligus berorientasi obyek (*object oriented programming*, OOP) yang berjalan seiring dan saling terkait. Pemrograman procedural yang dikerjakan bersama-sama dengan pemrograman berorientasi obyek dan dilengkapi desain interaktif menjadikan Visual Foxpro banyak digunakan oleh para programmer Windows saat ini.

Secara garis besar pembuatan aplikasi database meliputi proses-proses sebagai berikut :

- Merencanakan atau merancang aplikasi
- Membuat Database (tabel, view, relasi tabel).
- Membuat kelas (event, properti, method).
- Menyediakan akses kegunaan (form, menu, toolbar).
- Menyediakan akses informasi (query, report, grafik).
- Testing dan debugging.

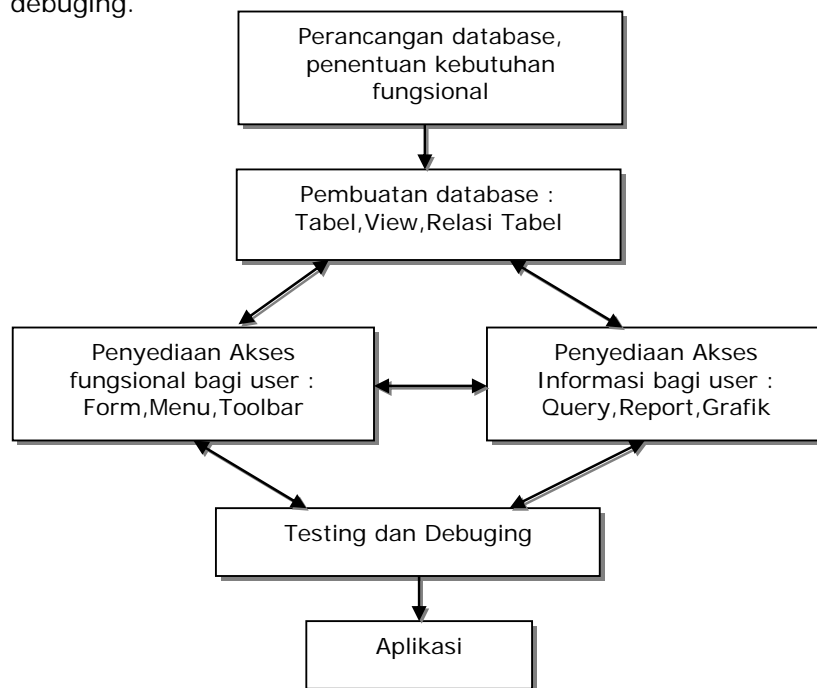


Diagram Proses Pembuatan Aplikasi Database

## 1.2. Memulai Visual Foxpro

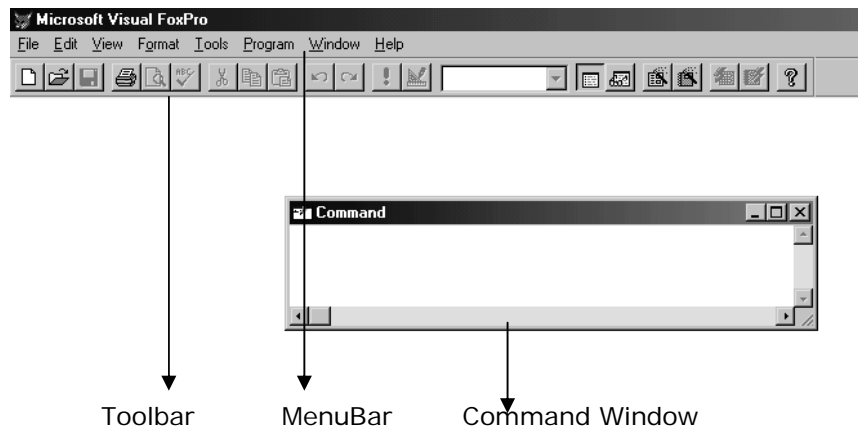
Ada beberapa cara memulai perangkat lunak Visual Foxpro, yaitu :

1. Memulai Visual Foxpro lewat menu **Start – Programs – Microsoft Visual Foxpro**.
2. Memulai Visual Foxpro lewat **Shortcut Visual Foxpro**.

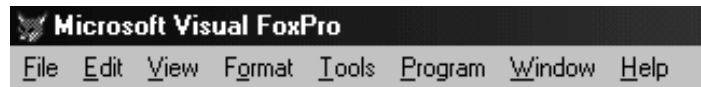
## 1.3. Tampilan Utama Visual Foxpro

### 1.3.1. Jendela Microsoft Visual Foxpro

Saat Microsoft Visual Foxpro dijalankan, sebuah jendela kerja akan tampil seperti gambar dibawah ini.



### 1.3.2. Menu Bar



### 1.3.3. ToolBar

ToolBar, seperti pada gambar dibawah, berisi ikon-ikon yang berfungsi mempersingkat perintah pada Menu Bar.



### 1.3.4 Jendela Project Manager

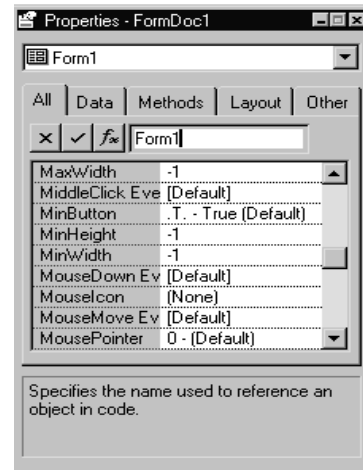


Jendela Project Manager menunjukkan proyek aktif dan file-file penyusun proyek tersebut.

### 1.3.5 Jendela Properties

Jendela Properties digunakan untuk mengubah properti suatu objek. Pengaturan properti pada jendela Properties dapat dilakukan dengan berbagai cara :

- Dengan mengetikkan nilai yang diinginkan (contoh : properti Caption, Width).
- Memilih nilai yang telah ada (contoh : properti Style, Aligment).
- Memilih sampel yang ada (contoh : properti BackColor, ForeColor).
- Browsing (contoh : properti Picture).



Selain dengan jendela properties, pengaturan properti juga dapat dilakukan lewat program melalui baris-baris listing.

## 1.4. Elemen Dasar Visual Foxpro

Elemen dasar Visual Foxpro terdiri dari Tipe Data (Type Data), Operator, Fungsi/Procedure dan perintah.

### 1.4.1. Tipe Data

Dalam membuat program tidak terlepas dari rumus dan formula yang berupa ekspresi. Dalam Visual Foxpro ekspresi atau rumus tidak boleh jenis data (type data) bercampur.

Tipe-tipe data yang tersedia dalam Visual Foxpro adalah karakter, karakter biner, numerik dan bilangan pecahan, bilangan bulat, bilangan pecahan ganda, bilangan uang, tanggal, tanggal plus jam, logika (L) dan memo (M).

#### Tipe Karakter

Tipe karakter adalah tipe data string berupa deretan karakter alfabet, dari A sampai Z, dan angka 0 – 9 yang dianggap karakter. Tipe karakter dibuat menggunakan (diapit) tanda kutip tunggal, kutip ganda atau tanda siku.

Contoh :

```
'Visual Foxpro'
"STMIK Tasikmalaya"
[Visual Foxpro Programming]
```

#### Tipe karakter Biner

Tipe karakter biner seperti tipe data karakter biasa, tetapi bersifat permanen.

#### Tipe Numerik dan Bilangan Pecahan

Tipe numerik (float) adalah tipe angka atau angka pecahan yang mempunyai jangkauan -0.999999999E+19 sampai dengan 0.999999999E+20.

**Tipe Bilangan Bulat**

Tipe bilangan bulat adalah tipe data angka yang mempunyai jangkauan antara -2147483647 sampai dengan 2147483646.

**Tipe Bilangan Pecahan Ganda**

Tipe bilangan pecahan ganda (lebih besar dari tipe float) adalah tipe angka atau angka pecahan yang mempunyai jangkauan +/-4.94065645841247E-324 sampai +/-8.9884656743115E307.

**Tipe Bilangan Uang**

Tipe bilangan uang adalah tipe data untuk bentuk data berupa uang yang mempunyai jangkauan -922337203685477.5808 sampai dengan 922337203685477.5807. Untuk menyatakan bilangan bentuk uang kita harus menggunakan simbol \$ (bentuk default).

**Tipe Tanggal**

Tipe tanggal adalah tipe data untuk bentuk data tanggal (date). Format defaultnya adalah bentuk tanggal US, dengan format mm/dd/yy (jika SET CENTURY OFF). Untuk memudahkan kita membentuk format tanggal tertentu, perhatikan tabel dibawah ini :

Pengaturan	Format saat SET CENTURY OFF
AMERICAN	mm/dd/yy
ANSI	yy.mm.dd
BRITISH/FRENCH	dd/mm/yy
GERMAN	dd.mm.yy
ITALIAN	dd-mm-yy
JAPAN / TAIWAN	yy/mm/dd
USA	mm-dd-yy
MDY	mm/dd/yy
DMY	dd/mm/yy
YMD	yy/mm/dd

**Tipe Logika (L)**

Tipe logika (L) adalah tipe data untuk bentuk logika TRUE (benar), dapat digunakan simbol T,t,Y atau FALSE (salah) dapat digunakan simbol F,f,N,n.

**Tipe Memo**

Tipe memo adalah tipe data untuk digunakan sebagai karakter panjang, dan hanya berlaku untuk field bukan variabel. Memo berukuran tidak tetap, seperti karakter dan disimpan dalam file .FPT.

## Tipe Umum (G)

Tipe data umum biasa dapat dipakai untuk menyimpan data OLE (Object Linking Embedded), seperti spreadsheet, grafik, suara yang dibuat dari aplikasi lain.

### 1.4.2. Operator

Operator adalah simbol yang dipakai untuk operasi tertentu, biasanya dipakai dalam ekspresi atau rumus tertentu. Contoh simbol – simbol operator seperti pada tabel berikut :




Operator	Kegunaannya	Untuk tipe data
=	Pengujian kesamaan	Semua tipe
+	Penjumlahan	Numeric, Character, Date, DateTime
! atau NOT	Negasi (kebalikan)	Logical
*, /	Perkalian/Pembagian	Numeric
^	Perpangkat	Numerik

**Catatan :** Dalam suatu ekspresi variabel harus memiliki tipe data yang sama, jika tidak sama harus dikonversikan dahulu tipe datanya.

### 1.4.3. Fungsi / Procedure

Fungsi atau procedure adalah kumpulan instruksi tertentu untuk tujuan tertentu yang diberi nama, sehingga jika ingin dipakai hanya menyebutkan nama fungsinya saja.

## 1.5. Perintah-perintah dasar Foxpro

-  **SET DEFAULT TO <path-drive>**  
 Digunakan untuk mengaktifkan/merubah drive kerja.  
 Contoh :       SET DEFAULT TO C:\FOXPRO ↵
-  **CLOSE ALL**  
 Digunakan untuk menutup semua jendela yang aktif.
-  **QUIT**  
 Perintah untuk keluar dari Visual Foxpro.

## 1.6. Aplikasi Dengan Form

Form merupakan sebuah objek pada Visual Foxpro yang digunakan untuk mengolah data tabel atau sebagai dasar rancangan tampilan aplikasi lainnya.

Untuk membuat form dapat kita lakukan dengan 3 cara yaitu :

1. Dengan bantuan Wizard
2. Desain
3. Programming

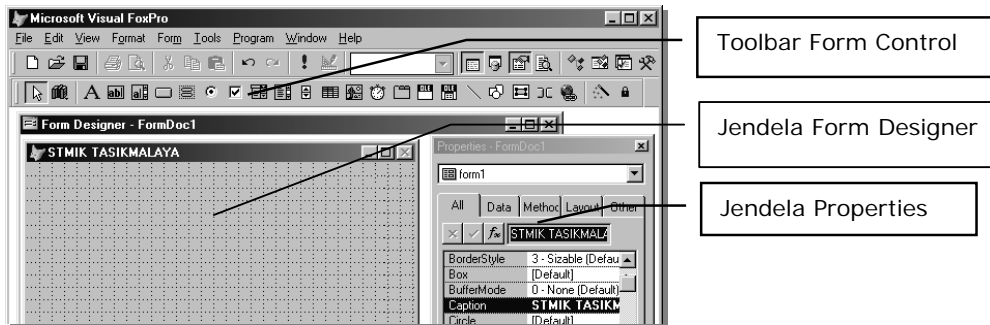
## Desain Form dan Modifikasi Form

Setelah kita bisa membuat form dengan Wizard, dan terlihat ada kelemahan. Maka untuk itu kita harus menggunakan cara kedua yaitu dengan mendesain Form sendiri. Langkah untuk membuat/mendesain form menurut keinginan sendiri sbb :

- Pada Command Window, ketikkan perintah :

**CREATE FORM <nama-form>** atau

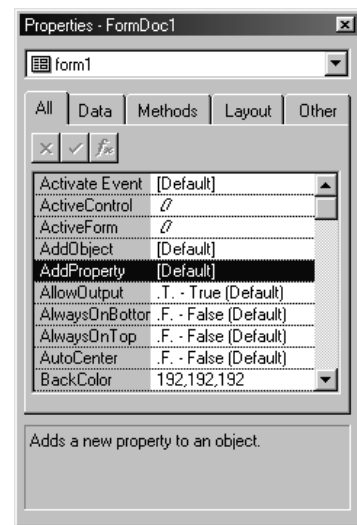
pilih menu **FILE – NEW – FORM – NEW FILE**, sehingga muncul kotak dialog :



### Jendela Properties

Jendela Properties digunakan untuk mengubah properti suatu objek. Pengaturan properti pada jendela Properties dapat dilakukan dengan berbagai cara :

- Dengan mengetikkan nilai yang diinginkan (contoh : properti Caption, Width).
- Memilih nilai yang telah ada (contoh : properti Style, Aligment).
- Memilih sampel yang ada (contoh : properti BackColor, ForeColor).
- Browsing (contoh : properti Picture).



- Mengaktifkan/menjalankan Form











Untuk menjalankan/mengaktifkan sebuah Form langkahnya yaitu pada Commad Window ketikkan perintah : **DO FORM <nama-form>** atau

Pilih Menu **Program – Do – <nama-file-fom>** atau

Pada Toolbar, klik tombol  atau pada Keyboard tekan **CTRL+E**.

- Menambah/mendesain Kontrol Form

Untuk menambahkan sebuah kontrol baru pada sebuah form, caranya adalah klik sebuah kontrol yang kita inginkan dari Toolbar Form Control, kemudian dilam jendela Form designe, Drag dan Drop pointer mouse untuk membuat kotak kontrol yang kita inginkan. Beberapa jenis kontrol yang biasa dipakai adalah :

Toolbar	Name	Function
	Label	Digunakan untuk membuat komentar atau keterangan pada Form
	Textbox	Sebuah kontrol yang dapat berisi data field atau sebuah variabel memori.
	EditBox	Sebuah kontrol yang dapat berisi data field atau sebuah variabel memori dan mempunyai scrolling bar.
	Command Button	Digunakan untuk mengeksekusi sebuah perintah program tertentu.
	Option	Digunakan untuk memberikan beberapa pilihan/option yang hanya bisa dipilih satu.
	CheckBox	Digunakan untuk memberikan pilihan/option yang bisa di pilih lebih dari satu.
	ComboBox	Digunakan untuk menginputkan data sekaligus bisa digunakan untuk pemilihan data tertentu.
	Grid	Digunakan untuk menampilkan data dari suatu tabel.
	Image	Fungsinya untuk menampilkan gambar yang biasanya diambil dari file yang terpisah.
	Line	Digunakan untuk membuat sebuah garis.

### TUGAS PRAKTIKUM I

1. Kerjakan Form dibawah ini Desain sendiri sehingga hasilnya seperti dibawah ini. Kemudian simpan dengan nama file **FORM1A.SCX**.



- Ketikkan pada objek **Form** Procedure **Init** dan Object **Command1** Procedure **Click**.

```
Thisform.text1.Value=""
Thisform.text2.Value=0
Thisform.text3.Value=0
Thisform.text4.Value=0
Thisform.text1.SetFocus()
```

- Ketikkan pada Objek **Command2** Procedure **Click**.

```
Thisform.release()
```

- Ketikkan pada Objek **Text3** Procedure **Valid**

```
A = Thisform.text2.value
B = Thisform.text3.value
C = A * B
Thisform.text4.value = C
```

## TUGAS PRAKTIKUM II

1. Buatlah sebuah program untuk menghitung penjumlahan, pengurangan, perkalian dan pembagian dari dua buah bilangan dengan desain output seperti berikut :

2. Simpanlah form tersebut dengan nama FORM1B.SCX

## TUGAS PRAKTIKUM III

1. Buatlah program untuk menghitung konversi dari celcius menjadi reamur, fanrenheit dan kelvin menggunakan rumus sebagai berikut :

Reamur =  $4/5$  celcius

Fanrenheit =  $9/5$  Celcius + 32

Kelvin = Celcius + 273

2. Dengan desain output seperti berikut :

3. Simpan form tersebut dnegan nama FORM1C.SCX



# Modul 2

## Logika Percabangan

### Proses Percabangan Bersyarat

Dari pertemuan sebelumnya, program yang kita buat adalah program yang bersifat sequence/terurut Artinya proses dimulai dari baris instruksi pertama dan berlanjut terus sampai instruksi terakhir secara terurut. Seandainya sebuah instruksi perlu meloncat atau mengerjakan sebuah instruksi dengan suatu syarat tertentu, maka diperlukan suatu perintah pengulangan dan perintah percabangan.

### Struktur Percabangan

Percabangan kondisional memungkinkan kita memeriksa suatu kondisi, kemudian berdasarkan hasil pemeriksaan tersebut membentuk operasi berbeda. Terdapat dua perintah di Visual Foxpro yang dipakai untuk membuat percabangan kondisional.

#### Perintah Logika IF

Perintah ini mengeksekusi secara kondisional sekumpulan perintah berdasarkan nilai ekspresi logika. Secara umum, perintah percabangan bersyarat yang biasa digunakan antara lain :

- **IF ... ENDIF**
- **IF ... ELSE ... ENDIF**

Perintah IF ... ENDIF

Sintak perintah ini adalah :

<pre> <b>IF &lt;kondisi&gt;</b>     .....     .....     ..... <b>ENDIF</b> </pre>	<p>Bagian program yang dijalankan bila kondisi terpenuhi/benar</p>
---	--

Perintah IF ... ELSE ... ENDIF

Sintak perintah ini adalah :

```

IF <kondisi>
    <Bagian program yang akan dijalankan bila kondisi benar>
ELSE
    <Bagian program yang akan dijalankan bila kondisi salah>
ENDIF

```

**Perintah DO CASE ... ENDCASE**

Perintah ini mengeksekusi sejumlah perintah pertama yang ekspresi kondisinya bernilai True (.T).

Sintak perintah ini adalah :

**DO CASE**

**CASE** <ekspresi-1 >

*<Bagian program yang akan dijalankan bila kondisi-1 benar>*

**CASE** <ekspresi-2 >

*<Bagian program yang akan dijalankan bila kondisi-2 benar>*

...

**CASE** <ekspresi-n >

*<Bagian program yang akan dijalankan bila kondisi-n benar>*

**OTHERWISE**

*<Bagian program yang akan dijalankan bila semua kondisi salah>*

**ENDCASE**

Buatlah contoh form dengan layout seperti dibawah ini :

Ketentuan Proses :**Nilai Abjad >>**

> 85 abjad A

76-85 abjad B

66- 75 abjad C

56-65 abjad D

<55 abjad E

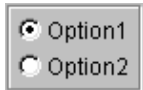
**Keterangan >>**

Jika nilai\_akhir>55, Lulus

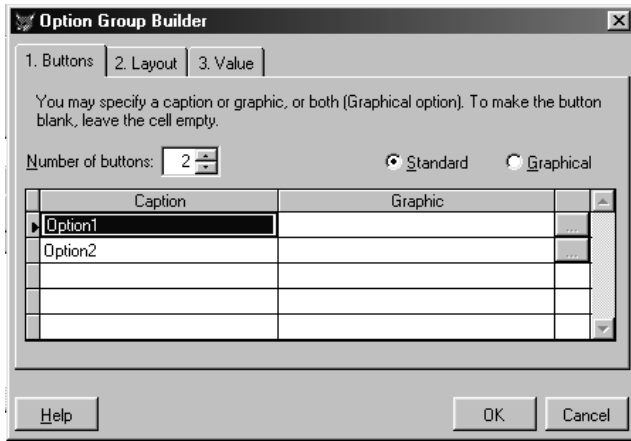
Jika nilai\_akhir<55, Tidak Lulus

Langkah – Langkah :

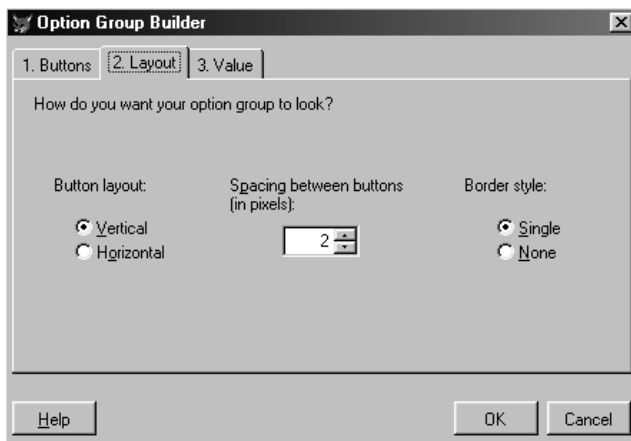
- Untuk membuat pilihan Jurusan gunakan control OptionGroup, langkahnya :
  - Tambahkan objek Optiongroup dari form control ke form, maka akan muncul :



- Untuk mengganti masing-masing option, klik kanan pada objek optiongroup tersebut, kemudian pilih **Builder**, maka akan muncul :



- Pada **Number Of Button**, ubah menjadi 2 ( untuk menentukan banyaknya option/pilihan pada Optiongroup tersebut).kemudian ganti masing-masing caption.
- Klik Tab Layout, untuk mengatur tampilan option.



- Untuk menset format angka pada object text3, ubahlah properties **Value = 0**.

- Pada objek Text3, masukkan code program dibawah ini pada procedure **InteractiveChange**.

```
nilai=thisform.text3.value
if nilai>=55
    thisform.text5.value="Lulus"
else
    thisform.text5.value="Tidak Lulus"
endif

do case
    case nilai>85
        thisform.text4.value="A"
    case nilai>75
        thisform.text4.value="B"
    case nilai>65
        thisform.text4.value="C"
    case nilai>55
        thisform.text4.value="D"
    otherwise
        thisform.text4.value="E"
endcase
```

- Pada objek Command1 (tombol Data baru), masukkan code program pada procedure **Click**.

```
thisform.text1.value=""
thisform.text2.value=""
thisform.optiongroup1.value=1
thisform.text3.value=0
thisform.text4.value=""
thisform.text5.value=""
thisform.text1.setfocus
```

- Pada objek Command2 (tombol Keluar), masukkan code program pada procedure **Click**.

```
Thisform.release
```

## TUGAS PRAKTIKUM IV

1. Kerjakan Form dibawah ini untuk memasukkan daftar pembelian komputer dan asesorisnya. Kemudian simpan dengan nama **FORM2A.SCX**

Aturlah Properties Object form diatas:

OBJECT	PROPERTIES	NILAI/VALUE
Form1	Caption	FORM PEMBELIAN
	AutoCenter	.T. – True
Label1	Caption	DAFTAR PEMBELIAN KOMPUTER & ACCESSORIES
	AutoSize	.T. – True
	FontSize	10
	FontBold	.T. – True
Label2	Caption	NAMA PEMBELI
	AutoSize	.T. – True
Label3	Caption	NAMA BARANG
	AutoSize	.T. – True
Label4	Caption	HARGA SATUAN
	AutoSize	.T. – True
Label5	Caption	JUMLAH BARANG
	AutoSize	.T. – True
Label6	Caption	TOTAL BAYAR
	AutoSize	.T. – True
Label7	Caption	DIBAYAR
	AutoSize	.T. – True
Label8	Caption	SISA
	AutoSize	.T. – True
Label9	Caption	KETERANGAN
	AutoSize	.T. – True
Check1	Caption	LUNAS
	AutoSize	.T. – True
Command1	Caption	\<MULAI
Command2	Caption	\<SELESAI

Object OptionGroup1 :

Option1	Caption	NAMA PEMBELI
---------	---------	--------------

	Autosize	.T. – True
Option2	Caption	NAMA PEMBELI
	Autosize	.T. – True

Ketikkan Procedure/Event pada Object dibawah ini :

- **Form1** Procedure **Init** dan **Command1** Procedure **Click**

```

Thisform.text1.Value = ""
Thisform.combo1.Value = ""
Thisform.text2.Value = 0
Thisform.text3.Value = 0
Thisform.text4.Value = 0
Thisform.text5.Value = 0
Thisform.text6.Value = 0
Thisform.check1.Value = 0
Thisform.optiongroup1.Value = 0
Thisform.text1.SetFocus()

```

- **Command2** Procedure **Click**

```

Thisform.Release()

```

- **Combo1** Procedure **LostFocus**

```

Kode = Thisform.combo1.Value
DO Case
    CASE kode = "KOMPUTER"
        hrg = 5000000
    CASE kode = "KEYBOARD"
        hrg = 25000
    CASE kode = "MONITOR"
        hrg = 750000
    CASE kode = "MOUSE"
        hrg = 30000
ENDCASE
Thisform.text2.Value = hrg

```

- **Text3** Procedure **Valid**

```

hrg = Thisform.text2.Value
jml = Thisform.text3.Value
Tb = Hrg * Jml
Thisform.text4.Value = Tb

```

- **Text5** Procedure **Valid**

```

Tb = Thisform.text4.Value
Dbyr = Thisform.text5.Value
Ss = Tb - Dbyr
DO CASE
    CASE ss = 0
        Thisform.label8.Caption = "SISA"
        Kt1 = 1
        kt2_1 = 0
        kt2_2 = 1

```

```

CASE ss>0
    Thisform.label8.Caption = "SISA"
    Kt1=0
    Kt2_1=0
    Kt2_2=1
CASE ss<0
    Thisform.label8.Caption = "KEMBALI"
    Kt1=1
    Kt2_1=1
    Kt2_2=0
ENDCASE
Thisform.text6.Value = ss
Thisform.check1.Value = kt1
Thisform.optiongroup1.option1.Value = kt2_1
Thisform.optiongroup1.option2.Value = kt2_2

```

- Setelah selesai, jalankan Form tersebut dan lihatlah hasilnya.

### TUGAS PRAKTIKUM V

- Buatlah program untuk menggeser kalimat dari atas ke bawah dan dari kiri ke kanan, dngan desain output seperti berikut ini !



Aturlah properties objek  
Timer1 dan Timer2 :

**Interval = 100**

- Ketikkan listing program sebagai berikut :

- Pada objek Timer1, klik Timer, letakkan listing program berikut ini :

```

If Thisform.label1.top <=250
    bantu = thisform.label1.Top
    thisform.label1.Top=bantu+3
ELSE
    thisform.label1.Top=0
ENDIF

```

- Pada objek Timer2, Klik Timer, letakkan listing program berikut ini :

```

If Thisform.label1.left <=350
    bantu = thisform.label1.left
    thisform.label1.left=bantu+3
ELSE
    thisform.label1.left=0
ENDIF

```

- Simpanlah form tersebut dengan nama FORM2B.SCX

## TUGAS PRAKTIKUM VI

1. Buatlah program aplikasi untuk menghitung total bayar pada suatu hotel dibawah ini. Aturlah Tampilan dan Properties Form tersebut sehingga tampilannya seperti dibawah ini :

Ketentuan Program :

- Jenis kamar yang ditawarkan : EKONOMI,BISNIS,EXECUTIVE,SWEETROOM
- Nama Tamu, Jenis Pengunjung, Tarif/hari, Tgl Masuk, Tgl Keluar dan Biaya lain-lain di inputkan
- Lama Menginap, pajak wisata, Biaya menginap dan total bayar otomatis ditampilkan.
- Lama Menginap Dihitung dari Tanggal keluar – Tanggal Masuk
- Biaya Menginap ditampilkan berdasarkan jenis kamar yang dipilih

Jenis Kamar	Biaya Nginap/hari
EKONOMI	150000
BISNIS	300000
EXECUTIVE	600000
SWEETROOM	850000

- Pajak Wisata dihitung dari 10% dari Biaya Menginap
  - Total Bayar dihitung dari (Biaya Menginap\*Lama Menginap)+Biaya Lain-lain+Pajak Wisata
2. Simpan Form tersebut dengan nama **FORM2C.SCX**



# Modul 3

## Logika Pengulangan / looping

### Proses Pengulangan

Perulangan memungkinkan kita mengeksekusi satu atau lebih baris kode sebanyak yang kita kehendaki. Terdapat tiga perintah di Visual Foxpro yang dipakai untuk perulangan, yaitu :

#### Perintah DO WHILE – ENDDO

Perintah ini mengeksekusi perintah-perintah yang ada didalam loop kondisional. Perulangan akan dilakukan terus menerus selama kondisi tersebut terpenuhi (.T. atau Benar).

Sintak dari perintah pengulangan tersebut adalah :

#### DO WHILE

```
.....
.....
.....
```

Bagian program yang akan diulang

#### ENDDO

Pengulangan tersebut akan dilakukan bila kondisi benar. Pengulangan dengan kondisi ini dapat pula menggunakan kondisi .T., yaitu :

#### DO WHILE .T.

```
.....
.....
.....
[Loop]
[Exit]
```

Bagian program yang akan diulang

#### ENDDO

*Contoh :*

```
x=1
DO WHILE x <= 20
  ? x
  x=x+1
ENDDO
```

Pengulangan EOF() dan BOF() Dalam Proses Berulang

Proses pengulangan dapat dilakukan dalam memproses record-record dalam suatu tabel satu persatu hingga habis. Untuk membatasi pengulangan biasa digunakan fungsi EOF() (End Of File) dan BOF() (Begin Of File). Fungsi tersebut akan memberikan fungsi nilai logika .T. bila pointer telah berada diakhir file.

Bentuk pengulangan untuk memproses file data sampai habis :

```

DO WHILE.NOT.EOF()
    .....
    .....
    .....
ENDDO

```

| Bagian program yang akan diulang

**Perintah FOR . . . ENDFOR**

Perulangan ini akan mengeksekusi perintah-perintah sebanyak yang kita tentukan. Nilai yang kita tentukan merupakan nilai tetap/konstanta.

Sintak dari perintah pengulangan tersebut adalah :

```

FOR <var> = <nilai_awal> TO <nilai_akhir>
    .....
    .....
    .....
ENDFOR

```

| Bagian program yang akan diulang

*Contoh :*

```

FOR   v_angka = 1 TO 10
      ? v_angka
ENDFOR

```

**Perintah SCAN . . . ENDSCAN**

Perintah ini memindahkan pointer record melewati tabel yang sedang dipilih dan mengeksekusi blok perintah setiap kali record memenuhi syarat kondisi yang ditentukan.

Sintak dari perintah pengulangan tersebut adalah :

```

SCAN [FOR <kondisi1>] [WHILE <kondisi2>]
    .....
    .....
    .....
[loop]
[Exit]
ENDSCAN

```

| Bagian program yang akan diulang

*Contoh :*

```

SCAN FOR kota = "Tasikmalaya"
      ? nama, jurusan
ENDSCAN

```

## TUGAS PRAKTIKUM VII

1. Buatlah program perulangan untuk menampilkan bilangan 1 sampai dengan 10 menggunakan fungsi for dan do while dengan desain outputnya seperti berikut. Kemudian simpan form tersebut dengan nama **FORM3A.SCX**



2. Kemudian isi kode program sesuai dengan objek dan event yang diperlukan, antara lain :

- Pada objek Tombol For, klik Click event, letakkan listing program berikut ini :
 

```
FOR i = 1 TO 10
    Thisform.list1.AddListItem('Nilai For Ke =' + ALLTRIM(STR(i)))
ENDFOR
```
- Pada objek Tombol Do While, klik Click event, letakkan listing program berikut ini :
 

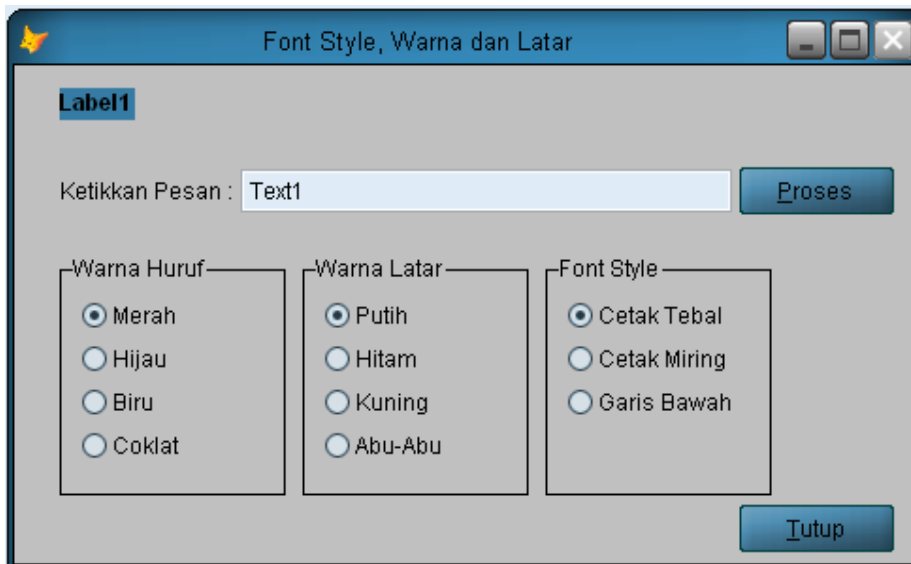
```
i=1
DO WHILE i<=10
    thisform.list1.AddListItem('Nilai Do While Ke =' + ALLTRIM(STR(i)))
    i=i+1
ENDDO
```
- Pada objek Tombol Hapus, klik Click event, letakkan listing program berikut ini :
 

```
thisform.list1.Clear
```
- Pada objek Tombol Keluar, klik Click event, letakkan listing program berikut ini :
 

```
thisform.Release
```

## TUGAS PRAKTIKUM VIII

1. Buatlah program untuk menformat huruf dengan font style, warna dan latar dengan desain outputnya seperti berikut. Simpan form tersebut dengan nama **FORM3B.SCX**



2. Ketikkan source code untuk masing-masing objek :
  - Pada objek OptionGroup1-Option1, klik Click event, letakkan listing berikut :  
`thisform.label1.ForeColor = RGB(255,0,0)`
  - Pada objek OptionGroup1-Option2, klik Click event, letakkan listing berikut :  
`thisform.label1.ForeColor = RGB(0,255,0)`
  - Pada objek OptionGroup1-Option3, klik Click event, letakkan listing berikut :  
`thisform.label1.ForeColor = RGB(0,0,128)`
  - Pada objek OptionGroup1-Option3, klik Click event, letakkan listing berikut :  
`thisform.label1.ForeColor = RGB(128,64,64)`
  - Pada objek OptionGroup2-Option1, klik Click event, letakkan listing berikut :  
`thisform.label1.backColor = RGB(255,255,255)`
  - Pada objek OptionGroup2-Option2, klik Click event, letakkan listing berikut :  
`thisform.label1.backColor = RGB(0,0,0)`
  - Pada objek OptionGroup2-Option3, klik Click event, letakkan listing berikut :  
`thisform.label1.backColor = RGB(255,255,0)`

- Pada objek OptionGroup2-Option4, klik Click event, letakkan listing berikut :  
`thisform.label1.backgroundColor = RGB(192,192,192)`
- Pada objek OptionGroup3-Option1, klik Click event, letakkan listing berikut :  
`thisform.label1.FontBold = .t.`  
`thisform.label1.FontUnderline = .F.`  
`thisform.label1.FontItalic = .F.`
- Pada objek OptionGroup3-Option2, klik Click event, letakkan listing berikut :  
`thisform.label1.FontBold = .F.`  
`thisform.label1.FontUnderline = .F.`  
`thisform.label1.FontItalic = .T.`
- Pada objek OptionGroup3-Option3, klik Click event, letakkan listing berikut :  
`thisform.label1.FontBold = .F.`  
`thisform.label1.FontUnderline = .T.`  
`thisform.label1.FontItalic = .F.`
- Pada objek Tombol Proses, klik Click event, Letakkan listing Berikut :  
`thisform.label1.Caption = thisform.text1.Value`
- Pada objek Tombol Keluar, klik Click event, Letakkan listing Berikut :  
`thisform.Release`

# Modul 4

## Fungsi Konversi & Message Box

### 4.1 FUNGSI

Dalam Visual foxpro menyediakan beberapa fungsi yang biasa dipakai dalam pemograman. Fungsi tersebut antara lain fungsi konversi, fungsi string, fungsi tanggal dan waktu dll.

#### 4.1.1 Fungsi Konversi

Fungsi ini berguna untuk mengubah elemen data dari suatu jenis ke jenis yang lainnya. Fungsi tersebut antara lain :

- **VAL (string alphanumerik)**  
Untuk mengubah string bilangan ke nilai numerik
- **DTOC (date)**  
Untuk merubah isi field date(tanggal) menjadi karakter
- **CTOD (string alphanumerik)**  
Untuk merubah string alphanumerik menjadi bentuk tanggal.

#### 4.1.2 Fungsi Tanggal Dan Waktu

- **TIME()** Menampilkan waktu yang ditunjukkan sistem komputer pada saat sekarang.
- **DATE()** Menampilkan tanggal sekarang
- **CDOW(date)** Menunjukkan nama hari yang ditunjukkan date.
- **CMONTH(date)** Menunjukkan nama bulan yang ditunjukkan date.
- **DOW(date)** Menunjukkan nama hari dalam seminggu. 1=minggu,2=senin,3=selasa,4=rabu,5=kamis,6=Jumat,7=sabtu

#### 4.1.3 Fungsi String

- **UPPER<alphanumerik>** : Merubah teks/huruf menjadi huruf kapital.
- **LOWER<alphanumerik>** : Merubah teks/huruf menjadi huruf kecil.
- **TRIM<alphanumerik>** : Untuk menghapus/memotong spasi.
- **LTRIM<alphanumerik>** : Untuk menghapus spasi dari sebelah kiri.
- **RTRIM<alphanumerik>** : untuk menghapus spasi dari sebelah kanan.
- **SUBSTR<alphanumerik>** : untuk mengambil karakter/kata dari suatu kalimat/kata.
- **LEFT<alphanumerik>** : untuk mengambil karakter dari kiri.
- **RIGHT<alphanumerik>** : untuk mengambil karakter dari kanan.
- **LEN<alphanumerik>** : untuk menghitung jumlah karakter.
- **ASC<alphanumerik>** : untuk menghasilkan nilai numerik dari karakter ASCII

## TUGAS PRAKTIKUM IX

1. Kerjakan Form dibawah ini untuk menggunakan perintah atau fungsi yang ada dalam foxpro. Kemudian simpan form tersebut dengan nama **FORM4A.SCX**

Ketikkan Procedure/Event pada Object dibawah ini :

- **Text1** Procedure **InteractiveChange**

WITH thisform

```
xtext = ALLTRIM(.text1.Value )
.text2.Value = LEFT(xtext,3)
.text3.Value = right(xtext,3)
.text4.Value = Substr(xtext,2,4)
.text5.Value = Len(xtext)
.text6.Value = Upper(xtext)
.text7.Value = Lower(xtext)
.text8.Value = Proper(xtext)
```

ENDWITH

- **Command1** Procedure **Click**

Thisform.release

3. Jalankan form tersebut dan lihatlah hasilnya.

## TUGAS PRAKTIKUM X

1. Kerjakan Form dibawah ini untuk menggunakan menggunakan fungsi operasi tanggal. Simpan form tersebut dengan nama **FORM4B.SCX**

Aturlah properties dibawah ini :

Nama Object	Properties	Value
Text1	ReadOnly	T - True
Text3	ReadOnly	T - True
Text4	ReadOnly	T - True
Text5	ReadOnly	T - True
Text6	ReadOnly	T - True
Text7	ReadOnly	T - True
Text8	ReadOnly	T - True
Command1	Caption	\<Jalankan
Command2	Caption	\<Keluar
Timer1	Interval	1000

Ketikkan Procedure/Event pada Object dibawah ini :

a. **Form1** Procedure **Init**

```
SET CENTURY ON && menampilkan format tanggal 4 digit
SET DATE BRITISH && format tanggal dd/mm/yy
thisform.text2.Value = DATE() && memasukkan tgl ke objek text1
```

b. **Timer1** Procedure **Timer**

```
Thisform.text1.Value = TIME() && memasukkan nilai waktu/jam yang sedang aktif
```

c. **Command1** Procedure **Click**

```
WITH thisform
    tanggal = .text2.value
    tglke = DAY(tanggal)
```



```

harike    = DOW(tanggal)
namahari  = CDOW(tanggal)
tahun     = YEAR(tanggal)
bulanke   = MONTH(tanggal)
namabln   = CMONTH(tanggal)

```

```

.text3.value = tglke
.text4.value = harike
.text5.value = namahari
.text6.value = tahun
.text7.value = bulanke
.text8.value = namabln
ENDWITH

```

d. **Command2 Procedure Click**

```
Thisform.release()
```

#### 4.2. KOTAK DIALOG / MESSAGEBOX

Kotak dialog digunakan untuk memberitahukan suatu informasi kepada user/pemakai, misalnya menampilkan suatu pesan kesalahan. Selain itu, kotak dialog messagebox juga digunakan untuk melakukan konfirmasi terhadap pemakai, misalnya konfirmasi YES dan NO.

Sintak Messagebox :

```
Messagebox("pesan", <tipe_kotak_dialog>, <judul>)
```

Argument kedua dan ketiga dapat juga tidak diikuti sertakan.

Contoh penggunaannya :

```
Messagebox("Yakin akan menutup program  
ini?", 4 + 32 + "Konfirmasi")
```

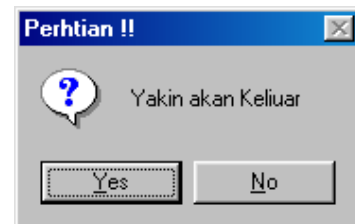
Atau

```
Messagebox("Data yang dimasukkan sudah ada !!!")
```





Ini layout untuk pesan konfirmasi "Yes" dan "No".

#### **Tabel Nilai untuk jenis Kotak dialog :**

Nilai Tipe Kotak	Tombol yang tersedia/tampil
0	Ok
1	Ok, Cancel
2	Abort, Retry, Ignore
3	Yes, No, Cancel
4	Yes, No
5	Retry, Cancel



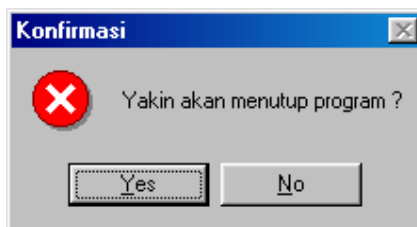
**Tabel Icon yang akan ditampilkan pada Kotak Dialog :**

Nilai Icon	Gambar Icon
16	
32	
48	
64	

**Nilai Tombol yang Dipilih :**

Nilai	Tombol yang dipilih
1	Ok
2	Cancel
3	Abort
4	Retry
5	Ignore
6	Yes
7	No

Untuk menyatakan tombol yang dipilih, dapat menggunakan pernyataan IF atau DO CASE.  
Contoh penggunaan :



```
Jwb = MessageBox(" Yakin akan menutup program ?",4+16+"Konfirmasi")
IF Jwb = 6    && jika tombol YES dipilih maka keluar dari program
    Thisform.release
ENDIF
```

---

**PENGATURAN PROPERTIES DAN EVENT OBJECT**

Obyek	Keterangan
Name	Memberikan nama sebuah obyek
Caption	Memberikan judul pada sebuah obyek
Text	Hampir sama dengan caption
FontName	Menentukan jenis huruf
BackColor	Menentukan warna latar belakang
ForeColor	Menentukan warna text atau huruf
Enabled	Mengatur kontrol supaya tidak dapat diakses
Visible	Menyembunyikan obyek
Height	Menentukan tinggi dari sebuah obyek
Width	Menentukan lebar dari sebuah obyek
Top	Menentukan posisi obyek dari atas
Left	Menentukan posisi obyek dari kiri
TabIndex	Menentuka urutan pemfocusan
Dragmode	Mengatur tingkah laku kontrol selama dragand drop
Mousepointer	Memungkinkan perubahan bentuk mouse pada saat melewati obyek
Autosize	Mengatur agar lebar suatu objek menyesuaikan dengan lebar objek
FontBold	Mengatur agar objek di Bold/cetak tebal
BackStyle	Mengatur agar objek transparan atau tidak.

#### Macam-macam Event

Click	Terjadi apabila menekan dan melepaskan tombol kiri mouse pada obyek
DbClick	Menekan tombol kiri 2 x pada sebuah obyek
DragDrop	Menahan tombol kiri dan menyeret obyek dari satu tempat ke tmpt lain
Activate	Terjadi apabila sebuah obyek sedang aktif
DeActivate	Kebalikan dari activate
GotFocus	Terjadi apabila sebuah obyek satu-satunya menjadi focus
LostFocus	Terjadi apabila sebuah obyek kehilangan Focus
Load	Terjadi apabila sebuah form dibuka atau dipanggil
UnLoad	Terjadi apabila sebuah form ditutup
Init	Memberikan nilai awal pada sebuah form
Valid	Memberikan validasi terhadap objek
KeyPress	Terjadi apabila menerima input tombol dari keyboard
Interactive Change	Terjadi apabila sebuah objek terjadi perubahan nilai

# Modul 5

## Membuat Project dan Database

### 5.1. Pendahuluan

Dalam program Visual Foxpro sebaiknya kita membuat program diawali dengan membuat sebuah Project, karena seluruh program, form, database, report, view, query dan lain-lainnya terletak dalam project. Hal ini akan memudahkan kita mengorganisasikannya. Dalam project bisa terdapat beberapa database yang bisa saling berhubungan. Sementara itu, dalam database bisa terdapat beberapa tabel yang masing-masing tabel dapat dihubungkan oleh kunci.

### 5.2. Project

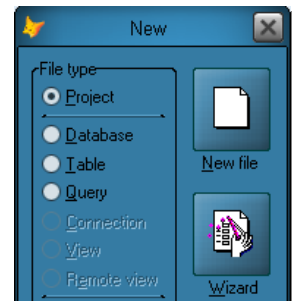
Project adalah suatu tempat untuk meletakkan data, dokumen, kelas, kode/program, dan lain-lainnya. Untuk membuat project ada 2 cara yaitu :

1. Lewat menu
2. Lewat perintah Visual Foxpro (pada command window)

#### 5.2.1. Membuat Project Lewat Menu

Untuk membuat Project lewat menu, lakukan langkah-langkah berikut ini :

1. Pada menu, pilih **File > New**
2. Kemudian pilih option **Project**
3. Klik **New**



#### 5.2.2. Membuat Project Lewat Perintah Visual Foxpro

Untuk membuat Project lewat perintah Visual Foxpro, kita dapat mengetikkan perintah pada jendela Command Window. Perintahnya adalah :

**Create Project [namaproject]**

### 5.3. Database

Pengertian database dalam visual foxpro berbeda dengan pengertian database dalam Foxpro versi sebelumnya, yang berarti tabel. Dalam Visual Foxpro, database adalah sesuatu jenis file yang ekstensinya adalah .DBC. Database merupakan kumpulan tabel-tabel. Membuat tabel dalam database, mempunyai beberapa keuntungan, antara lain :

- Menentukan jenis kunci primary atau candidate
- Membuat relasi permanen (Persitant Relation) antar tabel
- Membuat Caption (judul) field untuk judul Grid
- Mendefinisikan nama panjang untuk field
- Menentukan nilai default field

- Menentukan Input Mask (Format masukkan field)
- Membuat Tingga
- Membuat prosedur tersimpan (Stored Procedure)
- Membuat File View
- Membuat aturan Level Field dan Level Record
- Memberikan komentar untuk tabel, field dan database.

### 5.3.1. Membuat Database

Untuk membuat database dalam project, dapat dilakukan 2 cara yaitu :

#### Membuat Database Lewat Project Manager

Untuk membuat database lewat project manager, lakukan :

- Di project manager, pilih halaman **Data > Database > New**

#### Membuat Database Lewat Perintah Visual Foxpro

Untuk membuat database lewat perintah visual foxpro, kita dapat mengetikkan perintah berikut pada jendela Command Window :

**Create Database [namadatabase]**

### 5.4. Tabel

Tabel merupakan tempat kita meletakkan record atau data. Nama file perluasan (Extention) file tabel adalah .DBF.

#### 5.4.1. Membuat Tabel

Cara membuat tabel ada dua cara, yaitu :

- **Lewat Project Manager**

Untuk membuat tabel lewat project, langkahnya adalah :

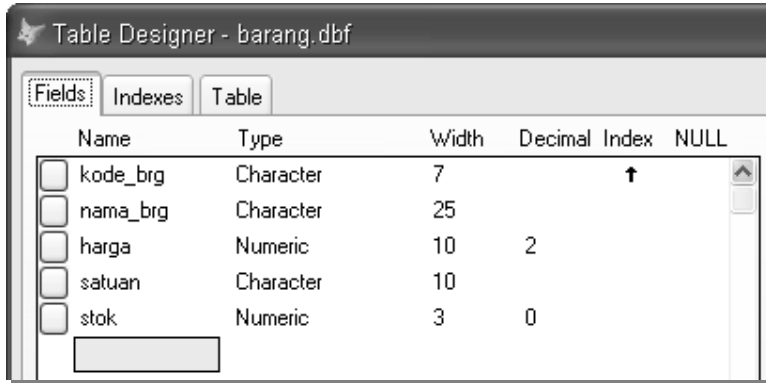
Pada Project Manager, pilih **Data | Free Table | New**

- **Lewat Perintah Visual Foxpro**

Untuk membuat tabel lewat perintah visual foxpro, ketikkan perintah berikut pada Command Window :

**Create [NamaTable]**

Maka dilayar akan Visual Foxpro akan tampil :



masukkanlah struktur tabel tersebut. Kemudian klik **OK**.

#### **Keterangan Structure Designer Dialog :**

Nama Kolom	Keterangan
Name	Menyatakan nama Field
Type	Menyatakan type data yang ditampung
Width	Menyatakan lebar/panjang untuk data field
Decimal	Menyatakan jumlah digit bagian pecahan.
Null	Bila muncul tanda checklist, dapat diberikan

#### **Aturan Penulisan Field :**

- Penulisan nama field, tidak boleh menggunakan spasi.
- Jangan menggunakan tanda koma (,), tanda pipe (|) dan tanda seru (!).

#### **Tipe-tipe Field :**

Type Field	Keterangan
<b>Character (C)</b>	Untuk menampung kumpulan karakter huruf, angka dan karakter ASCII yang dapat tercetak, lebar field tipe antara 1 sampai dengan 254 karakter.
<b>Currency</b>	Untuk menyimpan nilai mata uang dan berisi hingga 20 digit, termasuk +, -, dan . (tanda decimal)
<b>Float (F)</b>	Merupakan tipe data special dari field numeric yang didesain untuk data scientific
<b>Date</b>	Untuk menyimpan data tanggal memasukkan dengan format mm/dd/yy.
<b>DateTime</b>	Berisi nilai dari tanggal, waktu atau tanggal dan waktu nilai tersebut. Data nilai tersebut disimpan dengan format yy/mm/dd/hh/mm/ss
<b>Double</b>	Digunakan untuk melakukan perhitungan dengan nilai yang berisi tanda desimal (float)
<b>Integer</b>	Untuk menyimpan bilangan bulat 4-byte, value -2147483647 s/d 2147483647

<b>Logical</b>	Untuk menyimpan masukan True (T) atau False (F)
<b>Memo</b>	Untuk menyimpan sejumlah besar data, field memo dapat berisi tipe data apa saja.
<b>General</b>	Digunakan untuk menyimpan OLE (object Lingking and Embedding).
<b>Character (Binary)</b>	Digunakan untuk menampung banyaknya data karakter . dengan ukuran 1 byte per karakter s/d 254.
<b>Memo (binary)</b>	Digunakan untuk menampung banyaknya data memo dengan perawatan tanpa mengubah kode dengan ukuran 4 byte dalam tabel.

#### 5.4.2 Operasi pada Tabel

##### Mengaktifkan/membuka Tabel

Untuk mengaktifkan tabel, maka perintahnya : **USE <nama\_tabel>**

Misalnya kita akan mengaktifkan tabel barang, maka : **Use Barang**

##### Membuka/Memodifikasi Struktur Tabel

Pada saat tabel telah dibuat, jika terdapat kesalahan dalam pengetikan nama field, tipe data atau lebarnya dll, maka kita perlu memodifikasi struktur tabelnya tetapi harus diaktifkan dulu nama tabelnya. Sintaknya :

**USE <nama\_tabel>**                    <enter>

**MODIFY STRUCTURE**                    <enter>

##### Menampilkan struktur tabel ke layar

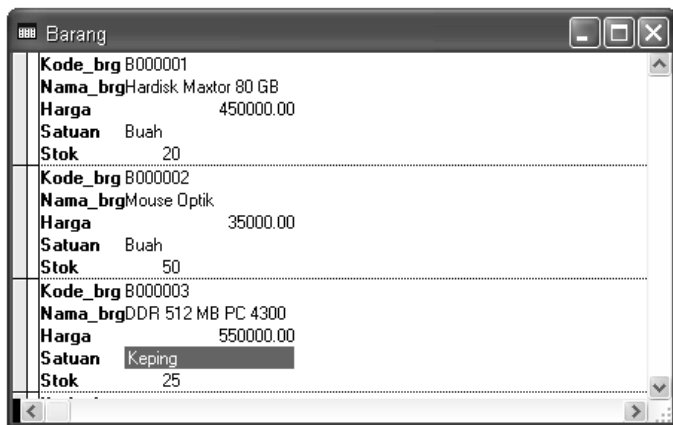
Untuk menampilkan struktur tabel yang aktif, maka ketikkan perintah :

**LIST STRUCTURE**

**DISPLAY STRUCTURE**

##### Mengisi Data Langsung kedalam Tabel

Untuk memasukkan data kedalam tabel, pertama pastikan tabel tersebut sudah aktif, kemudian ketikkan perintah : **APPEND** <enter>



untuk mengisi data tabel, tetapi data nya diambil dari tabel lain, langkahnya :

- Aktifkan tabel yang akan diisi :

**USE <nama\_tabel>**

- Kemudian ketikkan perintah.

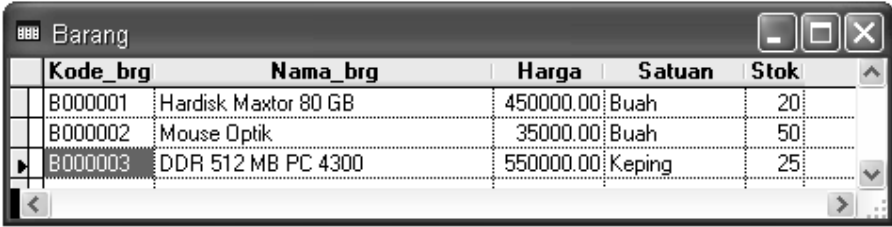
**APPEND FROM <nama\_tabel\_data> [FOR <kondisi>]**

### Melihat isi Tabel

Untuk melihat hasilnya bahwa data telah ditambahkan/dimasukkan, ketikkan perintah :

**BROWSE <enter>**

Maka akan muncul :



Kode_brg	Nama_brg	Harga	Satuan	Stok
B000001	Hardisk Maxtor 80 GB	450000.00	Buah	20
B000002	Mouse Optik	35000.00	Buah	50
B000003	DDR 512 MB PC 4300	550000.00	Keping	25

### Mengatur Letak Pointer

Ada beberapa perintah untuk mengatur letak pointer :

- Menuju record paling atas/awal : **GO TOP**
- Menuju record paling akhir/bawah : **GO BOTTOM**
- Menuju record tertentu : **GO <no\_record>**

### Menghapus data dalam Tabel

Untuk menghapus data pada tabel, menggunakan perintah : **DELETE**

Dalam penghapusan, ada beberapa cara/teknik yaitu :

- Menghapus data record ke 2, maka perintahnya :  
**GO 2**  
**DELETE**
- Menghapus data yang kode barangnya = B0003, maka perintahnya :  
**DELETE FOR kode\_brg='B000003'**

Untuk melihat hasilnya bahwa data telah dihapus, ketikkan perintah Browse

Ternyata data belum dihapus, tapi diberi **tanda hitam** disebelah kirinya. Karena perintah DELETE ini hanya memberi tanda pada data yang akan dihapus. Untuk melakukan penghapusan, maka ketikkan perintah :

**PACK <enter>**

Untuk melihat hasilnya bahwa data telah dihapus, ketikkan perintah Browse.



### Mengedit data tabel

Untuk mengedit/merubah data dalam tabel, kita bisa melakukan pengeditan langsung melalui tabel, atau menggunakan perintah :

**EDIT [FIELDS <field1>,<field2>,...] [FOR <kondisi>]**

### Mengganti data dalam Tabel

Untuk mengganti/update/ memasukkan data pada tabel, maka digunakan perintah :

**REPLACE ... WITH ...**

Misalnya mengganti nama barang untuk kode barang 'B001' menjadi "Hardisk", maka perintahnya :

**REPLACE nama\_brg WITH 'Hardisk Seagete 80 GB' FOR kode\_brg='B0000001'**

### Menutup Tabel yang sedang Aktif

Untuk menutup tabel yang sedang aktif, maka ketikkan perintah :

**CLOSE DATABASE <enter>**

Untuk menutup semua yang sedang aktif, maka ketikkan perintah :

**CLOSE ALL <enter>**

### Mengcopy tabel dan mengcopy struktur tabel

Untuk mengcopy tabel, pertama pastikan seluruh tabel telah di tutup/nonaktif. Kemudian ketikkan perintah :

**COPY FILE <nama\_tabel> TO <nama\_tabel\_baru> <enter>**

Dengan cara ini kita telah mengcopykan struktur dan isi data tabel tersebut. Untuk mengcopykan struktur tabelnya, maka langkahnya :

- Tutup semua tabel yang aktif

**CLOSE DATABASE**

- Aktifkan tabel yang akan di copy

**USE <nama\_tabel>**

- Kemudian ketikkan perintah :

**COPY STRUCTURE TO <nama\_tabel\_baru>**

# Modul 6

## Mengolah Data Dengan Form

### 6.1. Pendahuluan

Memasukkan data kedalam tabel dapat dilakukan dengan beberapa cara, diantaranya dengan langsung memasukkan kedalam tabel browse atau menggunakan Form. Form digunakan untuk mengolah data tabel. Kita dapat menambah, mengedit/merubah, menghapus atau mencari data melalui Form yang telah kita program/rancang dengan cepat dan mudah.

#### 1. Penyimpanan Data

Pemasukkan data pada form kemudian akan disimpan ke dalam database / tabel yang telah ditentukan. Proses penyimpanan tersebut pada saat kita melakukan event pada objek pada form, misalnya pada umumnya pada objek Command button. Sintak perintah untuk penyimpanan yaitu sebagai berikut :

**Replace** <field\_name> **With** <variable>

Misal Perintah dalam Pemasukkan Data Barang :

**Append Blank**

**Replace** kode\_brg **With** Thisform.text1.value

**Replace** Nm\_Brg **With** Thisform.text2.value

...

... Atau

**Append Blank**

**Replace** Kode\_brg **with** Thisform.text1.value, Nm\_Brg **With** Thisform.text2.value,

...

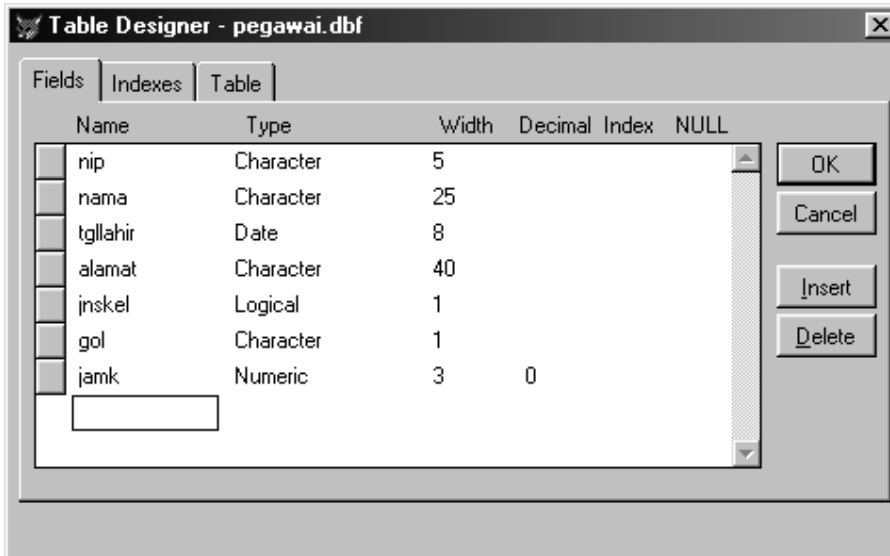
...

#### Catatan :

Perintah Append Blank digunakan apabila kita akan menambahkan data baru kedalam tabel. Append Blank berfungsi menambahkan record kosong pada tabel. Jika penyimpanan tersebut berfungsi untuk menggantikan/replace, maka perintah Append Blank tidak digunakan.

## TUGAS PRAKTIKUM XI - 1

1. Buatlah tabel dengan struktur dibawah ini dan simpan dengan nama **PEGAWAI.DBF**



kemudian masukkanlah data dibawah ini :

NIP	NAMA	ALAMAT	GOL	TGLLAHIR	JENISKE	JAMK
89001	LINA	JL. AHMAD YANI	2	12-02-1980	T	45
89002	ASEP	JL. ACEH RAYA	1	10-04-1981	F	40
89003	BUDI	JL. R.E. MARTHADINATA	3	11-05-1982	F	45

2. Setelah data tersebut dimasukkan, kemudian buatlah desain Form dibawah ini dan simpan dengan nama **FORM6.SCX**.

Aturlah Properties Objek Form seperti dibawah ini:

OBJECT	PROPERTIES	NILAI/VALUE
Form1	Caption	PT. CREATIVE COMPUTER TASIKMALAYA
	AutoCenter	.T. – True
Label1	Caption	PEMASUKKAN DATA PEGAWAI
	AutoSize	.T. – True
	FontSize	11
Label2	FontBold	.T. – True
	Caption	NIP
Label3	AutoSize	.T. – True
	Caption	NAMA PEGAWAI
Label4	AutoSize	.T. – True
	Caption	TANGGAL LAHIR
Label5	AutoSize	.T. – True
	Caption	ALAMAT
Label6	AutoSize	.T. – True
	Caption	GOLONGAN
Label7	AutoSize	.T. – True
	Caption	JENIS KELAMIN
Label8	AutoSize	.T. – True
	Caption	JAM KERJA

Combo1	RowSource	1,2,3
	RowSourceType	1 - Value
	Style	2 – DropDown List
Command1	Caption	\<BROWSE
Command2	Caption	\<SIMPAN
Command3	Caption	\<KELUAR

Object OptionGroup1 :

Option1	Caption	LAKI – LAKI
	AutoSize	.T. – True
Option2	Caption	PEREAMPUAN
	AutoSize	.T. – True

Ketikkan Procedure/Event pada Object dibawah ini :

▪ **Form1** Procedure **Init**

```

Thisform.text1.Value=""
Thisform.text2.Value=""
Thisform.text3.Value=DATE()
Thisform.text4.Value=""
Thisform.text5.Value=0
Thisform.combo1.Value=""
Thisform.optiongroup1.Value = 1
thisform.text1.SetFocus
    
```

▪ **Form1** Procedure **Load**

```

CLOSE DATABASE
SET TALK OFF
SET ECHO OFF

USE pegawai
    
```

- **Form1** Procedure **Unload**

*Close Database*

- **Command1** Procedure **Click**

*BROWSE*

- **Command2** Procedure **Click**

*APPEND BLANK*

*replace nip WITH thisform.text1.Value*

*replace nama WITH thisform.text2.Value*

*replace tgllahir WITH thisform.text3.Value*

*replace alamat WITH thisform.text4.value*

*replace gol WITH thisform.combo1.Value*

*IF thisform.optiongroup1.Value = 1*

*replace jnske WITH .T.*

*ELSE*

*replace jnske WITH .F.*

*ENDIF*

*replace jamk WITH thisform.text5.Value*

*thisform.Init*

- **Command3** Procedure **Click**

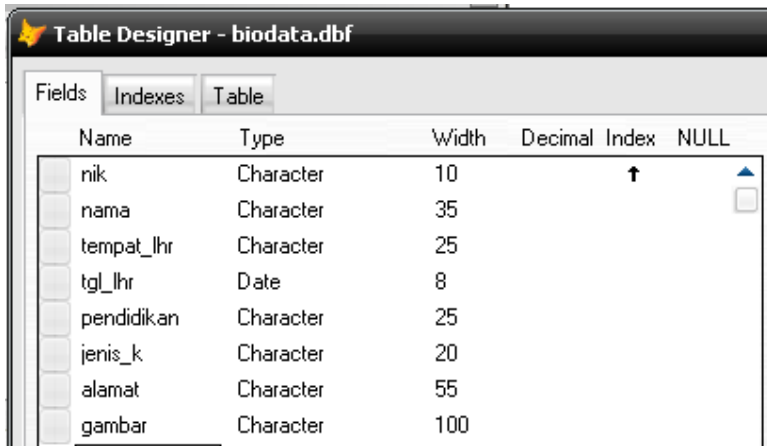
*Thisform.Release()*

3. Kemudian masukkan data-data dibawah ini dengan menggunakan form tersebut.

NIP	NAMA	ALAMAT	GOL	TGLLAHIR	JNSKE	JAMK
89004	DONI K	JL. HASANUDDIN	1	22-06-1979	F	40
89005	IRMAN	JL. LENGKONG	1	07-07-1983	F	40
89006	DELLY	JL. PURBARATU	3	12-07-1979	T	42
89007	SANDRA	JL. LENGKONG	3	13-05-1980	T	43
89008	INDRA L	JL. PANCASILA	1	01-06-1980	F	40
89009	MAWAR	JL. AHMAD YANI	2	12-11-1980	T	40
89010	INDAH P	JL. KEBANGSAAN	1	01-01-1979	T	42
89011	GOGON	JL. PATIMURA	3	02-05-1980	F	45
89012	BAHRUN	JL. MOH HATTA	2	14-09-1981	F	43
89013	VIRA	JL. CICAHEUM	1	18-03-1983	T	40

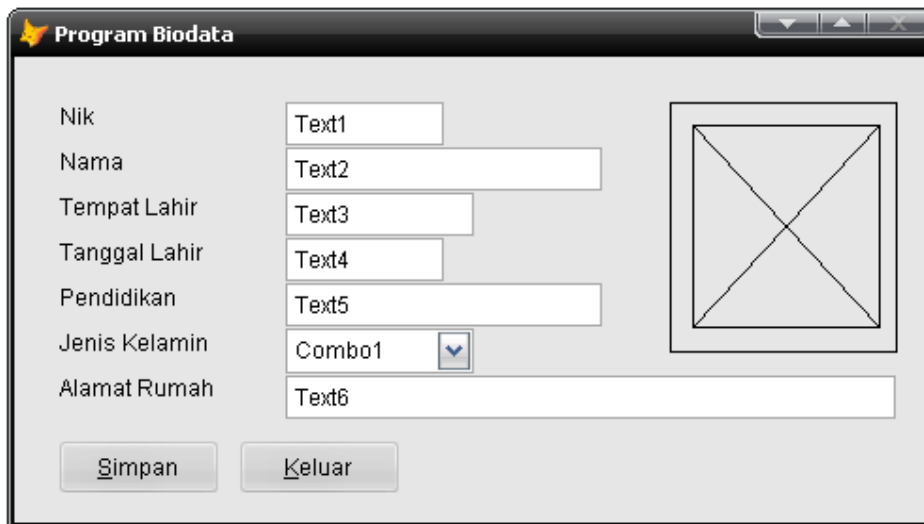
## TUGAS PRAKTIKUM XI - 2

1. Buatlah tabel dengan biodata seperti dibawah ini dan simpan dengan nama **Biodata.DBF**



Name	Type	Width	Decimal	Index	NULL
nik	Character	10		↑	<input type="checkbox"/>
nama	Character	35			<input type="checkbox"/>
tempat_lhr	Character	25			<input type="checkbox"/>
tgl_lhr	Date	8			<input type="checkbox"/>
pendidikan	Character	25			<input type="checkbox"/>
jenis_k	Character	20			<input type="checkbox"/>
alamat	Character	55			<input type="checkbox"/>
gambar	Character	100			<input type="checkbox"/>

2. Kemudian buat program input, simpan dengan nama **Biodata.SCX**. Dengan layout seperti dibawah ini :



The screenshot shows a form titled "Program Biodata" with the following fields and controls:

- Nik: Text1
- Nama: Text2
- Tempat Lahir: Text3
- Tanggal Lahir: Text4
- Pendidikan: Text5
- Jenis Kelamin: Combo1 (dropdown menu)
- Alamat Rumah: Text6
- Buttons: Simpan (Save), Keluar (Exit)
- A placeholder box for a picture (gambar) with a large 'X' over it.

Aturlah Properties Objek-objek :

Nama Objek	Properties	Value
Form1	Autocenter	.T.
	Caption	Program Biodata
Label1	Caption	Nik
Label2	Caption	Nama
Label3	Caption	Tempat Lahir
Label4	Caption	Tanggal Lahir
Label5	Caption	Pendidikan
Label6	Caption	Jenis Kelamin
Label7	Caption	Alamat Rumah
Combo1	RowSourceType	1 - Value
	RowSource	Laki-laki, Perempuan
	Style	2 - DropDown List
Command1	Caption	\<Simpan
Command2	Caption	\<Keluar

Kemudian isikan kode program sesuai dengan objek dan event yang diperlukan, antara lain:

- Object **Form1** Procedure **Load Event**  

```
CLOSE DATABASES
USE biodata
```
- Object **Form1** Procedure **Init Event**  

```
WITH thisform
  .text1.value = ''
  .text2.value = ''
  .text3.value = ''
  .text4.value = CTOD('')
  .text5.value = ''
  .combo1.value = ''
  .text6.value = ''
  .image1.picture = ''
  .text1.setfocus
ENDWITH
```
- Object **Image1** Procedure **Click Event**  

```
LOCAL nFile
nFile=''
nFile=GETPICT('BMP','Nama File','Buka')
thisform.image1.Picture = nFile
```
- Object **Command1** Procedure **Click Event**  

```
WITH thisform
  APPEND BLANK
  replace nik      WITH .text1.value
  replace nama     WITH .text2.value
  replace tempat_lhr WITH .text3.value
  replace tgl_lhr  WITH .text4.value
  replace pendidikan WITH .text5.value
  replace jenis_k  WITH .combo1.value
  replace alamat   WITH .text6.value
  replace gambar   WITH .image1.picture
  .refresh()
  .init()
ENDWITH
```
- Object **Command2** Procedure **Click Event**  

```
Thisform.release
```

# Modul 7

## Index & Searching

### 7.1. Pendahuluan

File Indeks merupakan file berisi data atau record yang terurut berdasarkan field tertentu. Kegunaan utama file indeks adalah untuk mencari data tertentu berdasarkan field yang diketahui, dan untuk pengurutan di laporan (report).

### 7.2. Jenis File Indeks

Dilihat dari jenis indeksnya, file indeks dalam Visual foxpro terdiri dari beberapa jenis :

#### a. File Indeks Primer (Primary Index)

File indeks primer atau file index utama merupakan file indeks yang tidak memperbolehkan duplikasi data kunci indeks, biasanya dipakai untuk relasi pada tabel primer (tabel satu).

#### b. File Indeks Kandidat (Candidate Index)

File indeks kandidat merupakan file indeks seperti file indeks primer, tetapi dalam satu tabel dapat memiliki lebih dari satu file indeks kandidat.

#### c. File Indeks Reguler (Reguler Index)

File indeks reguler merupakan file indeks yang memperbolehkan duplikasi data, biasanya dipakai pada tabel sekunder (tabel banyak).

#### d. File Indeks Unik (Unique Index)

File indeks unik merupakan file indeks yang tidak sesuai dengan nama. File indeks ini seperti file indeks reguler, tetapi hanya mencatat nilai kunci pertama dalam file indeks.

### 7.3. Membuat File Indeks

Untuk membuat file index ada beberapa cara yaitu :

- Membuat File Indeks Lewat Table Designer

Untuk membuat file indeks lewat Table Designer, yaitu dengan cara berikut. Pada **Table Designer**, pilih halaman **Indexes**.

- Membuat File Indeks Lewat Perintah Visual Foxpro

Membuat file indeks lewat perintah, untuk itu indeks jenis .IDX, sintaknya adalah :

```
Index On <field_kunci> To <FileIndeks> [Ascending|Descending]
```

*Contoh :*

Use Siswa

Index On Nis To SiswaIdx



#### 7.4. Membuka File Indeks

Dalam Visual foxpro untuk membuka file indeks dapat dilakukan beberapa cara yaitu :

- Membuka File Indeks Lewat Menu
  1. Pada menu, pilih **File > Open**
  2. Pilih **File of type** (IDX, \*.CDX), akan tampil file-file indeks
  3. Klik nama file indeks, kemudian klik **Ok**
- Membuka File Index Lewat Perintah

Untuk membuka file indeks lewat perintah, bentuk umum perintahnya adalah :

```
Set Index To <FileIndeksList>
```

Atau

```
Use <TableMaster> Index <FileIndeks>
```

Contoh :

Set Index To Siswaldx [atau]

Use Siswa Index Siswaldx

#### 7.5. Memilih File Indeks

Untuk menentukan file indeks yang aktif pada jenis file indeks, gunakan perintah :

```
Set Order To <indexNumber>
```

#### 7.6. Mereindeks Data atau Record

Jika sebuah file indeks sudah tidak sesuai dengan kondisi terbaru, kita dapat mereindeks ulang (reindeks). Ada beberapa cara untuk mereindeks.

**Mereindeks Data Lewat Table Designer :**

- Buka tabel, kemudian pilih menu **View > Browse**
- Pilih **Table > Rebuild Indexes**

**Mereindeks Data Lewat Perintah Visual Foxpro :**

Untuk mereindeks data lewat perintah, gunakan perintah seperti contoh dibawah ini :

```
Use Siswa
```

```
Reindex
```

### 7.7. Pencarian Data

Data yang kita inputkan/masukkan terkadang ingin kita lihat kembali untuk mengetahui informasi data tersebut. Misalnya pada data pegawai, kita ingin mengetahui alamat dari pegawai yang mempunyai NIP = "98012". Cara pertama kita bisa melihatnya langsung ke tabel browse. Tetapi hal tersebut bisa kita lakukan mungkin untuk data yang sedikit, untuk data yang banyak, kita memerlukan waktu yang agak lama untuk pencariannya.

Cara yang cepat yaitu kita membuat Form dan merancang program untuk pencarian data tersebut. Proses pencarian data kita dapat menggunakan perintah :

- **LOCATE <Expresi-Logika>**
- **SEEK <var>**

Misal untuk dalam pencarian data barang :

Menggunakan Perintah **LOCATE**

**Locate for** kd\_brg = Thisform.text1.value

**If Found()**

\*Jika ditemukan

...

**Else**

\*Jika tidak ditemukan

...

**Endif**

Menggunakan Perintah **SEEK**

Kode = Thisform.Text1.Value

**Seek** Kode

**If Found()**

\*Jika ditemukan

...

**Else**

\*Jika tidak ditemukan

...

**Endif**

## TUGAS PRAKTIKUM XII

1. Buatlah Form dibawah ini yang digunakan untuk mengedit/merubah data dan menghapus data dari hasil pencarian berdasarkan NIP. Simpan form tersebut dengan nama **FORM7.SCX**.

Aturlah Properties Objek Form seperti dibawah ini:

OBJECT	PROPERTIES	NILAI/VALUE
Form1	Caption	PT. CREATIVE COMPUTER TASIKMALAYA
	AutoCenter	.T. – True
Label1	Caption	PENCARIAN DATA PEGAWAI
	AutoSize	.T. – True
	FontSize	11
	FontBold	.T. – True
Label2	Caption	NIP
	AutoSize	.T. – True
Label3	Caption	NAMA PEGAWAI
	AutoSize	.T. – True
Label4	Caption	TANGGAL LAHIR
	AutoSize	.T. – True
Label5	Caption	ALAMAT
	AutoSize	.T. – True
Label6	Caption	GOLONGAN
	AutoSize	.T. – True
Label7	Caption	JENIS KELAMIN
	AutoSize	.T. – True
Label8	Caption	JAM KERJA
	AutoSize	.T. – True
Combo1	RowSource	1,2,3
	RowSourceType	1 - Value
	Style	2 – DropDown List
Command1	Caption	\<CARI NIP
Command2	Caption	\<EDIT DATA
Command3	Caption	\<HAPUS DATA
Command4	Caption	\<KELUAR

Object OptionGroup1 :

Option1	Caption	LAKI – LAKI
	AutoSize	.T. – True
Option2	Caption	PEREMPUAN
	AutoSize	.T. – True

Ketikkan Procedure/Event pada Object dibawah ini :

▪ **Form1 Procedure Init**

*\*MENGOSONGKAN SEMUA OBJEK FORM*

*Thisform.text1.Value=""*

*Thisform.text2.Value=""*

*Thisform.text3.Value=DATE()*

*Thisform.text4.Value=""*

*Thisform.text5.Value=0*

*Thisform.combo1.Value=""*

*Thisform.optiongroup1.Value = 1*

*\*MENONAKTIFKAN OBJEK FORM KECUALI TEXT1*

*Thisform.text2.Enabled = .F.*

*Thisform.text3.Enabled = .F.*

*Thisform.text4.Enabled = .F.*

*Thisform.text5.Enabled = .F.*

*Thisform.combo1.Enabled = .F.*

*Thisform.optiongroup1.Enabled = .F.*

*\*MENGAKTIFKAN CURSOR DI TEXT1*

*thisform.text1.SetFocus*

▪ **Form1 Procedure Load**

*CLOSE DATABASE*

*SET TALK OFF*

*SET ECHO OFF*

*SET SAFETY OFF*

*USE pegawai*

*INDEX ON nip TO pegawai*

*SET INDEX TO pegawai*

▪ **Form1 Procedure Unload**

*Close Database*

- **Command1** Procedure **Click**

```

vNIP = ALLTRIM(Thisform.text1.value)
SEEK vNIP
IF FOUND()
    *jika diketemukan, Tampilkan data
    thisform.text1.Value = nip
    thisform.text2.Value = nama
    thisform.text3.Value = tgllahir
    thisform.text4.value = alamat
    thisform.combo1.Value = gol
    IF jnske
        thisform.optiongroup1.Value =2
    ELSE
        thisform.optiongroup1.Value =1
    ENDIF
    Thisform.text5.Value = jamk
ELSE
    *Jika tidak diketemukan, tampilkan pesan
    MESSAGEBOX("Data Tidak Diketemukan !")
    Thisform.init
    thisform.text1.SetFocus()
ENDIF

```

- **Command2** Procedure **Click**

```

IF Thisform.Command1.Caption = "\<EDIT DATA"
    Thisform.Command1.Caption = "\<SIMPAN"
    Thisform.Command4.Caption = "\<BATAL"
    *MENGAKTIFKAN OBJEK FORM KECUALI TEXT1
    Thisform.text2.Enabled = .T.
    Thisform.text3.Enabled = .T.
    Thisform.text4.Enabled = .T.
    Thisform.text5.Enabled = .T.
    Thisform.combo1.Enabled = .T.
    Thisform.optiongroup1.Enabled = .T.

    *MENGAKTIFKAN CURSOR DI TEXT2
    thisform.text2.SetFocus
ELSE

```

```

IF      Thisform.Command1.Caption = "\<SIMPAN"
        Thisform.Command1.caption = "\<EDIT DATA"
        Thisform.Command4.Caption = "\<HAPUS DATA"
        *merubah data pegawai berdasarkan hasil pencarian
        replace nip    WITH thisform.text1.Value
        replace nama   WITH thisform.text2.Value
        replace tgllahir WITH thisform.text3.Value
        replace alamat    WITH thisform.text4.value
        replace gol     WITH thisform.combo1.Value
        IF thisform.optiongroup1.Value = 1
            replace jnske WITH .T.
        ELSE
            replace jnske WITH .F.
        ENDIF
        replace jamk WITH thisform.text5.Value
        Thisform.Init
    ENDIF
ENDIF

```

- **Command3** Procedure **Click**

```

Tombol = Thisform.command4.Caption
IF      Tombol = "\<HAPUS DATA"
        DELETE
        PACK
    ELSE
        Thisform.command1.Caption = "\<EDIT DATA"
    ENDIF
    Thisform.Init

```

- **Command4** Procedure **Click**

```

Thisform.Release()

```


# Modul 8

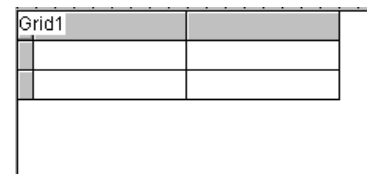
## Object Form II

### 8.1 MENGGUNAKAN OBJECT GRID, LISTBOX DAN PAGEFRAME

#### 8.1.1. Object Grid

Object Grid merupakan salah satu object dalam pembuatan suatu program aplikasi yang dapat digunakan untuk menampilkan data dari tabel atau juga dapat digunakan dalam pemasukkan data kedalam tabel. Cara pembuatan Object Grid :

- Aktifkan/klik object Grid pada tollbal Form Controls → 
- Kemudian Drag & Drop pada form designer, sehingga tampilannya seperti disamping :



- Kemudian aturlah beberapa properties Grid diantaranya:


Properties	Function
AllowAddNew	Untuk mengaktifkan grid supaya dapat menambah record. [.T. Or .F.]
ColumnCount	Menentukan/membuat jumlah kolom yang diperlukan didalam grid.
DeleteMark	Untuk menampilkan atau menghilangkan tanda pada saat penghapusan (DELETE) pada Grid. [.T. Or .F.]
GridLine	Untuk mengatur line/garis pembatas baris/kolom.
ReadOnly	Mengatur agar data didalam Grid hanya dapat dibaca.
RecordMark	Untuk menampilkan atau menghilangkan tanda pointer didalam Grid.
RecordSource	Menentukan sumber data/tabel yang akan ditampilkan didalam Grid.
RecordSourceType	Menentukan jenis pemanggilan sumber data tabel.

Setelah pengaturan Object Grid, kemudian atur juga properties tiap-tiap kolom didalam grid diantaranya Caption Header tiap-tiap column yang ada didalam Grid.

#### 8.1.2. Object ListBox

Object ListBox merupakan object yang fungsinya hampir sama dengan object Grid yaitu dapat menampilkan data dari tabel atau menampilkan data yang kita atur sendiri pada properties RowSource dan RowSourceType. Tetapi object ListBox ini tidak dapat berfungsi sebagai input data kedalam tabel seperti halnya object Grid.

Cara pembuatan Object ListBox :

- a. Aktifkan/klik Toolbar ListBox pada Toolbar Form Controls → 
- b. Kemudian Drag & Drop Object tersebut pada Form Designer sehingga akan tampil object ListBox pada Form seperti gambar :




- c. Aturlah Propertiesnya.

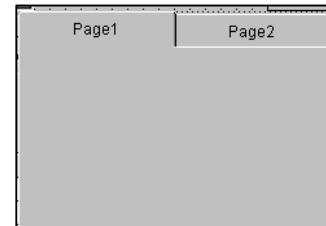
Properties	Function
RowSource	Menentukan Sumber Data
RowSourceType	Menentukan/mengatur jenis/tipe data tersebut ditampilkan kedalam object ListBox.

### 8.1.3. Object PageFrame

Object PageFrame merupakan object yang berfungsi untuk membagi tampilan menjadi beberapa frame/bagian. Jadi dalam suatu form dapat dibuat menjadi beberapa page/halaman yang terpisah menggunakan object PageFrame.

Langkah pembuatan PageFrame :

- a. Aktifkan/Klik object PageFrame pada Toolbar Form Controls → 
- b. Kemudian Drag & Drop object tersebut pada Form Designer sehingga tampilan object tersebut seperti gambar.



- c. Aturlah beberapa propertiesnya.

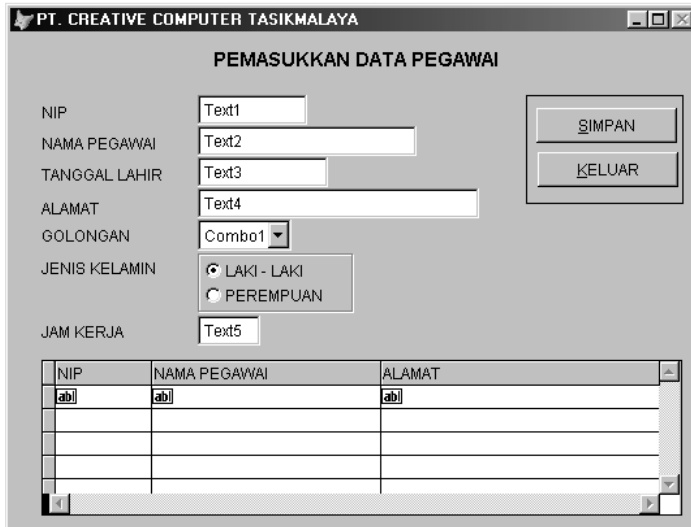
Properties	Function
PageCount	Menentukan jumlah Page/halaman yang diperlukan
TabStyle	Mengatur bentuk tampilan perataan Tab/frame

Kemudian atur juga properties Tiap Page dalam PageFrame tersebut diantaranya Caption untuk tiap – tiap Page.



**TUGAS PRAKTIKUM XIII**

1. Pindah kan drive kerja ke drive kerja masing-masing dengan perintah SET DEFAULT TO.
2. Bukalah kembali form **FORM6.SCX**. Kemudian aturlah kembali tampilan form tersebut dan tambahkan object Grid sehingga tampilannya seperti dibawah ini :



Aturlah Properties Objek Grid seperti dibawah ini:

OBJECT	PROPERTIES	NILAI/VALUE
Grid1	ColumnCount	7
	DeleteMark	.F. – False
	ReadOnly	.T. – True
	RecordSource	Pegawai
	RecordSourceType	2 – Prompt

Tambahkan isi Procedure/Event pada Object dibawah ini :

- **Command1** Procedure **Click (Tombol simpan)**

```

APPEND BLANK
replace nip WITH thisform.text1.Value
replace nama WITH thisform.text2.Value
replace tglahir WITH thisform.text3.Value
replace alamat WITH thisform.text4.value
replace gol WITH thisform.combo1.Value
IF thisform.optiongroup1.Value =2
    replace jeniske WITH .T.
ELSE

    replace jeniske WITH .F.
ENDIF
replace jamk WITH thisform.text5.Value
Thisform.Grid1.Refresh
thisform.Init
    
```

- **Grid1** Procedure **AfterRowColChange**

```

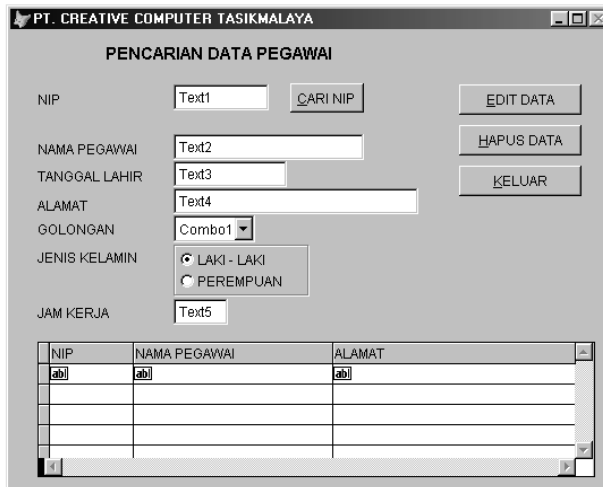
with thisform
    .text1.value=nip
    .text2.value=nama
    
```

```
.text3.value=tgllahir
.text4.value=alamat
.combo1.value=gol
if jnske
    .optiongroup1.value=2
else
    .optiongroup1.value=1
endif
.text5.value=jamk
Endwith
```

3. Setelah Form tersebut selesai, jalankan Form dan lihatlah hasil didalam Grid.
4. Simpan kembali Form tersebut .

**TUGAS PRAKTIKUM XIX**

1. Bukalah kembali Form **FORM7.SCX**, kemudian tambahkan Object grid pada form tersebut sehingga tampilan form tersebut seperti dibawah ini :



Aturlah Properties Objek Grid seperti dibawah ini:

OBJECT	PROPERTIES	NILAI /VALUE
Grid1	ColumnCount	7
	DeleteMark	.F. – False
	ReadOnly	.T. – True
	RecordSource	Pegawai
	RecordSourceType	2 – Prompt

Tambahkan isi Procedure/Event pada Object dibawah ini :

- **Command1** Procedure **Click**  
*Thisform.grid1.refresh*
- **Command2** Procedure **Click**  
*Thisform.grid1.refresh*

2. Jalankan Form tersebut, kemudian lihatlah hasilnya didalam grid tersebut.
3. Simpan kembali Form tersebut.

# Modul 9

## Bekerja Dengan Multi Table

### 9.1. Pendahuluan

Dalam database yang kompleks dan besar, penggunaan file tabel data sering dipakai lebih dari satu tabel. Sementara secara standar file tabel hanya bisa digunakan satu buah pada saat yang sama. Untuk menggunakan lebih dari satu tabel pada waktu yang bersamaan digunakan perintah **SELECT**.

Bentuk dari perintah SELECT ini adalah :

```
SELECT <daerah-kerja>
```

Contoh : untuk membuat program yang menggunakan 2 file data GAJI.DBF dengan PEGAWAI.DBF, kita dapat melakukan perintah dibawah ini :

<i>SELECT 1</i>		<i>SELECT A</i>
<i>USE Pegawai</i>	Atau	<i>USE Pegawai</i>
<i>SELECT 2</i>		<i>SELECT B</i>
<i>USE Gaji</i>		<i>USE Gaji</i>

### Perintah Untuk Menangani Pengolahan Banyak File

File-file yang telah dibuka di daerah kerja masing-masing/berbeda dapat dihubungkan satu sama lain dengan menggunakan perintah khusus, diantaranya perintah :

```
SET RELATION TO
```

### Perintah SET RELATION TO

Perintah ini digunakan untuk menghubungkan / mengaitkan file tabel database aktif dengan file database yang berada di daerah kerja lain, berdasarkan suatu Field kunci tertentu.

Sintaknya :

```
SET RELATION TO <Field-kunci> INTO <Alias>
```

Catatan :

- Field kunci harus terdapat dalam kedua file tersebut.
- Field kunci yang dikaitkan harus telah di Index berdasarkan Field kunci.

## TUGAS PRAKTIKUM XX

1. Pindah kan drive kerja ke drive kerja masing-masing dengan perintah SET DEFAULT TO.
2. Buatlah tabel dibawah ini untuk menampung data transaksi penjualan barang. Dengan struktur seperti dibawah ini :

**TRANSAKSI.DBF**

Field Name	Type	Width	Decimal
No_tran	Character	5	-
Tgl_tran	Date	8	-
Kode_lang	Character	5	-
Total_byr	Numeric	10	0

**DETILTRAN.DBF**

Field Name	Type	Width	Decimal
No_tran	Character	5	-
Kode_brg	Character	5	-
Jml_beli	Numeric	3	0

3. Buatlah form untuk pengolahan data transaksi penjualan barang. Simpan form tersebut dengan nama **TRANSAK.SCX**. Kemudian aturlah kembali tampilan form tersebut dan tambahkan object Grid sehingga tampilannya seperti dibawah ini :

Aturlah Properties :

Object	Properties	Value
Form1	Autocenter	.T.
	BorderStyle	2 – Fixed Dialog
	Caption	Transaksi Penjualan Barang
Text1	Maxlength	5
Text3	Maxlength	5
Text4	ReadOnly	.T.
Edit1	ReadOnly	.T.
Text5	ReadOnly	.T.
Text7	ReadOnly	.T.
Grid1	AllowAddNew	.T.
	DeleteMark	.F.
	RecordSource	Temp
	RecordSourceType	1 – Alias
	ColumnCount	5
Grid1.Column1.Header1	Caption	Kode Barang
Grid1.Column2.Header1	Caption	Nama Barang
Grid1.Column2.Header1.Text1	Enabled	.F.
Grid1.Column3.Header1	Caption	Harga
Grid1.Column3.Header1.Text1	Enabled	.F.
Grid1.Column4.Header1	Caption	Jml Beli
Grid1.Column5.Header1	Caption	Jml Bayar
Grid1.Column5.Header1.Text1	Enabled	.F.
Command1	Caption	\<Simpan
Command2	Caption	\<Keluar

Ketikkan listing procedure dibawah ini :

### Form1.Load

```
SET TALK OFF
```

```
SET ECHO OFF
```

```
SET SAFETY OFF
```

```
CLOSE DATABASES
```

```
CREATE TABLE temp.dbf (kode_brg c(5), nama_brg c(25), harga n(10), jml_beli n(3),;
                        jml_byr n(10))
```

```
USE temp
```

```
APPEND BLANK
```

```
CLOSE DATABASES
```

```
SELECT 1
```

```
USE barang
```

```
INDEX ON kode_brg TO barang
```

```
USE barang INDEX barang
```

```
SELECT 2
```

```
USE pelanggan
```

```
INDEX ON kode_lang TO pelanggan
```

```
USE pelanggan INDEX pelanggan
```

```
SELECT 3
use transaksi
INDEX ON no_tran TO transaksi
USE transaksi INDEX transaksi
```

```
SELECT 4
USE detiltran ALIAS detiltran
```

```
SELECT 20
USE temp ALIAS temp
```

### Form1.Init

```
WITH thisform
    .text1.value=""
    .text2.value=DATE()
    .text3.value=""
    .text4.value=""
    .edit1.value=""
    .text5.value=0
    .text6.value=0
    .text7.value=0

    SELECT temp
    ZAP
    APPEND BLANK

    .grid1.refresh
    .text1.setfocus
ENDWITH
```

### Form1.Unload

```
CLOSE DATABASES
DELETE FILE temp.dbf
```

### Text1.Lostfocus

```
WITH thisform
    IF ! EMPTY(.text1.value)
        SELECT 3
        SEEK ALLTRIM(.text1.value)
        IF FOUND()
            MESSAGEBOX("No.Transaksi sudah ada !",0+16,"Perhatian")
            .text1.setfocus
```

```
ENDIF  
ENDIF  
ENDWITH
```

### **Text3.Lostfocus**

```
WITH thisform  
    IF ! EMPTY(.text3.value)  
        SELECT 2  
        SEEK ALLTRIM(.text3.value)  
        IF FOUND()  
            .text4.value=nama_lang  
            .edit1.value=alamat  
        ELSE  
            MESSAGEBOX("Kode Pelanggan tidak ada!",0+16,"Perhatian")  
            .text3.setfocus()  
        ENDIF  
    ENDIF  
ENDWITH
```

### **Grid1.Column1.Text1.Lostfocus**

```
WITH thisform  
IF ! EMPTY(.grid1.column1.text1.value)  
    SELECT 1  
    SEEK ALLTRIM(.grid1.column1.text1.value)  
    IF FOUND()  
        .grid1.column2.text1.value=nama_brg  
        .grid1.column3.text1.value=harga  
    ELSE  
        MESSAGEBOX("Kode brg tidak ada !",0+16,"Perhatian")  
        .grid1.column1.text1.setfocus  
    ENDIF  
ENDIF  
ENDWITH
```

### **Grid1.Column1.Text1.KeyPress**

```
LPARAMETERS nKeyCode, nShiftAltCtrl  
*jika tekan tombol F5  
IF nKeyCode=-4  
    DO FORM listbrg  
ENDIF
```

**Grid1.Column4.text1.InteractiveChange**

WITH thisform

```
.grid1.column5.text1.value=.grid1.column3.text1.value*.grid1.column4.text1.value
```

```
SUM temp.jml_byr TO .text5.value
```

ENDWITH

**Text6.InteractiveChange**

WITH thisform

```
.text7.value=.text6.value-.text5.value
```

ENDWITH

**Command1.Click**

WITH thisform

```
IF ! EMPTY(.text1.value)
```

```
  SELECT 3
```

```
  *cari no.tran pada tabel transaksi
```

```
  SEEK ALLTRIM(.text1.value)
```

```
  IF      ! FOUND()
```

```
    *simpan ke tabel transaksi
```

```
      APPEND BLANK
```

```
      replace no_tran  WITH .text1.value
```

```
      replace tgl_tran WITH .text2.value
```

```
      replace kode_lang WITH .text3.value
```

```
      replace total_byr WITH .text5.value
```

```
    *simpan data pada grid ke tabel detiltran
```

```
      SELECT temp
```

```
      GO top
```

```
      DO while ! EOF()
```

```
        IF ! EMPTY(temp.kode_brg)
```

```
          SELECT 4
```

```
          APPEND BLANK
```

```
          replace no_tran  WITH .text1.value
```

```
          replace kode_brg WITH temp.kode_brg
```

```
          replace jml_beli WITH temp.jml_beli
```

```
        SELECT 1
```

```
        SEEK ALLTRIM(temp.kode_brg)
```

```
        replace stok WITH stok-temp.jml_beli
```



```

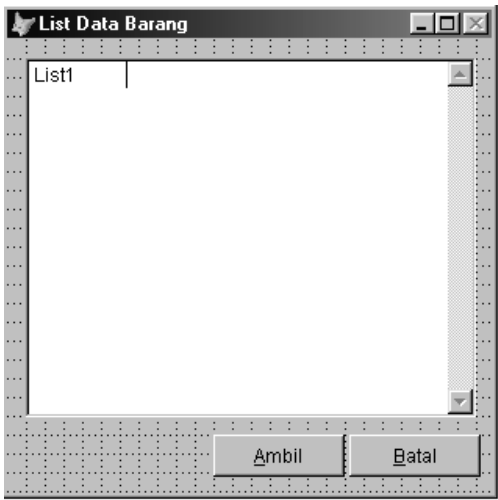
        ENDIF
        SELECT temp
        skip
    ENDDO
ENDIF
ELSE
    MESSAGEBOX("Masukkan No.Transaksi!",0+16,"Perhatian")
    .text1.setfocus
ENDIF
.init
ENDWITH

```

**Command2.Click**

```
Thisform.release
```

4. Buatlah Form seperti dibawah ini untuk menampilkan list data barang, kemudian simpan dengan nama **LISTBRG.SCX**

Aturlah Properties **List1** :

```

ColumnCount      = 2
Rowsource        = Barang
RowSourceType     = 2 - Alias

```

Ketikkan Listing Procedure :**List1.DbClick dan Command1.Click**

```

transak.grid1.column1.text1.value=barang.kode_brg
transak.grid1.column1.text1.lostfocus
thisform.Release

```

**Command2.Click**

```
Thisform.release
```

# Modul 10

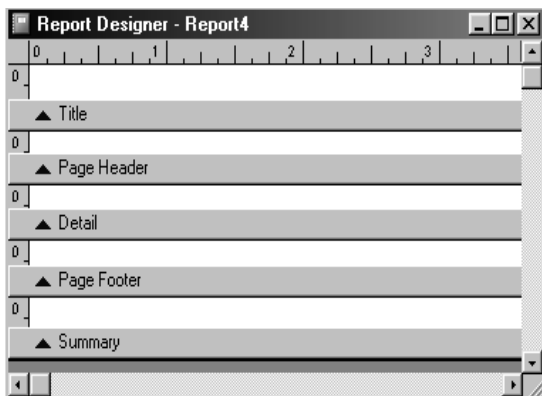
## Membuat Laporan

### 10.1. Pendahuluan

Hampir 70 persen konsentrasi pengembang (programmer) dalam membuat atau mengembangkan sistem atau program adalah pembuatan laporan (report). Tujuan akhir sistem atau program hampir dapat dipastikan adalah laporan (pencetakan) ke kertas. Dalam pembuatan report, pengembang harus membuat report mendekati bentuk laporan manual. Pengaturan printer yang dimilikinya juga membutuhkan waktu yang banyak, untuk itulah programmer harus berkonsentrasi pada laporan.

### 10.2. Bentuk Laporan

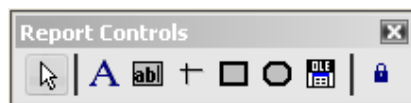
Untuk memahami cara membuat laporan (report) di Visual Foxpro, kita harus memahami bentuk laporan (layout), yaitu sebagai berikut :








- **Title**, merupakan judul laporan, yang akan muncul pada awal laporan.
- **Page Header**, Merupakan judul halaman, yang dipakai untuk judul field-field (atribut) pada laporan.
- **Group Header**, merupakan judul group, ditampilkan untuk mengklasifikasi record, berdasarkan group tertentu pada field.
- **Detail**, merupakan bagian untuk menampilkan record data yang berulang.
- **Group Footer**, dipakai untuk meletakkan subtotal group record tertentu dan letaknya dibawah.
- **Page Footer**, merupakan catatan di bawah halaman, ditampilkan sekali akhir halaman.
- **Summary**, merupakan ringkasan yang akan ditampilkan diakhir report.

### 10.3. Objek-Objek Pembentuk Laporan

Dalam pembuatan report di Visual Foxpro, kita banyak menggunakan objek-objek pembentuk report. Kelompok objek pada Toolbar Control Report (**Report Control Toolbar**), seperti gambar dibawah ini :



Icon	Name	Function
	<b>Select Object</b>	Digunakan untuk memilih objek/mengaktifkan suatu objek didalam laporan (memindahkan atau menghapus objek).
	<b>Label</b>	Untuk membuat seperti Judul laporan, informasi/keterangan pada laporan.
	<b>Field</b>	Untuk menampilkan data dari tabel dan sering diletakkan pada bagian Detail pada laporan.
	<b>Line</b>	Untuk membuat garis pada desain laporan.
	<b>Picture</b>	Digunakan untuk menampilkan gambar (bisa berupa logo perusahaan) pada desain laporan.

#### 10.4. Pembuatan Report Secara Wizard

Sama halnya dalam pembuatan Form, dalam pembuatan Report/laporan juga kita dapat menggunakan fasilitas Wizard. Dengan menggunakan wizard, kita dapat membuat laporan dengan mudah dan cepat dengan cara mengikuti langkah-langkahnya.

Langkah Pembuatan Laporan secara Wizard :

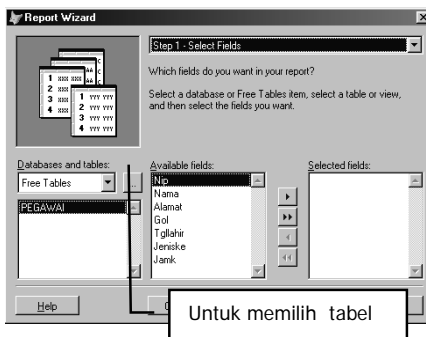
##### 1. Pilih menu **File – New – Report – Wizard**



Pada kotak dialog *Wizard Selection*, Pilih :

- **One-to-Many Report Wizard** untuk membuat laporan yang datanya lebih dari satu tabel.
- **Report Wizard** untuk membuat laporan dari satu tabel.

##### 2. Step 1, memilih Field yang akan ditampilkan pada laporan. Urutannya :



- Pilih tabel yang akan dibuat laporan.
- Pilih beberapa/seluruh field yang akan ditampilkan pada laporan.



Untuk memilih Beberapa Field satu Persatu



Untuk Memilih Seluruh Field Tabel

- Klik tombol **Next>**

3. Step 2, menentukan apakah kita akan melakukan Grouping Record.
4. Step 3, Tentukan Style laporan yang kita ingin inginkan.
5. Step 4, Tentukan bentuk laporan (Column atau Row), jumlah kolom dan posisi kertas.
6. Step 5, Tentukan apakah kita akan melakukan pengurutan data.
7. Step 6, Tentukan Judul untuk laporan tersebut.
8. Kemudian klik FINISH, simpan laporan tersebut.

Menampilkan Field Data pada laporan secara cepat :

1. Pilih **Quick Report** dari menu **Report**
2. Pilih nama tabel sumber data
3. Pilih bentuk laporan (Column atau Row)

### 10.5. Memanggil Laporan

Setelah kita rancang/desain laporan sesuai dengan keinginan kita, kita tinggal menampilkan laporan tersebut dan melakukan pencetakan ke media kertas atau printer. Untuk menampilkan hasil desain laporan dari *Report Designer*, kita tinggal pilih *Print Preview* pada menu *File* untuk melihat dilayar monitor atau pilih *Print* untuk mencetak ke printer.

Tetapi untuk memanggil laporan tersebut dari sebuah form yang telah kita buat sebelumnya, kita memerlukan perintah untuk menampilkan laporan tersebut.

Perintah Untuk memanggil/menampilkan Laporan :

- REPORT FORM <nama-file-lap.> PREVIEW** → Untuk mencetak ke Layar  
**REPORT FORM <nama-file-lap.> TO PRINTER** → Untuk mencetak ke printer

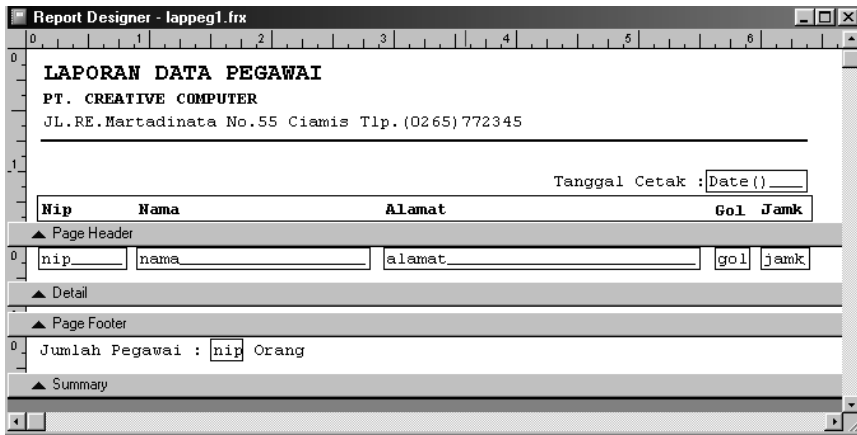
Perintah tersebut kita ketikkan pada Command Button yang sebelumnya kita buat pada Procedure **Click**.

## TUGAS PRAKTIKUM XXI

1. Buatlah laporan data pegawai dari tabel **Pegawai.DBF** dengan menggunakan cara Wizard. Kemudian simpan file laporan tersebut dengan nama **LAPPEG.FRX**.
2. Field tabel yang ditampilkan pada laporan yaitu : NIP,NAMA,TGLLAHIR,ALAMAT,GOL.
3. Lakukan Peng-Group-an berdasarkan GOL.
4. Pilihlah Style laporan sesuai dengan keinginan.
5. Lakukan Pengurutan berdasarkan field NIP secara Ascending.
6. Berilah Judul laporan LAPORAN DATA PEGAWAI
7. Lihatlah hasil laporan tersebut.

## TUGAS PRAKTIKUM XXII

1. Buatlah laporan dengan Report Designer dari tabel Pegawai.DBF dengan perintah *CREATE REPORT*. Rancanglah desain laporan dengan menggunakan Report Controls Toolbar, sehingga tampilannya seperti dibawah ini.



2. Kemudian setelah laporan tersebut didesain, simpanlah laporan tersebut dengan nama **LAPPEG1.FRX**.
3. Lihatlah hasil laporan tersebut dengan cara *PRINT PREVIEW*.
4. Bukalah kembali File Form **INPUTPEG.SCX** dan tambahkan satu tombol lagi pada form tersebut.

Aturlah Properties Command Button Tersebut :

 Caption : \<**LAPORAN**

Kemudian pada Command Button tambahkan Procedure **Click**. Untuk memanggil laporan LAPPEG1.FRX dari form tersebut.

#### REPORT FORM LAPPEG1.FRX PREVIEW

5. Kemudian jalankan Form tersebut, cobalah untuk mengklik tombol LAPORAN untuk menampilkan laporan yang telah kita rancang/desain.
6. Simpan kembali form tersebut.

## TUGAS PRAKTIKUM XXIII

1. Buatlah Form dibawah ini yang digunakan untuk memanggil laporan LAPPEG1.FRX untuk dicetak dilayar dan dicetak ke printer.



Aturlah properties masing-masing object form tersebut.

2. Simpan form tersebut dengan nama **CETAKLAP.SCX**

## TUGAS PRAKTIKUM XXIV

1. Buatlah kembali form dibawah ini untuk mencetak data pegawai per golongan dari laporan LAPPEG1.FRX kelayar atau ke printer.



The image shows a screenshot of a Visual FoxPro form window. The window title is "LAPORAN". Inside the window, the title "LAPORAN PEGAWAI PERGOLONGAN" is displayed. There are two main input areas: "Golongan" with a dropdown menu labeled "Combo1", and "Cetak Ke:" with two radio buttons, "Layar" (which is selected) and "Printer". At the bottom of the form, there are two buttons: "Cetak" and "Keluar".

Aturlah properties masing-masing object form tersebut.

2. Simpanlah form tersebut dengan nama **CETAKGOL.SCX**

# Modul 11

## Membuat Menu & Kompilasi

### 11.1. Pendahuluan

Agar program yang dibuat lebih mudah dipakai, pengembang perlu membuat menu. Menu merupakan salah satu objek Visual Foxpro. Dengan menu, pemakai akhir (end user) hanya memilih menu untuk menjalankan aplikasi tertentu. File menu di Visual Foxpro mempunyai ekstensi file .MNX dan file memonya .MPR.

### 11.2. Merancang Menu

Perencanaan menu yang tepat sangat penting untuk meyakinkan bahwa semua fungsi dalam aplikasi yang kita buat akan dimanfaatkan oleh pemakai secara maksimal dan tidak membingungkan. Pada saat merancang menu pertimbangkan pedoman berikut ini :

- Organisasikan sistem menurut tugas yang akan dilakukan user, bukan menurut hirarki program dalam aplikasi.
- Berikan judul yang memiliki arti yang jelas dan mudah bagi setiap menu.
- Organisasikan item-item menu menurut frekuensi penggunaannya.
- Buat garis pemisah diantara kelompok ke kelompok item menu.
- Batasi jumlah item dimenu pada saat satu layar komputer
- Juka panjangnya melebihi panjang layar komputer.
- Jika jumlahnya melebihi panjang layar, buatlah submeu untuk menu item yang sesuai
- Pilih tombol akses dan shortcut keyboard bagi menu dan item menu
- Gunakan kata yang jelas makanya bagi2 item-item menu.
- Gunakan kombinasi huruf besar dan huruf kecil pada item menu.

Untuk menyajikan semua fasilitas seperti diatas, misalnya dibuatlah rancangan menu sebagai berikut :

Menu Utama	Submenu
File	Input Data Pegawai
	Edit & Hapus Data Pegawai
	Pencarian Data Pegawai
	Keluar
Laporan	Laporan Pegawai keseluruhan
	Laporan Pegawai Per Golongan
Info	Tentang Program

### 11.3. Membuat Menu

Untuk membuat menu di Visual Foxpro ada beberapa cara yang bisa kita lakukan, yaitu diantaranya :

#### 11.3.1 Membuat Menu Lewat Menu Utama

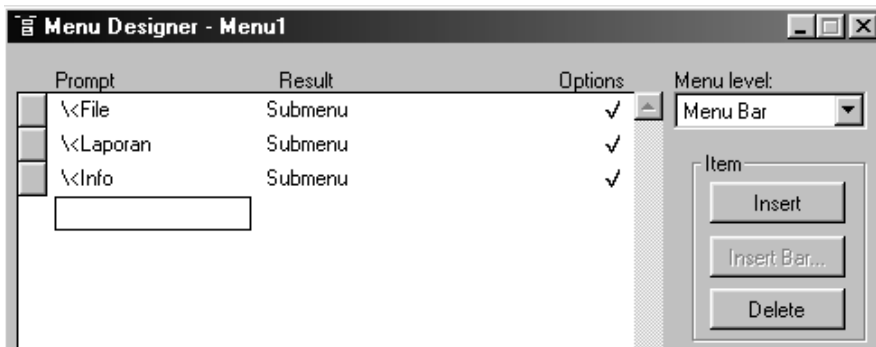
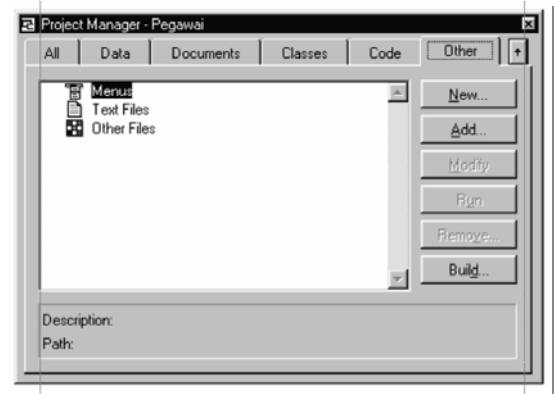
Untuk membuat menu lewat menu utama, caranya adalah :

- Dari menu utama, pilih **File > New**
- Pilih File Type nya > **Menu**
- Klik **New File**

#### 11.3.2 Membuat Menu Lewat Quick Menu

Untuk membuat menu lewat Quick Menu, langkahnya adalah :

- Di Project Manager, pilih tab **Other > Menus**
- Klik tombol **New**
- Pada kotak dialog **New Menu**, klik **Menu** sehingga tampil **Menu Designer**



#### 11.3.3 Men-generate Menu

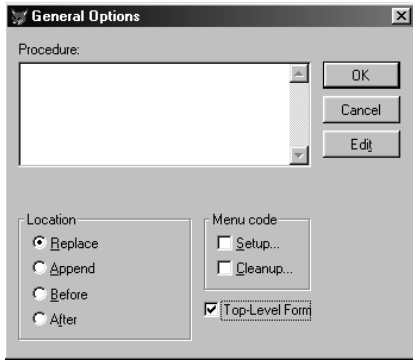
Untuk menjalankan menu harus di generate dahulu, sehingga menjadi file berekstensi **.MPR**. cara men-generate menu sebagai berikut :

1. Di **Project Manager**, pilih tab **Other > Menus**
2. Pilih **Menu**
3. Pilih **Menu > Generate ... > Generate**
4. Tentukan nama file menu tersebut.





11.3.4 Membuat Menu SDI ( Single-document Interface )



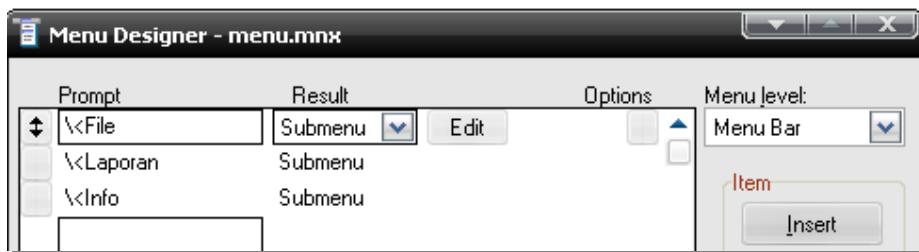
Menu SDI adalah menu yang tampil pada window in Single-Document Interface (SDI). Untuk membuat menu SDI, kita harus menandai bahwa menu tersebut akan digunakan pada form SDI saat kita merancang menu tersebut. Untuk membuat menu SDI, pada saat menu designer tampil :

- Pilih menu **View – General Options**
- Aktifkan tanda cek **Top-Level Form**

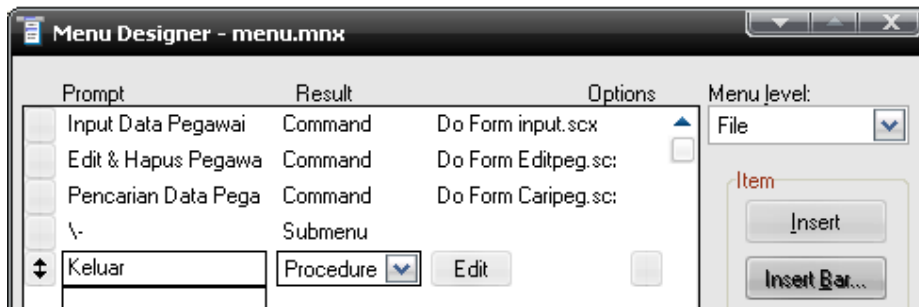
**TUGAS PRAKTIKUM XXV**

1. Buatlah sebuah menu dengan rancangan seperti dibawah ini :

Menu Utama	Submenu
File	Input Data Pegawai
	Edit & Hapus Data Pegawai
	Pencarian Data Pegawai
	Keluar
Laporan	Laporan Pegawai keseluruhan
	Laporan Pegawai Per Golongan
Info	Tentang Program



Untuk membuat submenu, pilih **Result > Submenu**, kemudian klik tombol **Edit**.



Untuk mengisi perintah pada tombol keluar, pilih **Result > Procedure**, kemudian klik **Edit**. Kembali lagi ke pilihan Menu Bar, pilih pada **Menu Level**.

Kemudian ketikkan listing programnya sebagai berikut :

```

Menu Designer - menu - Keluar Procedure *
jawab=MESSAGEBOX("Yakin akan keluar ?",4+32,'Perhatian')
IF jawab=6
    QUIT
ELSE
    application.ActiveForm.release()
    CLEAR ALL
ENDIF

```

2. Buatlah sebuah form untuk digunakan menampilkan menu. Rancanglah form tersebut seperti dibawah ini dan kemudian simpan dengan nama **MenuUtama.scx**:

Properties Form :

Object	Properties	Nilai/Value
Form1	Caption	Program Aplikasi Pegawai
	ShowWindow	2 – As Top-Level Form
	WindowState	2 – Maximized

Ketikkan Procedure pada object Form

**Form1** Procedure **Active**

*Do menu.mpr with thisform, .t.*

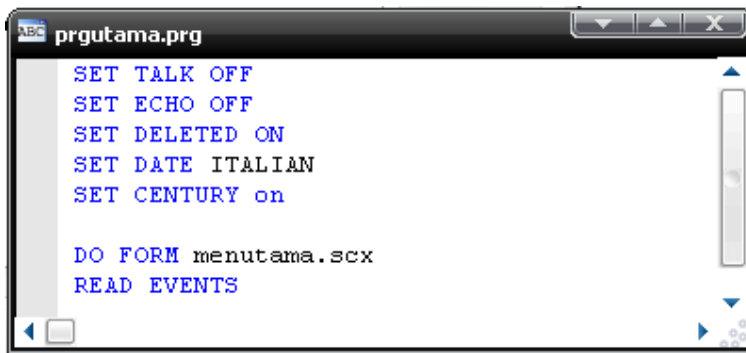
3. Jalankan form tersebut dan lihatlah hasilnya.

### 11.3.5 Kompilasi Project

Pada waktu kita membuat program, mungkin ada kesalahan yang terjadi. Untuk memudahkan kita mengetahui kesalahan dan lokasi kesalahan, kita perlu men-debug kesalahan program. Jika ingin mendistribusikan program ke komputer lain, kita perlu mengkompilasi. Program yang telah dikompilasi dapat dipakai pada komputer yang tidak ada sistem Visual Foxpro.

Membuat File EXE dari Project pegawai :

1. Buatlah program utama dan beri nama file dengan **Prgutama.prg**. Pada tab **Code**, pilih **Programs**, kemudian klik **New**. Kemudian ketikkan listing perintah didalamnya seperti dibawah ini :



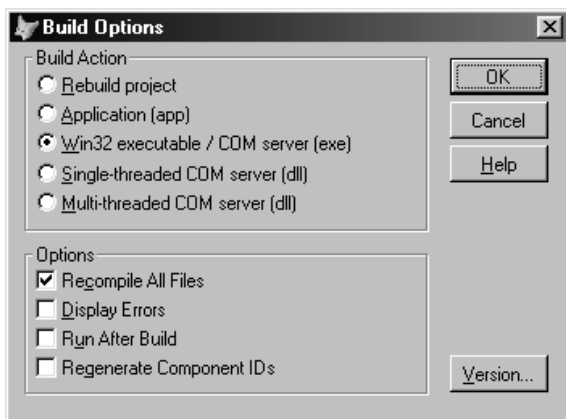
```

SET TALK OFF
SET ECHO OFF
SET DELETED ON
SET DATE ITALIAN
SET CENTURY on

DO FORM menutama.scx
READ EVENTS

```

2. Jadikan Prgutama.prg menjadi **Set Main**, dengan mengklik kanan pada Prgutama dan pilih **Set Main**.
3. Di **Project Manager**, pilih **Build...** , muncul kotak **Build Option**.



4. Pilih **Win32 Executable/COM server (exe)** di Build Action
5. Pilih **Option > Recompile All Files**, untuk mengkompilasi seluruh perintah yang pernah dikompilasi.