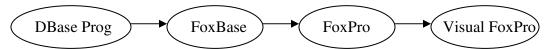
Bahasa Pemrograman Visual Foxpro 6 DIPLOMA III-AMIK BSI

Pertemuan 1

MENGENAL DAN BEKERJA DENGAN VISUAL FOXPRO

1. Sekilas Tentang VisualFoxPro



Visual FoxPro 6.0 datang besama produk-produk Microsoft lain dalam paket Visual Studio 6.0, seperti Visual Basic, Visual C++, Visual Interdev, Visual J++, Visual SourceSafe, dan MSDN Library.

FoxPro adalah sebuah Sistem Manajemen Database (*Database Management System / DBMS*) elektronik, yang berfungsi untuk mengumpulkan, mengambil dan menampilkan data. Sedangkan Visual FoxPro sendiri adalah bahasa pemrograman visual yang berorientasi pada obyek (*Object Oriented Programming / OOP*) dan juga sebagai Sistem Manajemen Database Relasional (*RDMS / Sistem Basis Data*) yang memungkinkan kita:

- Merancang sendiri format penyimpanan informasi.
- Menyimpan data dalam jumlah besar.
- Menata data dalam bentuk tabel.
- Mendapatkan informasi berdasarkan kriteria tertentu (query).
- Merancang sendiri form untuk memasukkan data ke dalam tabel.
- Mengolah data menjadi informasi dan menyajikannya dalam bentuk laporan.
- Merangkai berbagai unsur penggunaan di dalamnya menjadi suatu aplikasi.

No.	Spesifikasi	Maksimum
1.	Jumlah record per table	1 milyar
2.	Ukuran file table	2 giga byte
3.	Jumlah karakter per record	65500
4.	Jumlah field per record	255
5.	Jumlah tabel aktif dalam saat yang sama	2551
6.	Jumlah karakter per field	254

7.	Ukuran field numeric	20 digit
8.	Presisi perhitungan numeric	16 digit

Tabel 1. Spesifikasi Teknis Visual FoxPro

Beberapa istilah dalam Visual FoxPro:

✓ Project Manager : untuk mengatur pengembangan, integrasi dan

pembuatan aplikasi. Yang dinamakan *project* adalah

kumpulan file, data, dokumen dan obyek yang

digunakan dalam Visual FoxPro 6.0 dan mempunyai

ekstensi .PJX. *Project Manager* memudahkan

pemakai untuk mengatur dan mengorganisasi

database, tabel, form, laporan dan query selama

pembuatan aplikasi.

✓ Database : kumpulan dari tabel, view, dan connection. File

database berekstensi .DBC

✓ Tabel : tempat penyimpanan data (berekstensi .DBF).

✓ View : Tabel yang dibentuk dengan mengambil sebagian /

seluruh field dari satu / beberapa tabel / view lain.

✓ Connection : penghubung antara tabel dengan data dari sumber lain

yang formatnya berbeda (seperti : MS. Acces, Oracle,

SQL. Excel, Paradox, dll).

2. Cara Mengaktifkan Visual FoxPro

Ada berbagai cara untuk mengaktifkan Visual FoxPro:

a. Dari Taskbar : Start, Program, Microsoft Visual Studio 6.0,

Microsoft Visual FoxPro 6.0

b. Windows Explorer.

c. Shortcut.

Kemudian akan tampil layar menu utama Visual FoxPro 6.0 yang terdiri dari :

- ⇒ Baris Judul (*Title Bar*)
- ⇒ Baris Menu (Menu Bar)
- ⇒ Toolbar
- ⇒ Jendela Command (Command Window)
- ⇒ Baris Status

3. Mengenal Sistem Menu

Sistem menu pada MS. Visual FoxPro terdiri dari:

a. Menu Bar : terletak di bawah baris judul.

b. Menu Pad : menu yang terletak pada baris menu.

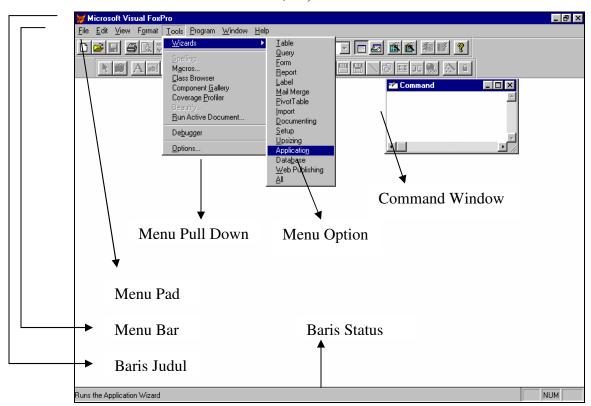
c. Menu Pull Down : submenu yang muncul ke bawah pada saat menu-

menu pad dipilih.

d. Menu Option : pilihan-pilihan yang tersedia pada suatu submenu.

e. Menu Pop UP : menu yang dapat muncul di berbagai tempat (semua

toolbar, dll)



4. Mengenal Jendela Command

Selain melalui menu yang sudah tersedia pada baris menu, pemakai juga dapat memasukkan instruksi secara langsung melalui jendela command. Cara mengaktifkannya .

- ✓ Klik menu Window
- ✓ Klik Command Window atau dengan menekan Ctrl-F2

5. Mengoperasikan Visual FoxPro

- a. Buat project baru : File, New, Project, Wizard, ketikkan nama file, ok.
- b. Buat tabel

Tabel pada Visual FoxPro dibedakan menjadi dua macam:

- 1. Tabel yang menjadi bagian suatu database.
- 2. Tabel yang TIDAK menjadi bagian dari suatu database (tabel bebas).

Secara prinsip, cara membuat kedua macam tabel tersebut sama.

Caranya:

Dari command window :

CREATE nama tabel

Contoh: CREA Latih1

Dari Menu :

File, New, Tabel, New File, tentukan direktori penyimpanan data, ketikkan nama file (Latih 1), Ok.

Contoh: Nama file tabel = Latih1 dengan struktur file sbb:

Nama Field	Tipe	Ukuran	Desimal
Nim	Character	9	
Nama_Mhs	Character	5	
Tgl_Lahir	Date	8	
Kelas			
Alamat	Character	25	
Phone	Numeric	15	0

- Nama Field:
 - Maksimal 10 karakter.
 - Diawali dengan huruf atau garis bawah (_).
 - Selanjutnya dapat berupa huruf, angka atau garis bawah.

Contoh: Nip, Nama_Mhs

- c. Isi file dengan 10 record.
- d. Aktifkan tabel
 - Dari command window :

USE nama tabel

Contoh: USE Latih1

Dari Menu:

File, Open, pilih nama file tabel (Latih1), Ok.

- e. Tampilkan tabel
 - Dari command window :

BROWSE

Dari Menu:

View, Browse.

- f. Modifikasi file
 - ✓ Ubah struktur file
 - Dari command window :

MODIFY STRUCTURE atau MODI STRU

Dari Menu:

View, Tabel Designer

- ✓ Tambah record
 - Dari command window :

APPEND atau APPE

Dari Menu:

Tabel, Append New Record

✓ Hapus record

Dari command window :

Delete [scope] nomor record.

Ket: Scope: All : Semua record

Next : Dari record yang aktif sampai record yang

ke N.

Record : Record ke N.

Rest : Dari record yang aktif sampai record yang

terakhir.

Contoh: Delete record 2

Dari Menu:

Tabel, Delete Record, pilih Scope dan tentukan nomor record, delete.

- Menyembunyikan record yang sudah dihapus:
 - Dari command window :

SET DELETE ON

- ✓ Menampilkan kembali record yang sudah dihapus :
 - Dari command window :

SET DELETE OFF

Brow

- ✓ Mengaktifkan kembali record yang sudah dihapus :
 - Dari command window :

Recall [scope] nomor record.

Contoh: Recall record 2

Dari Menu:

Tabel, Recall Record, pilih Scope dan tentukan nomor record, recall.

- ✓ Menghilangkan record.
 - Dari command window :

PACK

Dari Menu:

Tabel, Remove Deleted Record.

- g. Menutup tabel:
 - Dari command windowsCLOSE TABEL
- h. Keluar dari Visual FoxPro:
 - Dari command window: CLOS ALL

QUIT

Dari Menu: File, Exit

Bahasa Pemrograman Visual Foxpro 6 DIPLOMA III-AMIK BSI

Pertemuan 2

KONSEP DASAR DATABASE VISUAL FOXPRO

2.1. Pengertian Database

Database adalah suatu pengorganisasian data dengan tujuan agar data adapt diakses dengan mudah. Sebagai contoh, database dapat diterapkan untuk dikelola. Pada umumnya, sebuah database terdiri atas sejumlah table.

Sebagai gambaran, database tentang pelanggan adapt terdiri dari dua buah tabel. Tabel pertama tentang daftar pelanggan dan tabel kedua berupa data pembayaran perbulan. Tabel daftar pelanggan adapt dilukiskan sebagai kumpulan kartu dengan masing-masing kartu mewakili data seorang pelanggan.

Tingkatan dari database adalah:

- Database adalah gabungan dari record dan field yang sejenis
- Record adalah gabungan dari beberapa field.
- Field adalah elemen yang terkecil dari suatu data.
- Items adalah elemen huruf atau angka yang terdiri dari satu buah.

Sebuah database dinyatakan dengan extention **.DBC** (**Database Container**). File dengan ext ini berisi dari beberapa table yang berkaitan dengan database, dipakai untuk membuat suatu project file (**.pjx**)

2.2. Membuat tabel

Tabel dalam Visual FoxPro ada 2 macam yaitu:

1. Tabel didalam database extentionnya .DBC

2. Tabel diluar database (bebas) extentionnya .DBF

Prinsip dari pembuatan dari tabel ini hampir sama. Cara membuat tabel adalah :

- Create *nama_file* atau
- Menu bar
 - Klik File New Table New File Simpan *nama_filenya* Save.

♣ Mendefinisikan struktur tabel

Terdapat kolom yang berupa:

- Field_name : Menyatakan nama field

- Type : Jenis data

- Width : Lebar data field

- Decimal : Jumlah digit untuk pecahan

- Null : Membedakan nilai nol dengan nilai yang tidak diisi.

♣ Syarat nama file dan nama field yaitu :

- Max 10 char untuk tabel bebas dan 254 untuk tabel didalam database.

- Diawali dengan huruf atau garis bawah

- Tidak booleh ada spasi

- Selanjutnya dapat berupa angka

A Perintah-perintah yang dipakai dalam database adalah:

- Menyimpan database : Ctrl+W atau Ok

- Menutup database
- Close all, Close database, Use
- Membuka database
: Use nama_file, Klik File-Open

- Menampilkan database : List dan Browse

- Menambah database : Append

- Melihat file pada dos : Dir *. Dbf

❖ Tipe Data Visual FoxPro

No.	Tipe Field	Keterangan	Ukuran	
1.	Character	Alfanumerik, dapatberupa alfabet,	1 – 254	
		tanda baca, angka.		
2.	Currency	Mata uang, hingga 20 digit	8 (otomatis)	
		termasuk tanda +/- dan titik		
		desimal, dengan 4 digit desimal.		
3.	Numeric	Data bilangan, hingga 20 digit	1 – 20	
		termasuk tanda +/- dan titik		
		desimal.		
4.	Float	Data bilangan, hingga 20 digit	1 – 20	
		termasuk tanda +/- dan titik		
		desimal.		
5.	Date	Tanggal	8 (otomatis)	
6.	DateTime	Tanggal dan Jam	8 (otomatis)	
7.	Double	Data numeric presisi ganda,	8 (otomatis)	
		biasanya untuk data scientific.		
8.	Integer	Data bilangan bulat hingga 10	4 (otomatis)	
		digit.		
9.	Logical	Data logic, yaitu .T. / .F.	1 (otomatis)	
10.	Memo	Data catatan dalam jumlah yang 4 (otomatis		
		besar.		
11.	General	Objek OLE (Object Linking and	4 (otomatis)	
		Embedding)		
12.	Character	Data alfanumerik, disimpan dalam 1 – 254		
	(binary)	bentuk binary		
13.	Memo	Data catatan, disimpan dalam	4 (otomatis)	
	(binary)	bentuk binary.		

Bahasa Pemrograman Visual Foxpro 6 DIPLOMA III-AMIK BSI

Pertemuan 3

OPERASI TERHADAP TABEL

1. Membuka kembali suatu tabel

Mengaktifkan kembali tabel yang telah ditutup ada beberapa cara yaitu :

- Cara 1 :
 - 1. Pilih menu bar File
 - 2. Pilih Open
 - 3. Atur file dengan ext .DBF
 - 4. Klik nama tabelnya
- Cara 2 :

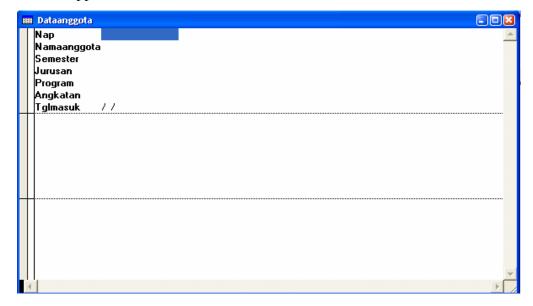
Use *nama_file* contohnya: Use Latih1

2. Menambah data

Data yang telah disimpan dalam tabel dapat ditampilkan dengan menggunakan perintah Append. Contohnya:

Use Latih1

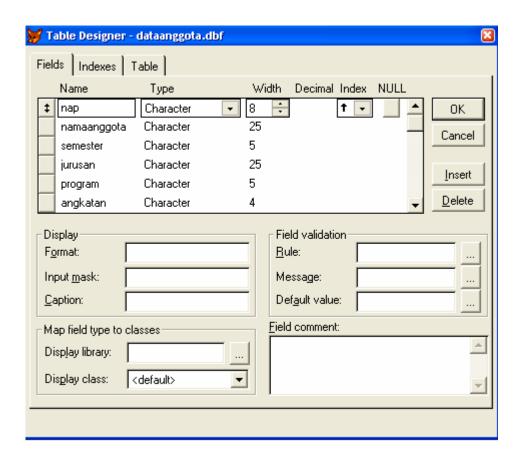
Append



3. Memodifikasi struktur tabel

Struktur dari tabel yang telah dibuat biasa ditambahkan, dimodifikasi, atau diperbaiki jika ada kesalahan dengan menggunakan perintah sepeti berikut ;

Modify Structure atau disingkat Modi Stru



4. Mengubah data dengan REPLACE

Pada model interaktif mengubah data dengan Browse tetapi pada model pemrograman mengubah/mengganti data dengan perintah Replace.

B.U. REPLACE nama_field with nama_field_baru

Contoh: use latih1

Go top 1

?nama,kota

Replace kota with SOLO

5. Perintah IIF

Fungsi iif() juga dapat bertindak seperti perintah if pada Excel, formatnya : iif(kondisi,ekspresi1,ekspresi2)

Bahasa Pemrograman Visual Foxpro 6

DIPLOMA III-AMIK BSI

Pertemuan 4

OPERASI TERHADAP TABEL (Lanjutan)

- 1. Aktifkan project anda.
- 2. Buatlah sebuah table dengan nama PEGAWAI.DBF dan struktur filenya sbb:

Nama Field	Tipe	Ukuran	Desimal
NIP	Character	5	
NAMA	Character	30	
J_KELAMIN	Character	1	
TEMPAT_LHR	Character	20	
TGL_LHR	Date	8	
ALAMAT	Character	30	
TELEPON	Numeric	15	
КОТА	Character	20	
TGL_MASUK	Date	8	
DEPT	Character	3	
GAJI	Numeric	8	0
UANG_MAKAN	Numeric	5	0
TRANSPORT	Numeric	5	0
CATATAN	Memo	4	
FOTO	General	4	

- 3. Isi file dengan 10 record.
- 4. Mengubah / mengganti / mengisi data secara otomatis.

Perintah BROWSE biasa digunakan untuk mengubah data pada mode interaktif. Pada mode pemrograman, perintah yang biasa digunakan untuk mengisi data secara otomatis adalah REPLACE.

Bentuk Umum : REPLACE [scope] field WITH statement

Contoh: GOTO 2

REPLACE KOTA WITH "JOGJA"

REPLACE ALL UANG_MAKAN WITH 10000

5. Fungsi IIF()

Digunakan untuk menyeleksi data dengan kondisi tertentu.

Bentuk Umum : IIF(kondisi,jawaban1,jawaban2)

Contoh: REPLACE ALL J_KELAMIN WITH IIF(LEFT(NIP,1)="1","L","P")

(Mengisi secara otomatis field J_KELAMIN dengan "L" jika angka pertama NIP = "1", selain itu "P").

6. Menyeleksi data dengan FILTER.

Filter merupakan salah satu utilitas pada Visual FoxPro untuk menyaring record data pada suatu table.

Caranya:

- ✓ Aktifkan file table.
- ✓ Tampilkan table : View, Browse
- ✓ Table
- ✓ Properti
- ✓ Klik Data Filter

✓ Isi kriteria pada ekspresi, misalnya :

J_KELAMIN == "P"

✓ Ok

✓ Ok

Catatan: Untuk menghapus kondisi filter, hapuslah ekspresinya dengan

cara seperti di atas.

Beberapa Fungsi Untuk Operasi Tabel II

1. Menghapus record

Record yang tidak diperlukan dapat dengan mudah dihapus. Perintah yang

digunakan adalah **DELETE**.

Untuk menghapus suatu record, digunakan perintah sbb:

B.U. Delete Record *nomor_record*

Co: Delete Record 4

Untuk menghapus seluruh record adalah: Delete All

2. Membatalkan penghapusan record

Apabila record yang telah dihapus (oleh delete), masih ada kesempatan untuk

membatalkan penghapusan terhadapnya.

B.U. Recall All

Tetapi record yang telah dihapus dengan perintah Pack tidak dapat di-recall all kembali.

3. Menghapus record secara permanen

Penghapusan dengan delete sebenarnya tidaklah menghapus secara permanen

didalam file tabel. Visual FoxPro menandai record dengan suatu tanda khusus, perintah

yang digunakan untuk menghapus secara permanen adalah PACK. Setelah perintah ini

diberikan tidak ada lagi record yang ditandai.

4. Mengosongkan tabel

Semua record yang ada pada suattu tabel dapat dikosongkan secara permanen dengan perintah **ZAP**.

5. Set Deleted

Perlu diketahui bahwa penghapusan denga delete sebenarnya tidak meghapus secara permanen. Berikut adalah perintah untuk pemakaian set deleted yaitu :

- 1. Buka tabel DBF misalnya: Use Latih1
- 2. Ketik Recall all untuk membatalkan semua penghapusan
- 3. Ketik Browse
- 4. Keluar dari Browse dan berikan perintah :

Go top

Delete

5. Ketik di jendela command dengan perintah

Set deleted on

6. Ketik Browse dan lihat hasilnya.

Bahasa Pemrograman Visual Foxpro 6

DIPLOMA III-AMIK BSI

Pertemuan 5

POINTER

Bentuk pointer di Visual FoxPro berupa tanda anak panah kecil di depan sebuah record yang berfungsi untuk menunjukkan record yang sedang aktif sekarang. Pada saat suatu table dibuka, pointer berada pada record pertama.

1. RECNO()

Perintah ini digunakan untuk mengetahui nomor record yang sedang aktif sekarang.

Caranya:

- ✓ Aktifkan table PEGAWAI.DBF
- ✓ Dari command window ketikkan:

? RECNO()

✓ Di layar monitor akan tampil nomor record yang sedang aktif sekarang.

2. GOTO

Pointer record dapat diatur untuk pindah ke record tertentu dengan menggunakan perintah GO atau GOTO.

Bentuk Umum:

GO nomor record

Dari command window :

GO₂

DISPLAY

Dari Menu:

Table, Go to Record, Record #, masukkan nomor record (2), Ok.

- 3. Pindah ke record pertama (BOF / Begining Of File):
 - Dari command window :

GO 1

DISP

Atau GO TOP

DISP

Dari Menu:

Table, Go to Record, Top

- 4. Pindah ke record terakhir (EOF / End Of File):
 - Dari command window :

GO BOTTOM

DISP

Dari Menu:

Table, Go to Record, Bottom

5. SKIP

Digunakan untuk pindah dari record yang sedang aktif ke record berikutnya.

Dari command window :

GO TOP

SKIP

? RECNO()

SKIP 2

? RECNO()

SKIP-1

? RECNO()

Dari Menu:

Table, Go to Record, Next / Previous

6. LOCATE

Digunakan untuk meletakkan pointer ke record yang dicari.

Dari command window :

LOCATE FOR KOTA="JOGJA"

? RECNO()

Dari Menu:

Table, Go to Record, Locate, pilih scopenya (misal : ALL), masukkan kriterianya di For (misal : Pegawai.kota=="JOGJA"), locate.

Jika record yang di cari tidak ada maka di baris status akan tampil "End of Locate scope".

Bahasa Pemrograman Visual Foxpro 6 DIPLOMA III-AMIK BSI

Pertemuan 6

FILE INDEX

Menata urutan record lebih dikenal dengan indeks. File hasil index berekstensi .CDX.

Fungsi utama indeks adalah untuk menata urutan data

sehingga lebih mempercepat pencarian data.

- ❖ Istilah dalam index :
 - * Index Key : satu atau beberapa field yang dijadikan dasar penentuan urutan (sering disebut kunci index / key field).

* Ascending : mengindex dari kecil ke besar (A - Z / 1 - 10)

Descending : mengindex dari besar ke kecil (Z - A / 10 - 1)

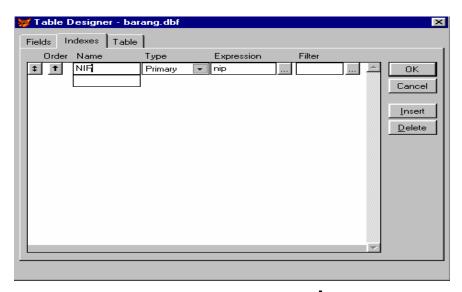
- **❖** Membuat index :
 - Dari command window :

Bentuk Umum: INDEX ON field TO namafile

Contoh : USE PEGAWAI

INDEX ON nip TO coba

- Dari Menu:
 - ✓ Aktifkan file tabel
 - ✓ View, Browse
 - ✓ Vie`w, Table designer
 - ✓ Klik tab indexes



Sorder: menunjukkan urutannya (♠Ascending /♦Descending).

➤ Name: menyatakan index tag (nama pengenal suatu index).

Primary: menyatakan kunci utama pada table yang terdapat dalam database. Index ini berguna untuk menangani field yang nilainya tidak boleh sama. Sebuah table hanya boleh memiliki sebuah index primary.

Candidate : sama halnya dengan index primary, nilai setiap field tidak boleh sama. Bedanya, jumlah index candidate boleh lebih dari satu dalam satu table.

Unique : index yang mengurutkan sejumlah record berdasarkan kemungkinan pertama dari nilai field yang dijadikan index. Nilai yang sama dari sejumlah record hanya akan diwakili oleh sebuah record.

Regular : index yang memungkinkan field yang dijadikan sebagai index memiliki nilai sama.

Expression: pernyataan dasar urutan (index key), dapat berupa satu field atau gabungan beberapa field.

- ✓ Pilih Ascending pada Order.
- ✓ Ketikkan Nip pada Name.
- ✓ Pilih Primary pada Type.
- ✓ Ketikkan Nip pada Expression.
- ✓ Ok, Yes
- Mengaktifkan file hasil index :
 - Dari command window :

Bentuk Umum: SET INDEX TO namafile

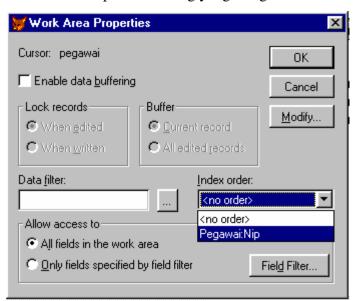
Contoh : USE PEGAWAI

SET INDEX TO coba

atau

USE PEGAWAI INDEX coba

- Dari Menu:
 - ✓ Aktifkan file tabel
 - ✓ View, Browse
 - ✓ Table, Property
 - ✓ Klik Index Order dan pilih index tag yang diinginkan.



- ✓ Ok
- ✓ Perhatikan apa yang terjadi pada tampilan Browse.

❖ Pencarian record dengan SEEK :

Pencarian suatu record tertentu dapat dilakukan dengan cepat seandainya file index digunakan. Namun tentu saja data untuk pencarian record harus sama dengan isi field kunci.

Dari command window :

```
Contoh: SEEK '11111' { mencari NIP = '11111' }
? FOUND() { jika hasilnya .T. = ditemukan }
? Nama { menampilkan namanya }
```

Microsoft Visual FoxPro 6.0 DIPLOMA III-AMIK BSI

Pertemuan 7

Pencarian Data

I. Pencarian data dengan LOCATE

Fungsi Locate adalah untuk meletakan pointer record ke record yang dicari. Bentuk perintah ini adalah :

Locate For nama_field

Contoh: Locate for kota="Klaten" (lihat pada baris status)

?Display (melihat hasilnya)

Continue (mencari record berikutnya)

End of Locate scope (jika record tidak ada lagi yang kota="Klaten")

Untuk mengetahui apakah yang dicari denga Locate ketemu atau tidak, dapat digunakan perintah **FOUND**(). Perintah memberi nilai berupa :

- .T. : data yang dicari ketemu

- .F. : data yang dicari tidak ketemu

Contoh: Locate for kota="Klaten"

?Found()

II. Pencarian record dengan SEEK

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

Contoh: Use Pegawai

Set order to tag nip SEEK "23970964"

?Found()

?Nip,Nama,Alamat

I.6. Pencarian record secara Eksak

Bila bermaksud mencari suatu data bertipe karakter pada field yang telah diindeks, perlu diketahui efek keadaan SET EXACT ON atau OFF.

Contoh: Use Pegawai Use Pegawai

Set order to tag nip

SET EXACT OFF

SEK "23970964"

SEK "23970964"

SEK "23970964"

?Found() ?Found()

?Nip,Nama,Alamat ?Nip,Nama,Alamat

Dari contoh diatas dapat diartikan jika set exact dalam kondisi **off** anda dapat mencari suatu nilai atau record, kalau dalam kondisi **on** tidak dapat.

SET EXACT juga berpengaruh terhadap pernyataan LOCATE (untuk mencari suatu data yang tidak diindeks) ataupun pembandingan string karakter dengan operator =

Bahasa Pemrograman Visual Foxpro 6 DIPLOMA III-AMIK BSI

Pertemuan 8

OPERASI DATA DENGAN TABEL MAJEMUK

1. Pengertian Tabel Majemuk

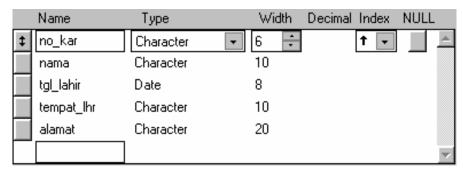
Tabel majemuk adalah jika pemakai dalam pembuatan sebuah aplikasi menggunakan sejumlah tabel (lebih dari satu tabel) atau beberapa database. Visual FoxPro menyediakan mekanisme yang disebut sesi data dan area kerja untuk menangani penggunaan sejumlah tabel, sebesar 32767 area kerja.

Area kerja adalah area bernomor yang kegunaannya untuk mengidentifikasi sebuah tabel yang sedang terbuka (setiap tabel yang terbuka memiliki area kerja sendiri).

2. Mengaktifkan Tabel pada Area Kerja

Bila sejumlah tabel akan digunakan, setiap tabel harus ditempatkan pada area kerja yang berbeda.

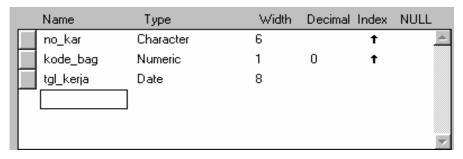
Buatlah tabel PIBADI.DBF dengan struktur file sbb :



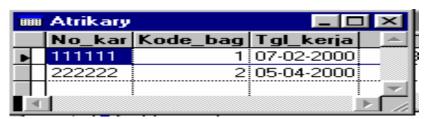
- ⇒ Index file berdasarkan no_kar dengan type primary.
- ⇒ Isi sebanyak 2 record

ĦĦ	Pribadi					X
	No_kar	Nama	Tgl_lahir	Tempat_lhr	Alamat	۸
	111111	Hasan	02-02-2002	Bandung	JI. Batujajar 7	
Þ	222222	Dewi	04-05-2000	Jakarta	Jl. Kramat 5	
						∇
Ŀ					Þ	Γ_{h}

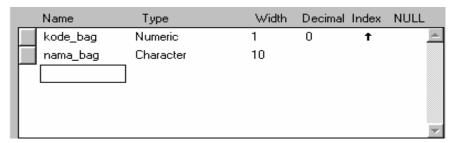
Buatlah tabel ATRIKARY.DBF dengan struktur file sbb:



- ⇒ Index file berdasarkan no_kar dengan type primary dan kode_bag dengan type regular.
- ⇒ Isi sebanyak 2 record



Buatlah tabel BAGIAN.DBF dengan struktur file sbb:



- ⇒ Index file berdasarkan ko_bag dengan type primary.
- ⇒ Isi sebanyak 2 record



Area kerja yang digunakan tabel dimulai dari nomor 1 atau huruf A. Penentuan area kerja dilakukan dengan menggunakan perintah SELECT.

Bentuk Umum : SELECT area_kerja

Contoh:

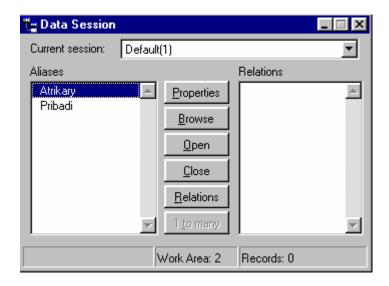
SELECT 1

USE pribadi (tabel PRIBADI.DBF diletakkan pada area kerja 1)

SELE 2

USE atrikary (tabel ATRIKARY.DBF diletakkan pada area kerja 2)

SET (menampilkan tabel-tabel yang sedang terbuka)

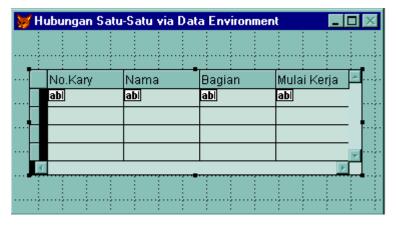


MERANCANG PROGRAM APLIKASI MENGGUNAKAN TABEL MAJEMUK DENGAN DATA ENVIRONMENT

1. Hubungan Satu-Satu pada Data Environment.

Seorang programmer dapat menggunakan lingkungan data (*data environment*) bila ingin membuat sebuah form yang menggunakan sejumlah tabel. Dengan cara seperti ini tabel-tabel yang dilibatkan akan dibuka secara otomatis dan juga akan ditutup secara otomatis ketika form berakhir. Programmer tidak perlu menuliskan kode untuk membuat hubungan antartabel sebagaimana yang biasa dilakukan pada kejadian **Load** milik form. Contoh program:

✓ Buatlah form dengan lay-out sbb :



✓ Hasil Program adalah sbb :



Caranya:

- 1. Aktifkan project anda.
- 2. Buat / aktifkan database personal.
- 3. Pastikan bahwa tabel pribadi, tabel atrikary, dan tabel bagian telah aktif.
- 4. Buatlah sebuah form baru.
- 5. Setelah muncul form kosong, lakukan pengaturan terhadap properti-properti milik form dan kemudian buatlah sejumlah kontrol serta aturlah properti-propertinya sesuai dengan table berikut :

Tabel Objek dan property yang perlu disertakan & diatur.

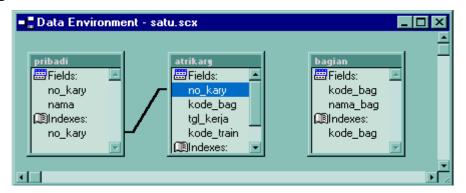
Nama Objek	Properti	Nilai
Form	Caption	Hubungan Satu-Satu via Data
		Environment.
Grid	ColumnCount	4
	Height	169
	Left	15
	Name	GridInformasi
	ReadOnly	.T.
	RecordSource	atrikary
	RecordSource Type	1
	Тор	38
	Width	343
Column	Name	Column 1
(bagian dari Grid)	ControlSource	Atrikary.no_kary
Header	Name	Header 1
(bagian Column 1)	Caption	No. Kary.
Column	Name	Column 2

(bagian dari Grid)	ControlSource	Pribadi.nama
Header	Name	Header 1
(bagian Column 2)	Caption	Nama
Column	Name	Column 3
(bagian dari Grid)	ControlSource	Bagian.nama_bag
Header	Name	Header 1
(bagian Column 3)	Caption	Bagian
Column	Name	Column 4
(bagian dari Grid)	ControlSource	Atrikary.tgl_kerja
Header	Name	Header 1
(bagian Column 4)	Caption	Mulai Kerja

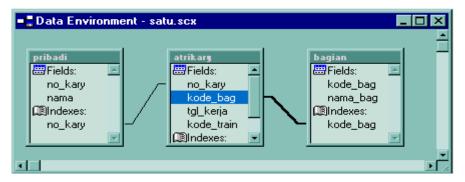
- 6. Klik kanan mouse pada bagian yang tidak ada obyeknya.
- 7. Pilih Data Environment
- 8. Klik double pribadi
- 9. Klik double atrikary
- 10. Klik double bagian
- 11. Klik Close



12. Klik pada field no_kary milik ATRIKARY dan gerakkan mouse ke no_kary pada bagian indexes milik tabel PRIBADI.



13. Klik pada field kode_bag milik ATRIKARY dan gerakkan mouse ke kode_bag pada bagian indexes milik tabel PRIBADI.



- 14. Tutup jendela Data Environment
- 15. Isi untuk form procedure Load:

SET DELETED ON

{ *Ket* : agar record yang terhapus tidak diproses }

16. Isi untuk form procedure Activate:

SELECT atrikary

{ Ket : menjadikan ATRIKARY sebagai tabel sekarang ketika form diaktifkan }

17. Jalankan program dengan mengklik icon Run [

18. Simpan file dengan nama Satu.SCX di direktori form.

2. Hubungan Satu-Banyak pada Data Environment.

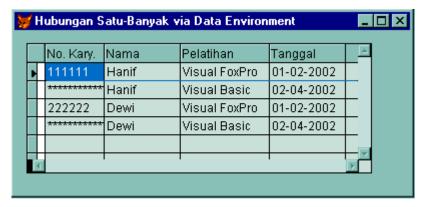
Contoh kasus : seseorang dapat mengikuti banyak pelatihan.

Contoh program:

✓ Buatlah form dengan lay-out sbb :



✓ Hasil Program adalah sbb :



Caranya:

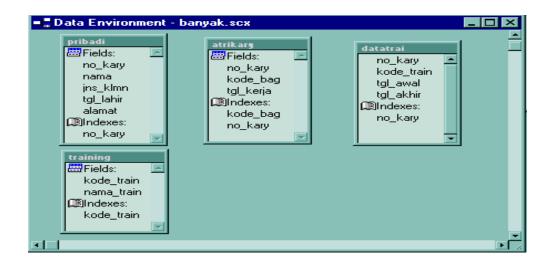
- 1. Aktifkan project anda.
- 2. Buat / aktifkan database personal.
- 3. Pastikan bahwa tabel pribadi, tabel atrikary, tabel datatrai dan tabel training telah aktif.
- 4. Buatlah sebuah form baru.
- 5. Setelah muncul form kosong, lakukan pengaturan terhadap properti-properti milik form dan kemudian buatlah sejumlah kontrol serta aturlah properti-propertinya sesuai dengan table berikut :

Tabel Objek dan property yang perlu disertakan & diatur.

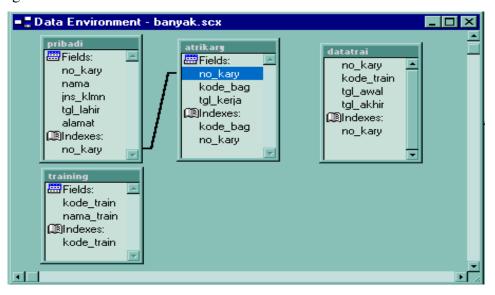
Nama Objek	Properti	Nilai
Form	Caption	Hubungan Satu-Banyak via
		Data Environment.
Grid	ColumnCount	4
	Name	GridInformasi
	ReadOnly	.T.

	RecordSource	atrikary
	RecordSource Type	1
Column	Name	Column 1
(bagian dari Grid)	ControlSource	Atrikary.no_kary
Header	Name	Header 1
(bagian Column 1)	Caption	No. Kary.
Column	Name	Column 2
(bagian dari Grid)	ControlSource	Pribadi.nama
Header	Name	Header 1
(bagian Column 2)	Caption	Nama
Column	Name	Column 3
(bagian dari Grid)	ControlSource	Training.nama_train
Header	Name	Header 1
(bagian Column 3)	Caption	Pelatihan
Column	Name	Column 4
(bagian dari Grid)	ControlSource	datatrai.tgl_awal
Header	Name	Header 1
(bagian Column 4)	Caption	Tanggal

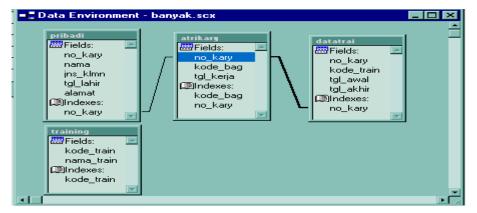
- 6. Klik kanan mouse pada bagian yang tidak ada obyeknya.
- 7. Pilih Data Environment
- 8. Klik double pribadi
- 9. Klik double atrikary
- 10. Klik double datatrain
- 11. Klik double training
- 12. Klik Close



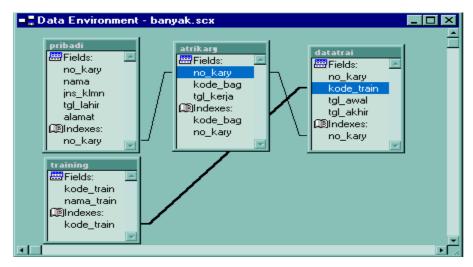
13. Klik pada field no_kary milik ATRIKARY dan gerakkan mouse ke no_kary pada bagian indexes milik tabel PRIBADI.



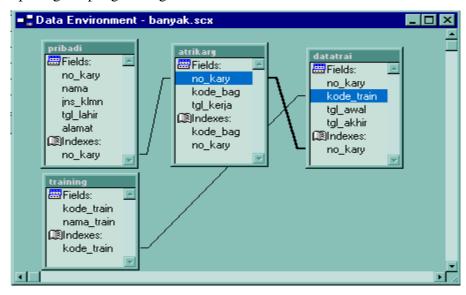
14. Klik pada field no_kary milik ATRIKARY dan gerakkan mouse ke no_kary pada bagian indexes milik tabel DATATRAI



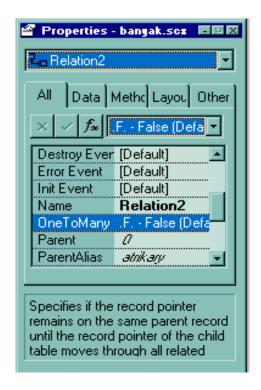
15. Klik pada field kode_train milik DATATRAI dan gerakkan mouse ke kode_train pada bagian indexes milik tabel TRAINING



16. Klik pada garis penghubung antara tabel ATRIKARY dan DATATRAI



17. Langkah ini akan mengakibatkan jendela properties menampilkan properti milik garis penghubung tersebut.



- 18. Gantilah nilai .F. pada properti **OneToMany** menjadi .T. (nilai .T. inilah yang menyatakan bahwa hubungan berupa satu-banyak).
- 19. Tutup jendela Data Environment
- 20. Isi untuk form procedure Load:

SET DELETED ON

21. Isi untuk form procedure Activate:

SELECT atrikary

- 22. Jalankan program dengan mengklik icon Run [
- 23. Simpan file dengan nama Banyak.SCX di direktori form.

Bahasa Pemrograman Visual FoxPro DIPLOMA III-AMIK BSI

Pertemuan 9

KONSEP DASAR RELASI DATA

3. One To One

Hubungan adalah suatu istilah yang menyatakan kaitan antara suatu tabel dengan tabel lain. Contoh: hubungan dapat dibentuk antara tabel PRIBADI.DBF dengan ATRIKARY.DBF. Dengan hubungan ini, pemakai dapat memperoleh data pada ATRIKARY.DBF berdasarkan sebuah nomor karyawan (NO_KAR) pada tabel PRIBADI.DBF. Hubungan dua tabel tersebut yang bersifat satu-satu disebut *One To One* (satu field pada tabel pertama *hanya* berhubungan dengan satu field pada tabel kedua **atau** ketiga).



Untuk pembentukan hubungan digunakan perintah SET RELATION.

Bentuk Umum: SET RELATION TO field INTO alias [ADDITIVE]

Ket: field = field yang menjadi relasi.

alias = tabel yang akan dihubungkan dengan tabel pada area kerja

sekarang.

additive = hubungan yang sudah ada pada tabel sekarang dibiarkan tetap ada.

Contoh1: SELE 1

USE pribadi

SET ORDER TO TAG no_kar

SELE 2

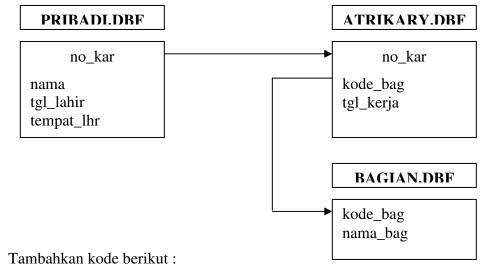
USE atrikary

SET RELATION TO no_kar INTO pribadi BROWSE field no_kar, pribadi.nama,tgl_kerja

m Atrikary		_ [] ×	
	No_kar	Nama	Tgl_kerja	-
Þ	1111111	ali	07-08-2002	
	222222	mahmud	05-04-2000	
4				<u> </u>

Contoh2:

Pemakai dapat mengembangkan lebih jauh dengan menghubungkan ATRIKARY.DBF dengan BAGIAN.DBF sehingga hubungan ketiga file sbb :



SELE 3

USE bagian

SET ORDE TO TAG kode_bag

SELE 2

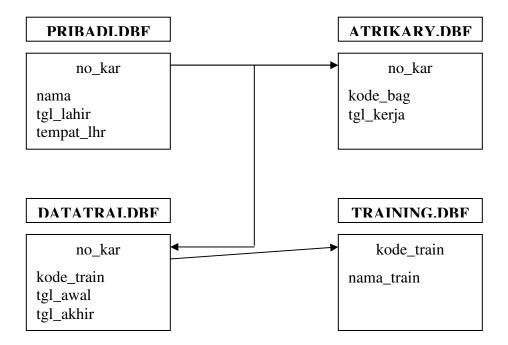
SET RELATION TO kode_bag INTO bagian ADDITIVE

BROW FIELD no_kar, pribadi.nama, bagian.nama_bag, tgl_kerja



4. One To Many

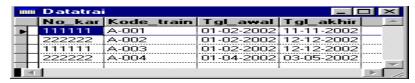
One To Many adalah satu field pada tabel pertama dapat berhubungan dengan satu field pada tabel kedua dan ketiga.



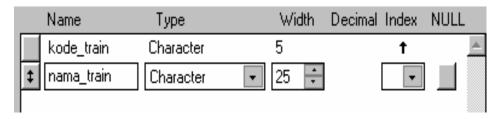
Buatlah tabel DATATRAI.DBF dengan struktur file sbb:



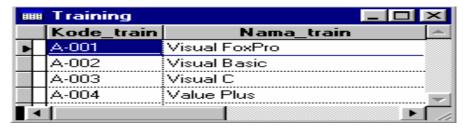
- ✓ Index file berdasarkan no_kar dengan type regular (field no_kar dapat sama karena seorang pegawai dapat mengikuti banyak pelatihan).
- ✓ Isi sebanyak 4 record



Buatlah tabel TRAINING.DBF dengan struktur file sbb:



- ✓ Index file berdasarkan kode_train dengan type primary
- ✓ Isi sebanyak 4 record



Dari Command Window :

MODI COMM ONETOMANY

Ketikkan program berikut ini :

Sele 1

USE c:\zidan\data\atrikary.dbf

set order to no_kar

Sele 2

USE c:\zidan\data\pribadi.dbf

set order to no_kar

Sele 3

USE c:\zidan\data\datatrai.dbf

set order to no_kar

Sele 4

USE c:\zidan\data\training.dbf

Set order to kode_train

Sele datatrai

Set relation to kode_train into training

*** Menjadikan ATRIKARY sebagai tabel orang tua / parent table***

Sele atrikary

Set relation to no_kar into pribadi

Set relation to no_kar into datatrai additive

*** Membentuk Hubungan Satu Banyak ***

Set skip to datatrai

Brow field no_kar,pribadi.nama,datatrai.kode_train,training.nama_train,tgl_kerja

Hasil program sebagai berikut :

HH	Atrikary				_
	No_kar	Nama	Kode_train	Nama_train	Tgl_kerja 📥
Þ	111111	Hasan	A-001	Visual FoxPro	07-02-2000
	*****	Hasan	A-003	ValuePlus	*****
	222222	Dewi	A-002	Visual Basic	05-04-2000
	*****	Dewi	A-004	Visual C	********
1					<u> </u>

(Ket : Hubungan One To Many disini terjadi pada saat seorang karyawan dapat mengikuti banyak pelatihan).

5. Many To Many

Many To Many adalah satu field pada tabel pertama hanya berhubungan dengan satu field pada tabel kedua, ketiga, dst dan hubungan yang terjadi bersifat banyak [Contoh: seorang karyawan (nama_kar) dapat mengikuti banyak pelatihan (nama_train) dapat diikuti oleh banyak karyawan[nama_kar].

DIPLOMA III-AMIK BSI

Pertemuan 10

FUNGSI - FUNGSI STANDAR

1. Operasi Tanggal dan Waktu.

Visual FoxPro menyediakan sejunlah fungsi yang berkaitan dengan operasi tanggal dan waktu. Tabel berikut ini mencantumkan namanama fungsi yang berkaitan dengan hal ini beserta kegunaannya.

Tabel Fungsi Tanggal dan Waktu

Nama Fungsi	Keterangan dan Contoh
CDOW()	Bentuk : CDOW(Tanggal)
	Fungsi ini menghasilkan nama hari dalam bahasa Inggris.
	Contoh: ? CDOW(ctod("8/20/2002") \Rightarrow Tuesday
CMONTH()	Bentuk : CMONTH(Tanggal)
	Fungsi ini menghasilkan nama bulan dalam bahasa Inggris.
	Contoh: ? CMONTH(ctod("8/20/2002") \Rightarrow August
CTOT()	Bentuk : CTOT(String)
	Fungsi ini menghasilkan nilai DateTime berdasarkan ekspresi
	bertipe string.
	Contoh: ? CTOT("8/20/2002 10:00") \Rightarrow 8/20/2002 10:00
DATE()	Bentuk : DATE()
	Fungsi ini menghasilkan nilai tanggal sekarang.
	Contoh: $? DATE() \Rightarrow 05/04/2002$
DATETIME()	Bentuk : DATETIME()
	Fungsi ini menghasilkan nilai tanggal dan jam sekarang
	sebagai nilai DateTime.
	Contoh: ? DATETIME() \Rightarrow 05/04/2002 09:36
DAY()	Bentuk : DAY(Tanggal)
	Fungsi ini menghasilkan tanggal (1 s.d 31).
	Contoh: ? DAY(ctod("8/20/2002") \Rightarrow 20
DMY()	Bentuk : DMY(Tanggal)

	Fungsi ini menghasilkan tanggal, bulan dan tahun
	Contoh: ? DMY(ctod("8/20/2002") \Rightarrow 20 August 2002
DOW()	Bentuk : DOW(Tanggal)
	Fungsi ini menghasilkan tanggal kode hari (1= Minggu,dst).
	Contoh: $? DOW(ctod("8/20/2002") \Rightarrow 3$
MDY()	Bentuk : MDY(Tanggal)
	Fungsi ini menghasilkan tanggal dengan format : bulan
	tanggal, tahun.
	Contoh: $? MDY(ctod("8/20/2002") \Rightarrow August 20, 2002$
MONTH()	Bentuk : MONTH(Tanggal)
	Fungsi ini menghasilkan bulan dari suatu tanggal (1 – 12)
	Contoh: $?$ MONTH(ctod("8/20/2002") \Rightarrow 8
TIME()	Bentuk : TIME()
	Fungsi ini menghasilkan jam sekarang, dengan format :
	JJ:MM:DD
	Contoh: $? TIME() \Rightarrow 10:08:53$
YEAR()	Bentuk : YEAR(Tanggal)
	Fungsi ini menghasilkan tahun dari suatu tanggal.
	Contoh: ? YEAR(ctod("8/20/2002") \Rightarrow 2002

2. Operasi Manipulasi String.

Visual FoxPro menyediakan sejumlah fungsi yang berkaitan dengan manipulasi string. Tabel berikut ini mencantumkan nama-nama fungsi yang berkaitan dengan hal ini beserta kegunaannya

Tabel Fungsi Tanggal dan Waktu

Nama Fungsi	Keterangan dan Contoh
AT()	Bentuk : AT(StDicari,StSumber,Awal) Fungsi ini menghasilkan letak string StDicari pada string StSumber. Bila Awal disediakan, pencarian akan dimulai dari posisi ini. Bila Awal tidak disediakan, pencarian dimulai dari string terkiri pada StSumber. Hasilnya 0 kalau StDicari tak terdapat pada StSumber. Contoh : ? AT("A","ABCDE") ⇒ 1 ? AT("ALI","BALI") ⇒ 2 ? AT("a","ABCDE") ⇒ 0
ALLTRIM()	Bentuk : ALLTRIM(String) Fungsi ini menghasilkan data sesuai dengan ekspresi,

	dengan seluruh spasi di bagian kiri dan kanan akan
	dibuang.
	Contoh: ? ALLTRIM(" ABC ") \Rightarrow ABC
ISALPHA()	Bentuk : ISALPHA(String)
	Fungsi ini menghasilkan .T. kalau string pertama dari
	argumen berupa huruf.
	Contoh : $?$ ISALPHA("A") \Rightarrow .T.
ISLOWER()	Bentuk : ISLOWER(String)
	Fungsi ini menghasilkan .T. kalau string pertama dari
	argumen berupa huruf kecil.
	Contoh : $?$ ISLOWER("a") \Rightarrow .T.
ISUPPER()	Bentuk : ISUPPER(String)
	Fungsi ini menghasilkan .T. kalau string petama dari
	argumen berupa huruf besar.
	Contoh : ? $ISUPPER("A") \Rightarrow .T$.
LEFT()	Bentuk : LEFT(String,n)
	Fungsi ini menghasilkan n buah huruf terkiri dari string.
	Contoh : ? LEFT("ABCDE",2) \Rightarrow AB
LEN()	Bentuk : LEN(String)
	Fungsi ini menghitung jumlah string.
	Contoh: ? LEN("ABCDE") \Rightarrow 5
LOWER()	Bentuk : LOWER(String,n)
	Fungsi ini merubah huruf besar menjadi huruf kecil.
	Contoh : ? LOWER("Visual") \Rightarrow visual
LTRIM()	Bentuk : LTRIM(String)
	Fungsi ini membuang spasi di depan string.
	Contoh : ? LTRIM(" Visual") \Rightarrow "Visual"
PROPER()	Bentuk : PROPER(String,n)
	Fungsi ini merubah semua awal kata menjadi huruf besar.
	Contoh : ? PROPER("visual foxpro") ⇒ Visual Foxpro
RAT()	Bentuk : RAT(StDicari,StTarget)
	Fungsi ini menghasilkan posisi StDicari pada StTarget.
	Contoh: $? RAT("C","ABCDEA") \Rightarrow 3$
REPLICATE()	Bentuk : REPLICATE(String,n)
"	Fungsi ini menghasilkan pengulangan string sebanyak n
	kali.
	Contoh: ? REPLICATE("=",10) \Rightarrow ========
RIGHT()	Bentuk : RIGHT(String,n)
	Fungsi ini menghasilkan n buah huruf terkanan dari string.
	Contoh : $? RIGHT("ABCDE",3) \Rightarrow CDE$
	? RIGHT("ABCDE",1) \Rightarrow E
RTRIM()	Bentuk : RTRIM(String)
1.11(11)()	Fungsi ini membuang spasi di belakang string.
<u> </u>	i ungoi ini membuang spasi ui belakang sunig.

	Contoh : ? RTRIM("Visual ") ⇒ "Visual"
SPACE()	Bentuk : SPACE(n)
	Fungsi ini menghasilkan pengulangan spasi sebanyak n
	buah.
	Contoh: $? SPACE(10) \Rightarrow "$
STUFF()	Bentuk : STUFF(Target, Awal, n, Pengganti)
	Fungsi ini menghasilkan string yang sesuai dengan string
	target, dengan n buah string dimulai dari posisi awal akan
	diganti dengan string pengganti.
	Contoh: $? STUFF("ABCDE",2,2,"XY") \Rightarrow AXYDE$
SUBSTR()	Bentuk : SUBSTR(String,Awal,n)
	Fungsi ini mengambil huruf di tengah string mulai dari
	Awal sebanyak n buah.
	Contoh: $?$ SUBSTR("ABCDEFG",3,2) \Rightarrow CD
TRANSFORM()	Bentuk : TRANSFORM(Target, Format)
	Fungsi ini menghasilkan string yang merupakan bentuk
	penyajian data Target dengan format tertentu.
	Contoh: ? TRANSFORM(12345678,"999,999,999.99") \Rightarrow
	12,345,678.00
TRIM()	Kegunaan sama dengan RTRIM().
UPPER()	Bentuk : UPPER(String)
	Fungsi ini merubah huruf kecil menjadi huruf besar.
	Contoh : ? UPPER("Visual") ⇒ VISUAL

DIPLOMA III-AMIK BSI

Pertemuan 11

FUNGSI – FUNGSI STANDAR LANJUTAN

3. Fungsi untuk Operasi Numerik.

Visual FoxPro menyediakan sejunlah fungsi yang berkaitan dengan operasi numeric (termasuk operasi trigonometri). Tabel berikut ini mencantumkan nama-nama fungsi yang berkaitan dengan hal ini beserta kegunaannya.

Tabel Fungsi untuk Operasi Numerik

Nama Fungsi	Keterangan dan Contoh
ABS()	Bentuk : ABS(Nilai)
	Fungsi ini menghasilkan nilai mutlak (absolut).
	Contoh: ? ABS(5) \Rightarrow 5
	? ABS(-5) \Rightarrow 5
ACOS()	Bentuk : ACOS(Nilai)
	Fungsi ini menghasilkan inversi cosinus dalam satuan radian.
	Nilai berkisar antara –1 s.d. 1.
	Contoh: ? $ACOS(-1) \Rightarrow 3.14$
	? $ACOS(0.8) \Rightarrow 0.64$
ASIN()	Bentuk : ASIN(Nilai)
	Fungsi ini menghasilkan inversi sinus dalam satuan radian.
	Nilai berkisar antara –1 s.d. 1.
	Contoh: $? ASIN(-1) \Rightarrow -1.57$
	? $ASIN(0.8) \Rightarrow 0.93$
ATAN()	Bentuk : ATAN(Nilai)
	Fungsi ini menghasilkan inversi tangen dalam satuan radian.
	Nilai berkisar antara –1 s.d. 1.
	Contoh: ? ATAN(-1) \Rightarrow -0.79
	? ATAN(0.8) \Rightarrow 0.67
CEILING()	Bentuk : CEILING(Nilai)
	Fungsi ini menghasilkan nilai bulat yang lebih dari atau sama
	dengan nilai argumen.

	Contoh: ? CEILING(1.1) \Rightarrow 2
	? CEILING(1.9) \Rightarrow 2
COS()	Bentuk : COS(Radian)
()	Fungsi ini menghasilkan nilai cosinus dari suatu sudut (sudut
	dinyatakan dalam radian).
	Contoh: $? COS(-1) \Rightarrow 0.54$
	$? \cos(0.8) \Rightarrow 0.70$
FLOOR()	Bentuk : FLOOR(Nilai)
V	Fungsi ini menghasilkan nilai bulat yang kurang dari atau
	sama dengan nilai argumen.
	Contoh: $? FLOOR(1.1) \Rightarrow 1$
	? $FLOOR(1.9) \Rightarrow 1$
INT()	Bentuk : INT(Nilai)
•	Fungsi ini menghasilkan bagian bulat dari suatu bilangan real.
	Contoh: $? INT(1.1) \Rightarrow 1$
	$? INT(1.9) \Rightarrow 1$
LOG()	Bentuk : LOG(Nilai)
	Fungsi ini menghasilkan nilai logaritma alami.
	Contoh: $? LOG(10) \Rightarrow 2.30$
	$? LOG(20) \Rightarrow 3.00$
LOG10()	Bentuk : LOG10(Nilai)
	Fungsi ini menghasilkan nilai logaritma berbasis 10.
	Contoh: $? LOG10(10) \Rightarrow 1$
	$? LOG10(100) \Rightarrow 2$
MOD()	Bentuk : MOD(Yang dibagi, Pembagi)
	Fungsi ini menghasilkan sisa pembagian.
	Contoh: $? MOD(6,2) \Rightarrow 0$
	$? MOD(7,2) \Rightarrow 1$
PI()	Bentuk : PI()
	Fungsi ini menghasilkan nilai π (3.14)
	Contoh: $? PI() \Rightarrow 3.14$
SIN()	Bentuk : SIN(Nilai)
	Fungsi ini menghasilkan nilai sinus dari suatu sudut/radian.
	Nilai berkisar antara –1 s.d. 1.
	Contoh: $? SIN(-1) \Rightarrow -0.84$
	$? SIN(0.8) \Rightarrow 0.72$
SQRT()	Bentuk : SQRT(Nilai)
	Fungsi ini menghasilkan akar kuadrat.
	Contoh: $? SQRT(25) \Rightarrow 5$
	$? SQRT(100) \Rightarrow 10$
TAN()	Bentuk : TAN(Nilai)
	Fungsi ini menghasilkan nilai tangen dari suatu sudut. Nilai
	berkisar antara –1 s.d. 1.
	Contoh: $? TAN(-1) \Rightarrow -1.56$

$9 \text{ TAN}(0.9) \rightarrow 1.25$	
$(\mathbf{A} \mathbf{N} \mathbf{U} \mathbf{A} \mathbf{I} \rightarrow \mathbf{I} \mathbf{A} \mathbf{I} $	
$: IM(0.0) \rightarrow 1.33$	

4. Fungsi Konversi Tipe dan Ekspresi

Visual FoxPro menyediakan sejumlah fungsi yang berkaitan dengan konversi tipe dan ekspresi. Tabel berikut ini mencantumkan namanama fungsi yang berkaitan dengan hal ini beserta kegunaannya

Tabel Fungsi Konversi Tipe dan Ekspresi

Nama Fungsi	Keterangan dan Contoh
ASC()	Bentuk : ASC(Karakter)
	Fungsi ini menghasilkan nilai ASCII dari karakter yang
	menjadi argumennya.
	Contoh: ? $ASC("A") \Rightarrow 65$
CHR()	Bentuk : CHR(Nilai)
	Fungsi ini menghasilkan sebuah karakter yang nilai
	ASCIInya dicantumkan dalam argumen.
	Contoh: $? CHR(65) \Rightarrow A$
CTOD()	Bentuk : CTOD(String Tanggal)
	Fungsi ini menghasilkan data bertipe tanggal berdasarkan
	string argumennya.
DTOC()	Bentuk : DTOC(Tanggal)
	Fungsi ini menghasilkan string yang menyatakan tanggal.
DTOS()	Bentuk : DTOS(Tanggal)
	Fungsi ini menghasilkan string tanggal dengan format
	YYYYMMDD (Y=tahun,M=bulan,D=tanggal).
MAX()	Bentuk : MAX(Nilai1,Nilai2)
	Fungsi ini menghasilkan nilai terbesar di antara kedua
	argumen. Tipe argumen dapat berupa karakter, tanggal,
	logika, numeric.
	Contoh: $? MAX(5,7) \Rightarrow 7$
	$? MAX(7,5) \Rightarrow 7$
MIN()	Bentuk : MIN(Nilai1,Nilai2)
	Fungsi ini menghasilkan nilai terkecil di antara kedua
	argumen. Tipe argumen dapat berupa karakter, tanggal,
	logika, numeric.
	Contoh: $? MIN(5,7) \Rightarrow 5$
	$? MIN(7,5) \Rightarrow 5$
STR()	Bentuk : STR(Nilai1)
	Fungsi ini menghasilkan nilai string dari suatu numeric.

	Contoh: $? STR(5) \Rightarrow "5"$
VAL()	Bentuk : VAL(String Nilai1).
	Fungsi ini menghasilkan nilai numerik dari suatu string.
	Contoh: $? STR(5) \Rightarrow "5"$

DIPLOMA III-AMIK BSI

Pertemuan 12

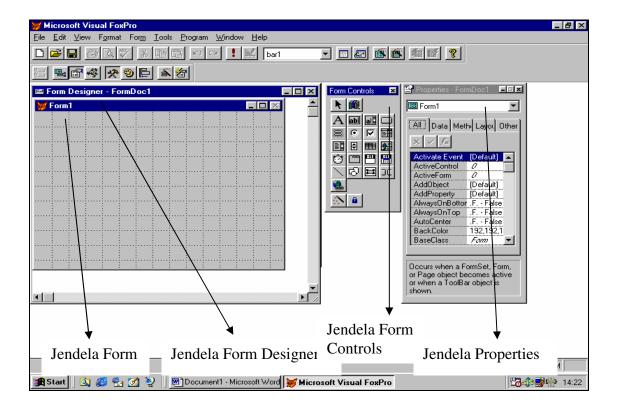
BEKERJA DENGAN PEMROGRAMAN VISUAL FOXPRO 6.0

A. Mengenal Object Form, Text Box dan Command Button.

Dalam pembuatan program Visual FoxPro akan melibatkan beberapa jendela, antara lain :

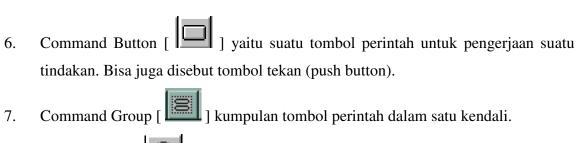
- Jendela Form Designer
- ☐ Jendela Form Controls, dan
- Jendela Properties

Form merupakan suatu bentuk tampilan berupa jendela yang akan muncul sebagai latar belakang dari tampilan suatu aplikasi. Sebuah aplikasi dapat memiliki banyak form (min. 1 form). Formlah yang menjadi antarmuka dengan pemakai aplikasi tersebut. Dalam perancangan, form berada pada **Form Designer**. Form Designer dapat mengandung lebih dari satu form.



Jendela *Form Controls* merupakan jendela yang memuat berbagai kontrol yang tersedia pada Visual FoxPro, antara lain :

- 1. Select Object [] untuk memilih object/kontrol yang digunakan.
- 2. View Classes [] untuk menampilkan kontrol yang lainnya. Jika pemakai mengklik kontrol ini maka akan tampil pilihan menu, yaitu :
 - Add : menambahkan kontrol baru.
 - Standard : menampilkan kontrol standar.
 - ActiveX Controls : menampilkan kontrol ActiveX.
- 3. Label [A] yang merupakan judul field adalah sama dengan nama field.
- 4. Text Box [] adalah objek yang berkaitan langsung dengan field berupa sebuah kotak. Di dalamnya pemakai dapat memasukkan suatu isian.
- 5. Edit Box [intuk menampilkan field memo.



- 8. Option Group [] / grup pilihan, digunakan bila ada sejumlah kemumgkinan pilihan, tetapi hanya satu pilihan yang boleh dipilih. Pemakai dapat mengklik pada lingkaran yang dikehendaki. Sering disebut dengan tombol radio.
- 10. Combo Box [] / kotak kombo, berupa suatu kotak yang memungkinkan pemakai mengisikan suatu data atau memilih data berdasarkan sejumlah pilihan yang tersedia.
- 11. List Box [] / kotak daftar, berupa suatu kotak yang berisi sejumlah pilihan Pemakai dapat menyorot pilihan yang dikehendaki, menggulung ke atas atau ke bawah.
- 12. Spinner [] kontrol / objek yang berfungsi untuk menaikkan / menurunkan nilai / value objek secara otomatis.
- 13. Grid [] kontrol yang berfungsi untuk menampilkan isi record suatu table.
- 14. Image [] untuk menampilkan gambar.
- 15. Timer [] pengatur pemrosesan suatu kegiatan berdasarkan waktu tertentu.
- 17. ActiveX Control [] kontrol ActiveX

- 18. ActiveX Bound Control [] kontrol ActiveX dengan batasan tertentu.
- 19. Line [] untuk membuat garis.
- 20. Shape [] untuk membuat objek kotak, lonjong, lingkaran dsb.
- 21. Container [] untuk menampung beberapa objek / kontrol didalamnya.
- 22. Separator [] batas pemisah antar kontrol.
- 23. Hyperlink []untuk membuat hyperlink.
- 24. Bulider Lock [] menampilkan / mengembalikan objek secara otomatis.

B. Pengertian Properties dan Event.

Jendela <u>Properties</u> merupakan tempat untuk mengatur hal-hal yang berkaitan dengan suatu kontrol yang terdapat pada form. Jendela ini mengandung lima buah tab:

- Data
- Methods
- Layout
- Other.

Karakteristik atau sifat (biasa disebut **property**) yang mendasari suatu kontrol dapat diatur melalui kelima tab di atas. Contoh property dari suatu kontrol adalah warna teks, ukuran teks, dsb. Semua property dapat diatur atau dilihat melalui tab All. Sedangkan tab-tab yang lain mengelompokkan property yang sejenis sesuai dengan judul tab.

Selain istilah properti, pemakai akan sering menjumpai istilah lain yaitu **kejadian** (*event*) dan **metode**. Kejadian menyatakan suatu tindakan seperti saat mouse

diklik, tombol ditekan, dan form ditutup. Metode adalah suatu fungsi yang terkait dengan objek.

Contoh 1 (Program tanpa menggunakan tabel) :

⇒ Buatlah sebuah form dengan lay-out sbb :



⇒ Hasil program adalah sbb :



Caranya:

- 1. Aktifkan project anda.
- 2. Buatlah sebuah form baru.
- 3. Setelah muncul form kosong, lakukan pengaturan terhadap propertiproperti milik form dan kemudian buatlah sejumlah kontrol serta aturlah properti-propertinya sesuai dengan table berikut :

Tabel	Objek	dan pi	operty	yang	perlu	disertakan	& diatur.
-------	-------	--------	--------	------	-------	------------	-----------

Nama Objek	Properti	Nilai	
Form	Caption	Program Tanggal dan Jam	
Label	Caption	Informasi	
Text Box	Name	txtinformasi	
Command Button	Caption	\ <jam sekarang<="" td=""></jam>	
	Name	cmdjam	
Command Button	Caption	Tanggal Sekarang	
	Name	cmdtanggal	
Command Button	Caption	Selesai	
	Name	cmdselesai	

4. Klik 2x pada command button Jam Sekarang dan ketikkan programnya :

ThisForm.txtInformasi.value="Jam Sekarang:" + time()

5. Klik 2x pada command button Tanggal Sekarang dan ketikkan programnya:

ThisForm.txtInformasi.value="Tanggal Sekarang:" + dtoc(date())

6. Klik 2x pada command button Selesai dan ketikkan programnya :

ThisForm.Release

- 8. Simpan file dengan nama Tgl.SCX di direktori form.

DIPLOMA III-AMIK BSI

Pertemuan 16

TAHAPAN MEMBUAT PROGRAM

- Menciptakan Database Lewat Project Manager

Project Manager merupakan fasilitas yang sangat berguna untuk mengorganisasikan komponen-komponen yang menyusun sebuah aplikasi. Dengan Project Manager juga memudahkan kompilasi komponen-komponen yang menyusun aplikasi. Cara terbaik membuat proyek adalah pada awal membuat suatu aplikasi.

Membuat Project

- 1. File >> New >> Project atau ketik pada jendela command perintah CREATE PROJECT
- 2. Beri nama project anda sesuai dengan nama project yang anda inginkan.

Project manager mengelompokkan komponen-komponen yang terlibat dalam sebuah proyek ke dalam beberapa halaman :

× All

Berisi semua komponen.

* Data

Berisi komponen yang berhubungan dengan data (database, tabel bebas, query)

* Docs

Berupa komponen dokumen (form, laporan, label)

X Classes

Berisi kelas, yang digunakan untuk membuat objek.

X Code

Berkaitan dengan kode (progran, pustaka API, aplikasi)

× Other

Berisi menu, fike teks, dan file gambar.

Membuat database

- 1. Klik tab Data atau klik tanda ∃ di depan item pada jendela Project Manager sehingga muncul komponen : Database, Free tables dan Queries.
- 2. klik pada item 🔁 Databases
- 3. klik tombol New.... >> Beri nama file database anda, lalu save pada directory yang sudah ditentukan.
- 4. Jendela Database designer muncul dengan kondisi kosong tanpa tabel. Database anda akan terlihat pada jendela project manager. Ekstension file database anda adalah .dbc.

Membuat tabel-tabel dalam database

- 1. Pada jendela Project Manager klik tanda

 yang ada di depan nama database anda.

 □ yang ada di depan nama database anda.
- 2. Klik item Tables lalu klik tombol New.... >> Beri nama file tabel anda, lalu save pada directory yang sama dengan directory anda menyimpan proyek.
- 3. Jendela Table Designer muncul setelah proses saving. Isikan field-field yang ada dalam rancangan tabel anda.
- 4. Jika selesai klik tombol Ok.
- 5. Ulangi langkah satu jika anda ingin membuat tabel yang lain.
- Cara lain untuk membuat tabel adalah : pada jendela Database designer klik kanan
 lalu pilih menu New table >> New table. Langkah selanjutnya sama seperti point no. 3

Note: Penamaan tabel yang dibuat dalam database harus konsisten, karena tabel tersebut akan dideklarasikan dalam kode program. Ketidakkonsistenan penamaan akan mengakibatkan kesalahan sewaktu penulisan kode program. Nama tabel sama seperti nama variable dalam pemrograman scientific.

Merancang Form

- 1. Pada jendela Project Manager klik tanda 🗉 yang ada di depan item 🕮 Documents.
- 2. Klik item Forms lalu klik tombol New.... >> New Form
- 3. Jendela Form Designer dengan Form1 di dalamnya akan muncul.
- 4. Atur disain anda sesuai dengan rancangan form di atas. Gunakan object-object yang ada pada jendela Form Control.
- 5. Simpan form dengan nama yang sesuai dengan rancangan form anda.
- 6. Jalankan form anda untuk melihat hasilnya dengan menekan tombol pada jendela Microsoft Visual Foxpro.
- 7. Tekan tombol untuk kembali ke jendela perancangan form.

Perancangan Program

Sebelum masuk ke dalam tahap penulisan kode-kode program, Anda harus membuat terlebih dulu alur logika program dengan flow chart. Dalam penulisan tugas akhir, flowchart sangat penting sebab dari sinilah Anda menjelaskan alur logika program yang anda buat kepada dosen penguji. Flowchart yang baik akan menghasilkan program yang efisien dan mudah untuk dipahami.

Menuliskan Kode Program Visual Foxpro

Pada Visual Foxpro, tiap object memiliki kejadian (event). Kode program langsung ditulis pada object yang bersangkutan sesuai dengan kejadiannya dengan cara :

- Klik kanan pada object yang ingin dituliskan kode programnya, pada menu yang muncul pilih Code. Jendela editor kode program akan muncul. Atau dengan cara
- \circ Klik menubar <u>View</u> > Code

Contoh Program tanpa menggunakan tabel):

⇒ Buatlah sebuah form dengan lay-out sbb :



⇒ Hasil program adalah sbb :



Caranya:

- 9. Aktifkan project anda.
- 10. Buatlah sebuah form baru.
- 11. Setelah muncul form kosong, lakukan pengaturan terhadap propertiproperti milik form dan kemudian buatlah sejumlah kontrol serta aturlah properti-propertinya sesuai dengan table berikut :

Tabel Objek dan property yang perlu disertakan & diatur.

Nama Objek	Properti	Nilai
Form	Caption	Program Tanggal dan Jam
Label	Caption	Informasi
Text Box	Name	txtinformasi
Command Button	Caption	\ <jam sekarang<="" td=""></jam>
	Name	cmdjam
Command Button	Caption	Tanggal Sekarang
	Name	cmdtanggal
Command Button	Caption	Selesai
	Name	cmdselesai

- 12. Klik 2x pada command button Jam Sekarang dan ketikkan programnya :

 ThisForm.txtInformasi.value="Jam Sekarang : " + time()
- 13. Klik 2x pada command button Tanggal Sekarang dan ketikkan programnya:

ThisForm.txtInformasi.value="Tanggal Sekarang:" + dtoc(date())

14. Klik 2x pada command button Selesai dan ketikkan programnya:

ThisForm.Release

- 15. Jalankan program dengan mengklik icon Run [].
- 16. Simpan file dengan nama Tgl.SCX di direktori form.

DIPLOMA III-AMIK BSI

Pertemuan 17

OPERATOR

1. Operator Relasi

Adalah operator yang digunakan untuk membandingkan dua buah ekspresi bertipe data sama dan menghasikan nilai .T.(benar) atau .F. (salah).

Tabel Operator Relasi

Operator	Keterangan
<	Kurang dari
>	Lebih dari
=	Sama dengan
==	Tepat sama dengan (bermanfaat untuk pembandingan string karakter)
<> atau # atau !=	Tidak sama dengan
<= atau =<	Kurang dari atau sama dengan
>= atau =>	Lebih dari atau sama dengan
\$	Pembadingan string

Tabel Contoh Operator Relasi

Ekspresi	Hasil	Keterangan
'A' > 'B'	.F.	Menurut tabel ASCII, 'A' terletak sebelum 'B'.
		Jadi nilai 'A' lebih kecil daripada 'B"
'A' < 'a'	.T	Menurut tabel ASCII, 'A' terletak sebelum 'a'

2 > 1	.T	Nilai 2 memang lebih besar dari 1
"Yogya" = "YOGYA"	.F	Huruf kecil berbeda dengan huruf besar
"ABCD" = "ABC"	.Т.	Kalau SET EXACT dalam keadaan OFF
"ABCD" = "ABC"	.F.	Kalau SET EXACT dalam keadaan ON
"ABCD" == "ABC"	.F.	Selalu menghasilkan .F., tidak tergantung pada
		keadaan SET EXACT
"AB" \$ "ABCD"	.Т.	String "AB" ada pada "ABCD"
"BC" \$ "ABCD"	.Т.	String "BC" ada pada "ABCD"
"ABCD" \$ "AB"	.F.	String "ABCD" tidak ada pada string "AB"

Catatan:

- Contoh diatas menggunakan operator relasi pada dua operand (di sebelah kiri dan sebelah kanan operator) yang bertipe sama.
- Bila tipe data yang ingin di bandingkan tidak sama, salah satunya harus dikonversikan sehingga kedua operand bertipe sama.
- Visual FoxPro menyediakan berbagai fungsi untuk melakukan konversi. Contoh :

□ DTOC() : untuk mengubah data bertipe tanggal menjadi karakter.

CTOD() : untuk mengubah data bertipe karakter menjadi tanggal.

STR() : untuk megkonversi nilai numeric ke string karakter.

S VAL() : untuk mengkonversi string karakter menjadi numeric.

2. Operator Logika

Digunakan untuk membandingkan dua buah ekspresi logika dan menghasilkan nilai .T. (benar) atau .F. (salah).

Tabel Operator Logika

Operator	Keterangan	Format
.NOT. atau!	Operator "Tidak"	.NOT. <ekspresi></ekspresi>
		Hasil .T. kalau <ekspresi> bernilai .F.</ekspresi>
		Hasil .F. kalau <ekspresi> bernilai .T.</ekspresi>

.AND.	Operator "Dan"	<ekspresi1> .AND. <ekspresi2></ekspresi2></ekspresi1>		
		Hasil berupa .T. hanya kalau kedua <ekspresi> bernilai .T.</ekspresi>		
.OR.	Operator "Atau"	<ekspresi1> .AND. <ekspresi2></ekspresi2></ekspresi1>		
		Hasil berupa .T. kalau ada <ekspresi> yang</ekspresi>		
		bernilai .T.		

Catatan:

- .NOT. mempunyai pengutamaan yang paling tinggi.
- AND. mempunyai pengutamaan yang lebih tinggi daripada .OR.

DIPLOMA III-AMIK BSI

Pertemuan 18

Kasus Program Tanpa Tabel

Contoh (Program tanpa menggunakan tabel):

⇒ Buatlah sebuah form dengan lay-out sbb :



⇒ Hasil program adalah sbb :



Caranya:

- 17. Aktifkan project anda.
- 18. Buatlah sebuah form baru.

19. Setelah muncul form kosong, lakukan pengaturan terhadap propertiproperti milik form dan kemudian buatlah sejumlah kontrol serta aturlah properti-propertinya sesuai dengan table berikut :

Tabel Objek dan property yang perlu disertakan & diatur.

Nama Objek	Properti	Nilai	
Form	Caption	Program Tanggal dan Jam	
Label	Caption	Informasi	
Text Box	Name	txtinformasi	
Command Button	Caption	\ <jam sekarang<="" td=""></jam>	
	Name	cmdjam	
Command Button	Caption	Tanggal Sekarang	
	Name	cmdtanggal	
Command Button	Caption	Selesai	
	Name	cmdselesai	

20. Klik 2x pada command button Jam Sekarang dan ketikkan programnya:

ThisForm.txtInformasi.value="Jam Sekarang:" + time()

21. Klik 2x pada command button Tanggal Sekarang dan ketikkan programnya:

ThisForm.txtInformasi.value="Tanggal Sekarang:" + dtoc(date())

22. Klik 2x pada command button Selesai dan ketikkan programnya:

ThisForm.Release

- 23. Jalankan program dengan mengklik icon Run [
- 24. Simpan file dengan nama Tgl.SCX di direktori form.

DIPLOMA III-AMIK BSI

Pertemuan 19

OPERASI PERCABANGAN

1. IF....ENDIF

Digunakan untuk menyeleksi data dalam mengambil suatu keputusan.

Bentuk Umum: IF kondisi

Perintah

ENDIF

Ket: Jika kondisi benar maka perintah dijalankan. Jika

kondisi salah maka perintah tidak dijalankan.

Contoh: Skip

If eof()

Pesan=messagebox("Data diakhir

record",16,"Informasi")

Go bottom

Endif

2. IF....ELSE...ENDIF

Digunakan untuk menyeleksi suatu data yang mempunyai dua pilihan untuk mengambil suatu keputusan.

Bentuk Umum: IF kondisi

Perintah1

ELSE

Perintah2

ENDIF

Ket : Jika kondisi benar maka perintah1 dijalankan. Jika kondisi salah maka perintah2 dijalankan.

Contoh: If Nilai >= 60

Hasil = "LULUS"

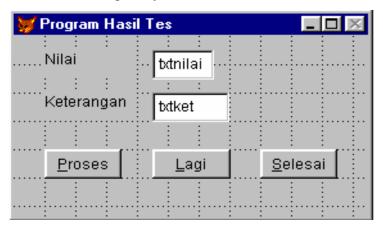
Else

Hasil = "GAGAL"

Endif

Contoh Program:

✓ Buatlah form dengan lay-out sbb :



✓ Hasil Program adalah sbb :



- ✓ Caranya:
 - 25. Aktifkan project anda.
 - 26. Buatlah sebuah form baru.

27. Setelah muncul form kosong, lakukan pengaturan terhadap properti-properti milik form dan kemudian buatlah sejumlah kontrol serta aturlah properti-propertinya sesuai dengan table berikut :

Tabel Objek dan property yang perlu disertakan & diatur.

Nama Objek	Properti	Nilai
Form	Caption	Program Hasil Tes
Label	Caption	Nilai
	Name	lblnilai
Label	Caption	Keterangan
	Name	Lblket
Text Box	Name	txtnilai
Text Box	Name	txtket
Command Button	Caption	\ <proses< td=""></proses<>
	Name	cmdproses
Command Button	Caption	\ <lagi< td=""></lagi<>
	Name	cmdlagi
Command Button	Caption	\ <selesai< td=""></selesai<>
	Name	cmdselesai

28. Klik 2x pada command button Proses dan ketikkan programnya:

Nilai = val(thisform.txtnilai.value)

If Nilai ≥ 60

ket = "Lulus"

else

ket = "Gagal"

endif

thisform.txtket.value=ket

29. Klik 2x pada command button Lagi dan ketikkan programnya:

thisform.txtnilai.value=""

thisform.txtket.value=""

thisform.txtnilai.setfocus()

30. Klik 2x pada command button Selesai dan ketikkan programnya:

ThisForm.Release

31. Jalankan program dengan mengklik icon Run [



32. Simpan file dengan nama Lulus.SCX di direktori form.

3. IF MAJEMUK (IF Bertingkat / NESTED IF)

Digunakan untuk menyeleksi suatu data yang mempunyai banyak pilihan untuk mengambil suatu keputusan.

```
Bentuk Umum: IF kondisi1
Perintah1
ELSE
IF kondisi2
Perintah2
ELSE
Perintah N
ENDIF
ENDIF
```

Ket: Jika kondisi1 benar maka perintah1 dijalankan. Jika kondisi1 salah dan kondisi2 benar maka perintah2 dijalankan. Jika kondisi2 salah maka perintah N dijalankan.

```
Contoh: If Nilai >= 90

Grade = "A"

Else

If Nilai >= 70

Grade = "B"

Else

If Nilai >= 60

Grade = "C"

Else

If Nilai >= 50

Grade = "D"

Else

Grade = "E"

Endif
```

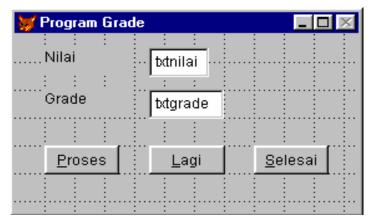
Endif

Endif

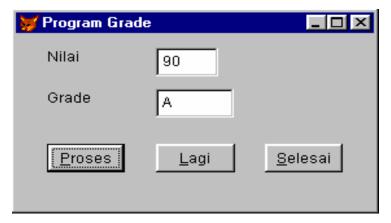
Endif

Contoh Program:

✓ Buatlah form dengan lay-out sbb :



✓ Hasil Program adalah sbb :



✓ Caranya:

- 1.Buatlah sebuah form baru.
- 2.Setelah muncul form kosong, lakukan pengaturan terhadap properti-properti milik form dan kemudian buatlah sejumlah kontrol serta aturlah properti-propertinya sesuai dengan table berikut:

Tabel ... Objek dan property yang perlu disertakan & diatur.

Nama Objek	Properti	Nilai
Form	Caption	Program Grade
Label	Caption	Nilai
	Name	lblnilai

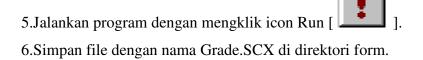
Label	Caption	Grade
	Name	Lblgrade
Text Box	Name	txtnilai
Text Box	Name	Txtgrade
Command Button	Caption	\ <proses< td=""></proses<>
	Name	cmdproses
Command Button	Caption	\ <lagi< td=""></lagi<>
	Name	cmdlagi
Command Button	Caption	\ <selesai< td=""></selesai<>
	Name	cmdselesai

3.Klik 2x pada command button Proses dan ketikkan programnya:

```
Nilai = val(thisform.txtnilai.value)
if Nilai >= 90
    grade = ''A''
else
   if Nilai  = 70 
       grade = "B"
 else
      if Nilai >= 60
          grade = "C"
      else
          if Nilai >= 50
                grade = "D"
          else
               grade = "E"
          endif
       endif
  endif
endif
thisform.txtgrade.value=grade
```

4.Klik 2x pada command button Selesai dan ketikkan programnya :

ThisForm.Release



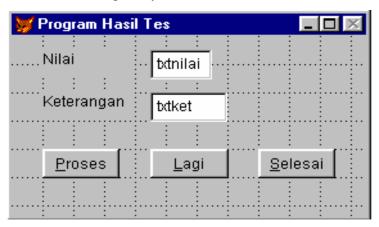
DIPLOMA III-AMIK BSI

Pertemuan 20

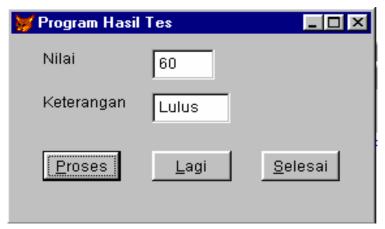
OPERASI PERCABANGAN LANJUTAN

Contoh Program:

✓ Buatlah form dengan lay-out sbb :



✓ Hasil Program adalah sbb :



- ✓ Caranya:
 - 33. Aktifkan project anda.
 - 34. Buatlah sebuah form baru.

35. Setelah muncul form kosong, lakukan pengaturan terhadap properti-properti milik form dan kemudian buatlah sejumlah kontrol serta aturlah properti-propertinya sesuai dengan table berikut :

Tabel Objek dan property yang perlu disertakan & diatur.

Nama Objek	Properti	Nilai
Form	Caption	Program Hasil Tes
Label	Caption	Nilai
	Name	lblnilai
Label	Caption	Keterangan
	Name	Lblket
Text Box	Name	txtnilai
Text Box	Name	txtket
Command Button	Caption	\ <proses< td=""></proses<>
	Name	cmdproses
Command Button	Caption	\ <lagi< td=""></lagi<>
	Name	cmdlagi
Command Button	Caption	\ <selesai< td=""></selesai<>
	Name	cmdselesai

36. DO CASE ... ENDCASE

Fungsinya sama seperti If Majemuk dan dapat digunakan sebagai alternatif dari If majemuk.

Bentuk Umum: DO CASE

CASE kondisi1

Perintah1

CASE kondisi1

Perintah2

OTHERWISE

Perintah N

ENDCASE

Contoh: DO CASE

CASE Nilai >= 90

Grade = "A"

CASE Nilai >= 70

Grade = "B"

CASE Nilai >= 60

Grade = "C"

CASE Nilai >= 50

Grade = "D"

OTHERWISE

Grade = "E"

ENDCASE

DIPLOMA III-AMIK BSI

Pertemuan 21

OPERASI PERULANGAN

1. DO WHILEENDDO

Bentuk Umum: DO WHILE kondisi

Perintah

ENDDO

Contoh Program:

✓ Buatlah form dengan lay-out sbb :



✓ Hasil Program adalah sbb:



- ✓ Caranya:
 - 1. Aktifkan project anda.
 - 2. Buatlah sebuah form baru.

3. Setelah muncul form kosong, lakukan pengaturan terhadap properti-properti milik form dan kemudian buatlah sejumlah kontrol serta aturlah properti-propertinya sesuai dengan table berikut :

Tabel Objek dan property yang perlu disertakan & diatur.

Nama Objek	Properti	Nilai
Form	Caption	Program Total
Label	Caption	Program Menghitung Jumlah Nilai
	FontBold	.T.
	FontName	Script MT Bold
	FontSize	11
	Name	lblnilai
Label	Caption	Dari 1 s.d 100 menggunakan DoWhile
	FontBold	.T.
	FontName	Script MT Bold
	FontSize	11
	Name	lbldari
Command Button	Caption	\ <proses< td=""></proses<>
	Name	cmdproses
Command Button	Caption	\ <selesai< td=""></selesai<>
	Name	cmdselesai

4. Klik 2x pada command button Proses dan ketikkan programnya:

```
jumlah=0
bilangan=0
do while bilangan <=100
    jumlah=jumlah+bilangan
    bilangan=bilangan+1
enddo
messagebox("Jumlahnya = " + str(jumlah),"Informasi")</pre>
```

5. Kik 2x pada command button Selesai dan ketikkan programnya :

ThisForm.Release

- 6. Jalankan program dengan mengklik icon Run [
- 7. Simpan file dengan nama DoWhile.SCX di direktori form.

2. FOR ... NEXT

B.U : FOR variable memori = eksprN1 TO eksprN2 [STEP eksprN3]

Peritah

NEXT

Ket: Variabel me

Variabel memori adalah variable yang akan diberi nilai awal sama dengan eksprN1 hingga eksprN2, dengan kenaikan sebesar eksprN3 (jika STEP tidak diberikan maka kenaikan sebesar 1).

Contoh Program:

✓ Buatlah form dengan lay-out sbb :



✓ Hasil Program adalah sbb :





✓ Caranya:

- 1. Buatlah sebuah form baru.
- 2. Setelah muncul form kosong, lakukan pengaturan terhadap properti-properti milik form dan kemudian buatlah sejumlah kontrol serta aturlah properti-propertinya sesuai dengan table berikut :

Tabel Objek	dan pro	perty yang	perlu c	disertakan	& diatur.
-------------	---------	------------	---------	------------	-----------

Nama Objek	Properti	Nilai
Form	Caption	Program Total
Label	Caption	Program Menghitung Jumlah Nilai
	FontBold	.T.
	FontName	Script MT Bold
	FontSize	11
	Name	lblnilai
Label	Caption	Dari 1 s.d 100 menggunakan ForNext
	FontBold	.T.
	FontName	Script MT Bold
	FontSize	11
	Name	lbldari
Command Button	Caption	\ <proses< td=""></proses<>
	Name	cmdproses
Command Button	Caption	\ <selesai< td=""></selesai<>
	Name	cmdselesai

3. Klik 2x pada command button Proses dan ketikkan programnya :

jumlah=0

for bilangan = 1 to 100

jumlah=jumlah+bilangan

next

messagebox("Jumlahnya = " + str(jumlah),"Informasi")

4. Kik 2x pada command button Selesai dan ketikkan programnya:

ThisForm.Release

- 5. Jalankan program dengan mengklik icon Run [
- 6. Simpan file dengan nama Fornext.SCX di direktori form.

3. SCAN ...ENDSCAN

Bentuk Umum: SCAN

Perintah

ENDSCAN

Bentuk perintah diatas identik dengan:

DO WHILE .NOT. EOF()

Perintah

Skip

ENDDO

Dengan kata lain, SKIP (pindah kerecord berikutnya) dilakukan pada SCAN...ENDSCAN.

Contoh: Use Pegawai

Jumrecord = 0

SCAN

Jumrecord = jumrecord + 1

ENDSCAN

Contoh Program:

✓ Buatlah form dengan lay-out sbb :



✓ Hasil Program adalah sbb :



✓ Caranya:

- 19. Aktifkan project anda.
- 20. Buatlah sebuah form baru.
- 21. Setelah muncul form kosong, lakukan pengaturan terhadap properti-properti milik form dan kemudian buatlah sejumlah kontrol serta aturlah properti-propertinya sesuai dengan table berikut :

Tabel Objek dan property yang perlu disertakan & diatur.

Nama Objek	Properti	Nilai
Form	Caption	Menghitung Record
Shape	Name	Bingkai
Label	Autosize	.T.
	Caption	[Pilihan]
Option Group	Autosize	Pilihan
	BorderStyle	0 – none
	ButtonCount	3
	Value	0
	Pada tab 1. Button:	
	Caption:	
	Keseluruhan Data	
	Data Karyawan Pria	
	Data Karyawan Wanita	
Label	Caption	Jumlah Record
Text Box	Name	Txtjumrecord
Command Button	Caption	\ <selesai< td=""></selesai<>
	Name	cmdselesai

22. Listing Program

• Isi untuk form1 procedure Load :

set dele on

use c:\zidan\data\pegawai

Isi untuk form1 procedure unLoad :

Use

Isi untuk grup pilihan procedure Click :

go top

jumrecord=0

do case

```
case thisform.pilihan.value=1
   scan
      jumrecord=jumrecord+1
   endscan
   case this form.pilihan.value=2
      scan
         if jns_klmn='L'
            jumrecord=jumrecord+1
         endif
      endscan
   case this form.pilihan.value=3
      scan
         if jns_klmn='P'
            jumrecord=jumrecord+1
          endif
      endscan
Endcase
Thisform.txtjumlah.value = jumrecord
```

23. Kik 2x pada command button Selesai dan ketikkan programnya:

ThisForm.Release

- 24. Jalankan program dengan mengklik icon Run [].
- 25. Simpan file dengan nama Scan.SCX di direktori form.

DIPLOMA III-AMIK BSI

Pertemuan 22

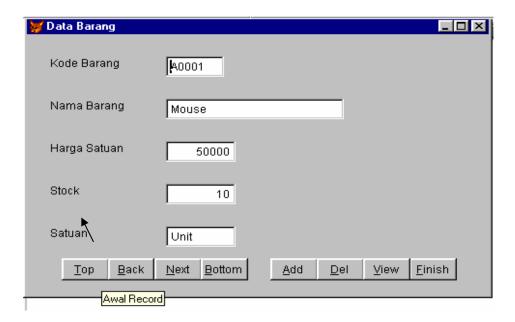
Kasus Program dengan Tabel

Contoh 2 (Program menggunakan table & control source) :

⇒ Buatlah sebuah form dengan lay-out sbb :



⇒ Hasil program adalah sbb :



Caranya:

- 1. Aktifkan project anda.
- 2. Buatlah sebuah tabel dengan struktur file sbb:

Name	Туре	Width	Decimal Inde	×
brg_kd	Character	5	t	
brg_nm	Character	25		
brg_hg	Numeric	7	0	
brg_stk	Numeric	3	0	
brg_st	Character	10		

- 3. Index file berdasarkan brg_kd dengan tipe Primary.
- 4. Isi sebanyak 5 record.
- 5. Buatlah form baru.
- 6. Klik kanan mouse.
- 7. Klik Data Environment.
- 8. Klik file Barang.DBF
- 9. Drag & drop pada field hingga ke form.
- 10. Susun bentuk form.
- 11. Setelah itu lakukan pengaturan terhadap properti-properti milik form dan kemudian buatlah sejumlah kontrol serta aturlah properti-propertinya sesuai dengan table berikut :

Tabel Objek dan property yang perlu disertakan & diatur.

Nama Objek	Properti	Nilai
Form	Caption	Data Barang
	ShowTips	.T.
Label	Caption	Kode Barang
	Name	Lblbrg_kd
Label	Caption	Nama Barang
	Name	Lblbrg_nm
Label	Caption	Harga Satuan
	Name	Lblbrg_hg
Label	Caption	Stock
	Name	Lblbrg_stk
Label	Caption	Satuan
	Name	Lblbrg_st
Text Box	ControlSource	barang.brg_kd
	Name	txtbrg_kd
	MaxLength	5
Text Box	ControlSource	barang.brg_nm
	Name	txtbrg_nm
	MaxLength	0
Text Box	ControlSource	barang.brg_hg
	Name	txtbrg_hg
	MaxLength	0
Text Box	ControlSource	barang.brg_stk
	Name	txtbrg_stk
	MaxLength	0
Text Box	ControlSource	barang.brg_st
	Name	txtbrg_st
	MaxLength	0
Command Button	Caption	\ <top< td=""></top<>
	Name	cmdtop
	ToolTipText	Awal Record
Command Button	Caption	\ <back< td=""></back<>
	Name	cmdback
	ToolTipText	Mundur
Command Button	Caption	\ <next< td=""></next<>
	Name	cmdnext
	ToolTipText	Maju
Command Button	Caption	\ <bottom< td=""></bottom<>
	Name	cmdbottom
	ToolTipText	Akhir Record
Command Button	Caption	\ <add< td=""></add<>
	Name	cmdadd

	ToolTipText	Tambah Record
Command Button	Caption	\ <del< td=""></del<>
	Name	cmddel
	ToolTipText	Hapus Record
Command Button	Caption	\ <view< td=""></view<>
	Name	cmdview
	ToolTipText	Tampilkan Record
Command Button	Caption	\ <finish< td=""></finish<>
	Name	cmdfinish
	ToolTipText	Selesai

12. Isi untuk procedure Activate untuk form1:

Set dele on

13. Isi untuk procedure Click untuk cmdtop:

Go top

```
Thisform.refresh
```

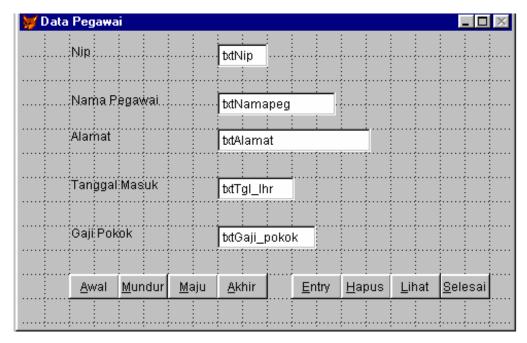
```
14. Isi untuk procedure Click untuk cmdback :
   skip -1
   if bof()
       pesan=messagebox("Data diawal record",16,"Informasi")
       go top
   endif
   thisform.refresh
15. Isi untuk procedure Click untuk cmdnext:
   skip
   if eof()
    pesan=messagebox("Data diakhir record",16,"Informasi")
    go bottom
   endif
   thisform.refresh
16. Isi untuk procedure Click untuk cmdbottom:
   go bottom
```

```
thisform.refresh
17. Isi untuk procedure Click untuk cmdadd:
   appe blan
   thisform.refresh
   thisform.txtbrg_kd.setfocus
18. Isi untuk procedure Click untuk cmddel:
   pesan=messagebox("Data akan dihapus",32+1,"Informasi")
   if pesan=1
     dele
     skip
     if eof()
       go bottom
     endif
   endif
   thisform.refresh
19. Isi untuk procedure Click untuk cmdview:
   brow
20. Isi untuk procedure Click untuk cmdfinish:
   thisform.release
21. Jalankan program dengan mengklik icon Run [ ]
```

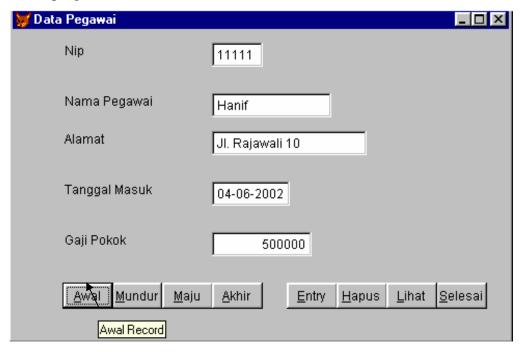
Latihan:

⇒ Buatlah sebuah form dengan lay-out sbb :

22. Simpan file dengan nama Barang.SCX di direktori form.



⇒ Hasil program adalah sbb:



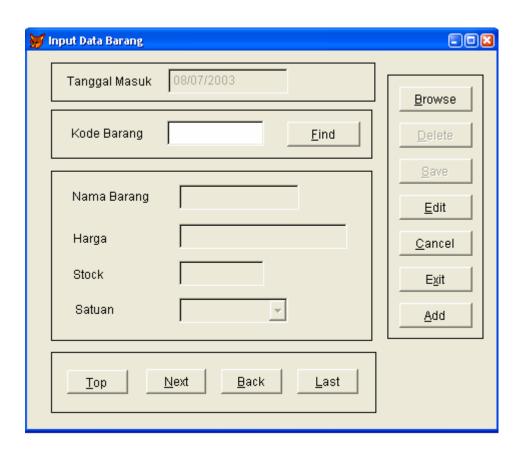
DIPLOMA III-AMIK BSI

Pertemuan 23

OPERASI RECORD

Operasi record yang dimaksud disini merupakan suatu operasi untuk menggerakan suatu record dari awal record sampai akhir record ataupun dari akhir record sampai awal record, yang sudah tersimpan didalam suatu tabel, sebagai contoh form berikut ini. Biasanya operasi record tersebut terdalam didalam suatu Command Button seperti Top, Previous, Next dan Bottom.

- Top berarti melihat diawal record
- Previous berarti melihat record selanjutnya dengan rotasi mundur
- Next berarti melihat record berikutnya dengan rotasi maju
- Bottom berarti melihat diakhir record



```
Bentuk programmnya adalah:
Tombol Top
       Deklarasi
                     : Go Top
       Contoh
                             thisform.text1.enabled=.f.
                             thisform.cmddel.enabled=.t.
                             thisform.cmdedit.enabled=.t.
                             thisform.cmdadd.enabled=.t.
                             go top
                             thisform.tampildata
                             thisform.refresh
Tombol Previous/Back
       Deklarasi
                     : Skip -1
       Contoh
                             thisform.text1.enabled=.f.
                             thisform.cmddel.enabled=.t.
                             thisform.cmdedit.enabled=.t.
                             thisform.cmdadd.enabled=.t.
                             skip -1
                             if bof()
                               go top
                               messagebox("Sudah diawal record",48,"Informasi")
                             thisform.tampildata
                             thisform.refresh
 Tombol Next
       Deklarasi
                     : Skip
       Contoh
                             thisform.text1.enabled=.f.
                             thisform.cmddel.enabled=.t.
                             thisform.cmdedit.enabled=.t.
                             thisform.cmdadd.enabled=.t.
                             skip
                             if eof()
                                go bottom
                                messagebox("Sudah diakhir record",48,"Informasi")
                             thisform.tampildata
                             thisform.refresh
 Tombol Bottom/Last
       Deklarasi
                     : Go bottom
       Contoh
                             thisform.text1.enabled=.f.
                             thisform.cmddel.enabled=.t.
                             thisform.cmdedit.enabled=.t.
                             thisform.cmdadd.enabled=.t.
                             go bottom
                             thisform.tampildata
```

thisform.refresh

DIPLOMA III-AMIK BSI

Pertemuan 24

Merancang Program Aplikasi dengan Tabel Majemuk

- Buatlah Project Baru
- Buatlah Tabel dengan Field sebagai berikut

1. Tabel Barang

Nama Fields	Tipe Data	Width
Kodebar	Character	8
Namabar	Character	20
Harga	Numeric	5
Jumbar	Numeric	2

⁻ Simpan dengan nama Barang.dbf

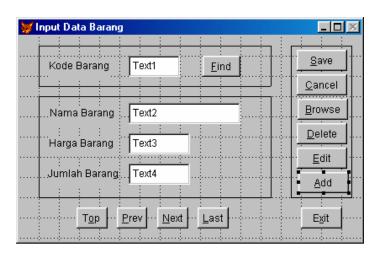
2. Tabel Customer

Nama Fields	Tipe Data	Width	
Kodecus	Character	8	
Namacus	Character	20	
Alamat	Character	20	
Telpon	Character	12	

⁻ Simpan dengan nama Customer.dbf

- Buat Tampilan Form Sebagai berikut

1. Form Barang



- Kode Program untuk Form Barang

```
PROCEDURE FORM ACTIVATE
SET DELETED ON
USE BARANG
SET ORDER TO TAG kd_barang
THISFORM.TEXT1.ENABLED=.T.
THISFORM.TEXT2.ENABLED=.F.
THISFORM.TEXT3.ENABLED=.F.
THISFORM.TEXT4.ENABLED=.F.
THISFORM.TEXT1.VALUE=" "
THISFORM.TEXT2.VALUE=" "
THISFORM.TEXT3.VALUE=" "
THISFORM.TEXT4.VALUE=" "
THISFORM.COMMAND2.ENABLED=.F.
THISFORM.COMMAND1.VISIBLE=.T.
THISFORM.COMMAND10.ENABLED=.F.
THISFORM.COMMAND11.ENABLED=.F.
THISFORM.COMMAND12.ENABLED=.F.
THISFORM.TEXT1.SETFOCUS()
PUBLIC ISI
ISI=.T.
PROCEDURE FORM LOAD
CLOSE INDEX
CLOSE DATABASE
USE
PROCEDURE FIND
IF EMPTY(THISFORM.TEXT1.VALUE)
   MESSAGEBOX ("KODE HARUS DIISI !", 48, "PERHATIAN")
   THISFORM. TEXT1. SETFOCUS ()
   RETURN
ENDIF
IF SEEK(TRIM(THISFORM.TEXT1.VALUE))
   THISFORM.TEXT1.ENABLED=.F.
   THISFORM.TEXT2.ENABLED=.F.
   THISFORM.TEXT3.ENABLED=.F.
   THISFORM.TEXT4.ENABLED=.F.
   THISFORM. TAMPILDATA
   THISFORM.COMMAND11.ENABLED=.T.
   RETURN
ENDIF
```

```
THISFORM.ATUROBJECT (.F.)
THISFORM.TEXT2.VALUE = ""
THISFORM.TEXT3.VALUE = ""
                                    Command1 = tombol find
THISFORM.TEXT4.VALUE = ""
                                    Command2 = tombol save
THISFORM.COMMAND2.ENABLED = .F.
                                    Command3 = tombol cancel
                                    Command4 = tombol browse
THISFORM.COMMAND1.VISIBLE = .T.
THISFORM. TEXT1. SETFOCUS ()
                                    Command5 = tombol exit
                                    Command6 = tombol top
THISFORM.ATUROBJECT(.T.)
THISFORM.COMMAND1.VISIBLE=.F.
                                    Command7 = tombol prev
THISFORM.COMMAND2.ENABLED=.T.
                                    Command8 = tombol next
                                    Command9 = tombol last
THISFORM.COMMAND3.ENABLED=.T.
                                    Command10 = tombol delete
THISFORM.COMMAND10.ENABLED=.F.
                                    Command11 = tombol edit
THISFORM.COMMAND12.ENABLED=.F.
                                    Command12 = tombol add
THISFORM.TEXT2.SETFOCUS()
PROCEDURE SAVE
IF EMPTY(THISFORM.TEXT2.VALUE)
   MESSAGEBOX ("NAMA HARUS DIISI !", 48, "PERHATIAN")
   THISFORM. TEXT2. SETFOCUS ()
   RETURN
ENDIF
IF EMPTY(THISFORM.TEXT3.VALUE)
   MESSAGEBOX ("HARGA HARUS DIISI !", 48, "PERHATIAN")
   THISFORM. TEXT3. SETFOCUS ()
   RETURN
ENDIF
IF EMPTY(THISFORM.TEXT4.VALUE)
   MESSAGEBOX ("JUMLAH HARUS DIISI !", 48, "PERHATIAN")
   THISFORM.TEXT4.SETFOCUS()
   RETURN
ENDIF
IF ISI=.T.
 APPEND BLANK
 REPLACE KD BARANG WITH THISFORM. TEXT1. VALUE
 REPLACE NM_BARANG WITH THISFORM.TEXT2.VALUE
 REPLACE HARGA WITH VAL (THISFORM. TEXT3. VALUE)
 REPLACE JUMLAH WITH VAL (THISFORM. TEXT4. VALUE)
ELSE
 REPLACE KD BARANG WITH THISFORM. TEXT1. VALUE
 REPLACE NM BARANG WITH THISFORM. TEXT2. VALUE
 REPLACE HARGA WITH THISFORM. TEXT3. VALUE
 REPLACE JUMLAH WITH THISFORM. TEXT4. VALUE
ENDIF
```

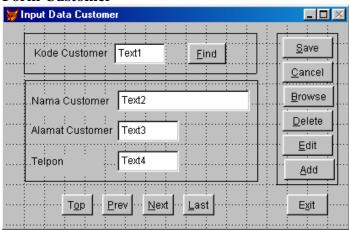
```
THISFORM.KONDISIAWAL
THISFORM.REFRESH
PROCEDURE CANCEL
THISFORM.KONDISIAWAL
PROCEDURE DELETE
IF RECCOUNT() <= 0</pre>
RETURN
ENDIF
PESAN =
             MESSAGEBOX ("BENAR DATA
                                           AKAN
                                                    DIHAPUS
?",36, "KONFIRMASI")
IF PESAN = 6
DELETE
PACK
ENDIF
THISFORM. TAMPILDATA
THISFORM.REFRESH
PROCEDURE EDIT
ISI=.F.
THISFORM.TEXT1.ENABLED=.F.
THISFORM.TEXT2.ENABLED=.T.
THISFORM.TEXT3.ENABLED=.T.
THISFORM.TEXT4.ENABLED=.T.
THISFORM.COMMAND12.ENABLED=.F.
THISFORM.COMMAND1.ENABLED=.F.
THISFORM.COMMAND2.ENABLED=.T.
THISFORM.TEXT2.SETFOCUS
THISFORM.REFRESH
PROCEDURE ADD
ISI = .T.
THISFORM.TEXT1.ENABLED=.T.
THISFORM.TEXT2.ENABLED=.F.
THISFORM.TEXT3.ENABLED=.F.
THISFORM.TEXT4.ENABLED=.F.
THISFORM.TEXT1.VALUE=""
THISFORM.TEXT2.VALUE=""
THISFORM.TEXT3.VALUE=""
THISFORM.TEXT4.VALUE=""
THISFORM.COMMAND11.ENABLED=.F.
THISFORM.TEXT1.SETFOCUS
THISFORM.REFRESH
```

PROCEDURE EXIT THISFORM. RELEASE PROCEDURE BROWSE BROWSE THISFORM.TEXT1.VALUE="" THISFORM.TEXT2.VALUE="" THISFORM.TEXT3.VALUE="" THISFORM.TEXT4.VALUE="" PROCEDURE TOP THISFORM.TEXT1.ENABLED=.F. THISFORM.COMMAND10.ENABLED=.T. THISFORM.COMMAND11.ENABLED=.T. THISFORM.COMMAND12.ENABLED=.T. GO TOP THISFORM. TAMPILDATA THISFORM.REFRESH PROCEDURE LAST THISFORM.TEXT1.ENABLED=.F. THISFORM.COMMAND10.ENABLED=.T. THISFORM.COMMAND11.ENABLED=.T. THISFORM.COMMAND12.ENABLED=.T. GO BOTTOM THISFORM. TAMPILDATA THISFORM.REFRESH PROCEDURE PREVIOUS THISFORM.TEXT1.ENABLED=.F. THISFORM.COMMAND10.ENABLED=.T. THISFORM.COMMAND11.ENABLED=.T. THISFORM.COMMAND12.ENABLED=.T. SKIP-1 IF BOF() GO TOP MESSAGEBOX ("Sudah di awal record", 48, "Perhatian") ENDIF THISFORM.TAMPILDATA THISFORM. REFRESH PROCEDURE NEXT THISFORM.TEXT1.ENABLED=.F. THISFORM.COMMAND10.ENABLED=.T. THISFORM.COMMAND11.ENABLED=.T. THISFORM.COMMAND12.ENABLED=.T.

```
SKIP
IF EOF()
 GO BOTTOM
MESSAGEBOX ("Sudah di akhir record", 48, "Perhatian")
ENDIF
THISFORM. TAMPILDATA
THISFORM. REFRESH
PROCEDURE ATUROBJECT
PARAMETER KONDISI
THISFORM.TEXT1.ENABLED = .NOT. KONDISI
THISFORM.TEXT2.ENABLED = KONDISI
THISFORM.TEXT3.ENABLED = KONDISI
THISFORM.TEXT4.ENABLED = KONDISI
PROCEDURE KONDISIAWAL
THISFORM.ATUROBJECT (.F.)
THISFORM.TEXT1.VALUE = ""
THISFORM.TEXT2.VALUE =
THISFORM.TEXT3.VALUE =
THISFORM.TEXT4.VALUE = ""
THISFORM.COMMAND2.ENABLED = .F.
THISFORM.COMMAND1.VISIBLE = .T.
THISFORM. TEXT1. SETFOCUS ()
PROCEDURE TAMPILDATA
THISFORM.TEXT1.VALUE=barang.kd_barang
THISFORM.TEXT2.VALUE=barang.nm_barang
THISFORM.TEXT3.VALUE=barang.harga
THISFORM.TEXT4.VALUE=barang.jumlah
```

- Jalankan form Barang tersebut
- Simpan Form tersebut dengan nama Barang.Frm

2. Form Customer



- Lakukan perintah seperti proses Form barang, hanya merubah nama tabel yang digunakan.
- Simpan Form tersebut dengan nama Customer.frm

DIPLOMA III-AMIK BSI

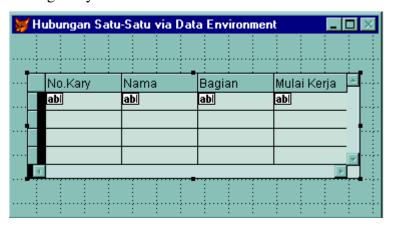
Pertemuan 25

MERANCANG PROGRAM APLIKASI MENGGUNAKAN TABEL MAJEMUK DENGAN DATA ENVIRONMENT

3. Hubungan Satu-Satu pada Data Environment.

Seorang programmer dapat menggunakan lingkungan data (*data environment*) bila ingin membuat sebuah form yang menggunakan sejumlah tabel. Dengan cara seperti ini tabel-tabel yang dilibatkan akan dibuka secara otomatis dan juga akan ditutup secara otomatis ketika form berakhir. Programmer tidak perlu menuliskan kode untuk membuat hubungan antartabel sebagaimana yang biasa dilakukan pada kejadian **Load** milik form. Contoh program:

✓ Buatlah form dengan lay-out sbb :



✓ Hasil Program adalah sbb :



✓ Caranya :

- 26. Aktifkan project anda.
- 27. Buat / aktifkan database personal.
- 28. Pastikan bahwa tabel pribadi, tabel atrikary, dan tabel bagian telah aktif.
- 29. Buatlah sebuah form baru.
- 30. Setelah muncul form kosong, lakukan pengaturan terhadap properti-properti milik form dan kemudian buatlah sejumlah kontrol serta aturlah properti-propertinya sesuai dengan table berikut :

Tabel Objek dan property yang perlu disertakan & diatur.

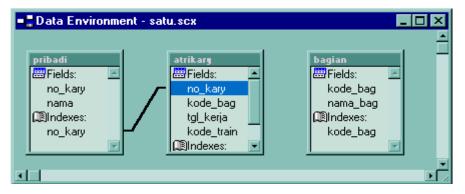
Nama Objek	Properti	Nilai
Form	Caption	Hubungan Satu-Satu via Data
		Environment.
Grid	ColumnCount	4
	Height	169
	Left	15
	Name	GridInformasi
	ReadOnly	.T.
	RecordSource	atrikary
	RecordSource Type	1
	Тор	38
	Width	343
Column	Name	Column 1
(bagian dari Grid)	ControlSource	Atrikary.no_kary
Header	Name	Header 1
(bagian Column 1)	Caption	No. Kary.
Column	Name	Column 2

(bagian dari Grid)	ControlSource	Pribadi.nama
Header	Name	Header 1
(bagian Column 2)	Caption	Nama
Column	Name	Column 3
(bagian dari Grid)	ControlSource	Bagian.nama_bag
Header	Name	Header 1
(bagian Column 3)	Caption	Bagian
Column	Name	Column 4
(bagian dari Grid)	ControlSource	Atrikary.tgl_kerja
Header	Name	Header 1
(bagian Column 4)	Caption	Mulai Kerja

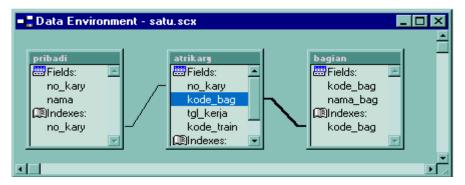
- 31. Klik kanan mouse pada bagian yang tidak ada obyeknya.
- 32. Pilih Data Environment
- 33. Klik double pribadi
- 34. Klik double atrikary
- 35. Klik double bagian
- 36. Klik Close



37. Klik pada field no_kary milik ATRIKARY dan gerakkan mouse ke no_kary pada bagian indexes milik tabel PRIBADI.



38. Klik pada field kode_bag milik ATRIKARY dan gerakkan mouse ke kode_bag pada bagian indexes milik tabel PRIBADI.



- 39. Tutup jendela Data Environment
- 40. Isi untuk form procedure Load:

SET DELETED ON

{ *Ket* : agar record yang terhapus tidak diproses }

41. Isi untuk form procedure Activate:

SELECT atrikary

{ Ket : menjadikan ATRIKARY sebagai tabel sekarang ketika form diaktifkan }

- 42. Jalankan program dengan mengklik icon Run [🛂].
- 43. Simpan file dengan nama Satu.SCX di direktori form.

4. Hubungan Satu-Banyak pada Data Environment.

Contoh kasus : seseorang dapat mengikuti banyak pelatihan.

Contoh program:

✓ Buatlah form dengan lay-out sbb :



✓ Hasil Program adalah sbb :



✓ Caranya :

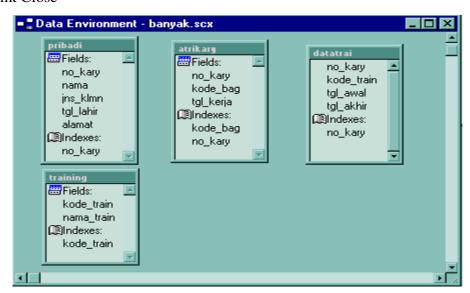
- 24. Aktifkan project anda.
- 25. Buat / aktifkan database personal.
- 26. Pastikan bahwa tabel pribadi, tabel atrikary, tabel datatrai dan tabel training telah aktif.
- 27. Buatlah sebuah form baru.
- 28. Setelah muncul form kosong, lakukan pengaturan terhadap properti-properti milik form dan kemudian buatlah sejumlah kontrol serta aturlah properti-propertinya sesuai dengan table berikut :

Tabel Objek dan property yang perlu disertakan & diatur.

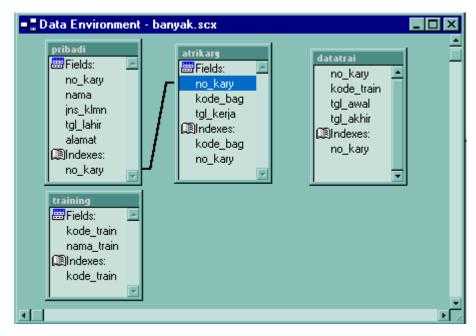
Nama Objek	Properti	Nilai
Form	Caption	Hubungan Satu-Banyak via
	_	Data Environment.
Grid	ColumnCount	4
	Name	GridInformasi
	ReadOnly	.T.
	RecordSource	atrikary
	RecordSource Type	1
Column	Name	Column 1
(bagian dari Grid)	ControlSource	Atrikary.no_kary
Header	Name	Header 1
(bagian Column 1)	Caption	No. Kary.
Column	Name	Column 2
(bagian dari Grid)	ControlSource	Pribadi.nama
Header	Name	Header 1
(bagian Column 2)	Caption	Nama
Column	Name	Column 3
(bagian dari Grid)	ControlSource	Training.nama_train
Header	Name	Header 1

(bagian Column 3)	Caption	Pelatihan
Column	Name	Column 4
(bagian dari Grid)	ControlSource	datatrai.tgl_awal
Header	Name	Header 1
(bagian Column 4)	Caption	Tanggal

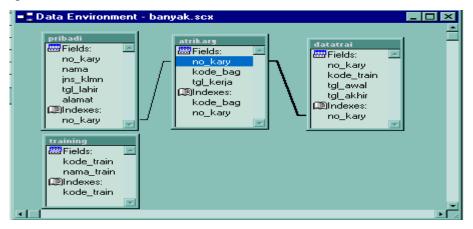
- 29. Klik kanan mouse pada bagian yang tidak ada obyeknya.
- 30. Pilih Data Environment
- 31. Klik double pribadi
- 32. Klik double atrikary
- 33. Klik double datatrai
- 34. Klik double training
- 35. Klik Close



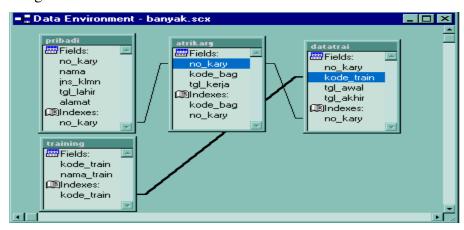
36. Klik pada field no_kary milik ATRIKARY dan gerakkan mouse ke no_kary pada bagian indexes milik tabel PRIBADI.

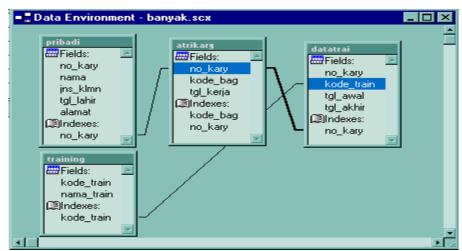


37. Klik pada field no_kary milik ATRIKARY dan gerakkan mouse ke no_kary pada bagian indexes milik tabel DATATRAI



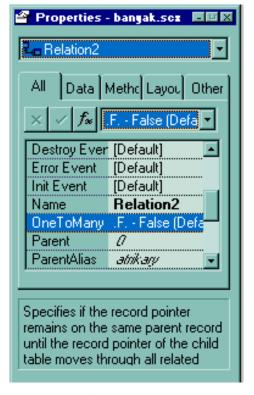
38. Klik pada field kode_train milik DATATRAI dan gerakkan mouse ke kode_train pada bagian indexes milik tabel TRAINING





39. Klik pada garis penghubung antara tabel ATRIKARY dan DATATRAI

40. Langkah ini akan mengakibatkan jendela properties menampilkan properti milik garis penghubung tersebut.



- 41. Gantilah nilai .F. pada properti **OneToMany** menjadi .T. (nilai .T. inilah yang menyatakan bahwa hubungan berupa satu-banyak).
- 42. Tutup jendela Data Environment
- 43. Isi untuk form procedure Load:

SET DELETED ON

44. Isi untuk form procedure Activate:

SELECT atrikary

- 45. Jalankan program dengan mengklik icon Run [🗓].
- 46. Simpan file dengan nama Banyak.SCX di direktori form.

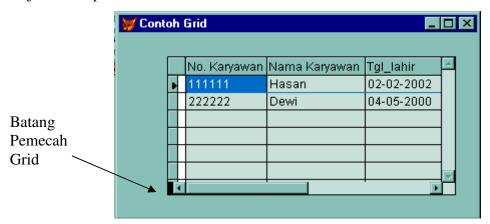
DIPLOMA III-AMIK BSI

Pertemuan 26

DATA GRID

1. Mengenal Objek Data Grid.

Grid adalah suatu objek yang memungkinkan sejumlah record data ditampilkan pada bentuk tabel. Grid tergolong sebagai kontainer yang dapat berisi sejumlah objek khusus yang disebut kolom. Dalam hal ini setiap kolom dapat memiliki judul (objek header) dan objek lain seperti kotak teks atau kotak cek.



Baik grid maupun objek yang berada di dalamnya memiliki properti, kejadian, dan metode tersendiri.

Tabel Properti dan Metode pada Grid

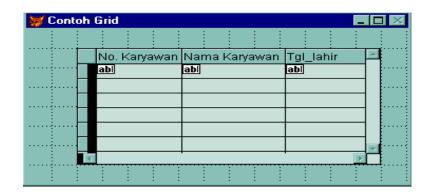
Properti dan Metode	Keterangan	
ColoumnCount	Properti ini menentukan jumlah kolom pada grid.	
	Default sama dengan -1 (artinya semua field akan	
	ditampilkan).	
ReadOnly	Properti ini digunakan untuk menentukan dapat tidaknya	
	data diubah pada grid.	

	.T. = data hanya dapat dibaca		
	.F. = data dapat diubah		
RecordSourceType	Properti ini menentukan macam sumber record untuk		
	grid.		
	0 = Tabel. Tabel yang disebutkan pada properti.		
	RecordSource akan dibuka secara otomatis.		
	1 = Alias. Alias yang disebutkan pada properti.		
	RecordSource akan dijadikan sebagai sumber		
	record untuk grid.		
	2 = Prompt. Pemakai akan diminta untuk		
	menyebutkan sumber record untuk grid saat form		
	dieksekusi.		
	3 = Query (.QPR). File .QPR yang disebutkan pada		
	properti RecordSource akan dijadikan sebagai		
	sumber record untuk grid.		
	4 = pernyataan SQL. Pernyataan SQL yang		
	dicantumkan pada properti RecordSource akan		
	dijadikan sebagai sumber record untuk grid.		
RecordSource	Properti ini menentukan sumber record untuk grid.		
AllowHeaderSizing	Properti ini memungkinkan pengubahan tinggi judul		
	saat pengeksekusian.		
	.T. = judul dapat diubah ketika dieksekusi		
	.F. = judul tidak dapat diubah ketika dieksekusi		
AllowRowSizing	Properti ini memungkinkan pengubahan tinggi baris		
	record saat pengeksekusian.		
	.T. = baris dapat diubah ketika dieksekusi		
	.F. = baris tidak dapat diubah ketika dieksekusi		
DeleteMark	Properti ini menentukan tanda record yang dihapus		
	ditampilkan atau tidak pada grid.		
	.T. = tanda record dihapus ditampilkan		
	.F. = tanda record dihapu tidak ditampilkan.		

SplitBar	Properti ini menentukan batang pemecah grid		
	ditampilkan atau tidak.		
	.T. = batang pemecah ditampilkan		
	.F. = batang pemecah tidak ditampilkan		
AddObject	Metode ini berguna untuk menambahkan objek pada		
	grid.		
AddColumn	Metode ini berguna untuk menambahkan kolom pada		
	grid.		

Contoh 1:

✓ Buatlah form dengan lay-out sbb :

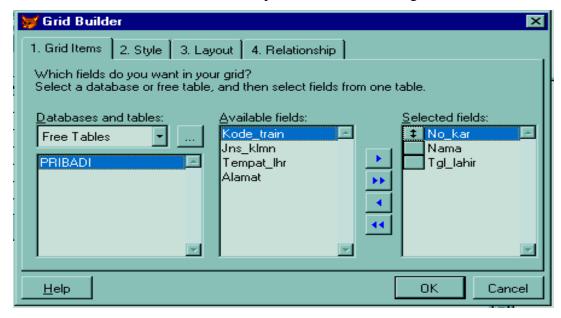


✓ Hasilnya adalah sbb :

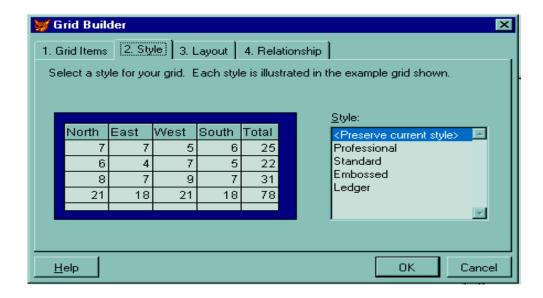


- 1. Buat form baru.
- 2. Klik kontrol grid.

- 3. Drag & drop kedalam form.
- 4. Klik kanan mouse pada grid.
- 5. Pilih <u>B</u>uilder pada menu pop-up.
- 6. Klik tombol
- 7. Pilih Pribadi.dbf
- 8. Ok
- 9. Pada Available fields, klik 2x pada No_kar, Nama, Tgl_lahir



10. Klik tab Style untuk menentukan bentuk grid (Preserve current style, Profesional, Standard, Embossed, Ledger).



11. Klik tab Layout untuk mengatur kolom.



- 12. Untuk mengubah judul kolom no_kar, klik kolom tersebut dan kemudian gantilah kotak teks di sebelah kanan tulisan Caption dengan No.Karyawan.
- 13. Lakukan langkah yang sama untuk kolom Nama Karyawan dan Tgl.Lahir
- 14. OK
- 15. Pada jendela properti, aturlah agar properti **ReadOnly** milik grid bernilai .T. (untuk mencegah pengubahan data pada grid).
- 16. Jalankan form dengan mengklik icon Run [...]].
- 17. Simpan file dengan nama Grid1.SCX di direktori form.

2. Mengurutkan Data pada Grid

Pemakai dapat menampilkan data PRIBADI.DBF urut menurut no_kar dengan cara sebagai berikut :

- 1. Aktifkan form grid1.scx
- 2. Menu File, Save As, isikan GRID2, Ok
- 3. Ketikkan untuk Form1 procedure Activate:

SET ORDER TO TAG no_kar

4. Jalankan form.

3. Mengakses Elemen Grid Via Kode.

Contoh 2:

✓ Buatlah form dengan lay-out sbb :



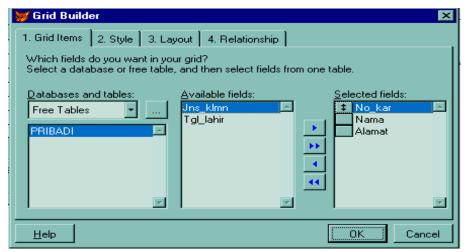
✓ Hasilnya adalah sbb :



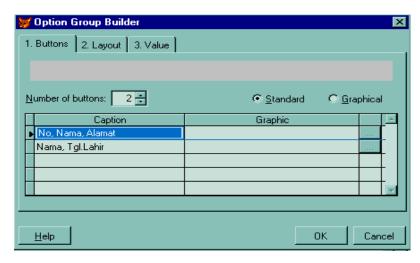
- 1. Buat form baru.
- 2. Klik kontrol grid.
- 3. Drag & drop kedalam form.
- 4. Klik kanan mouse pada grid.
- 5. Pilih <u>B</u>uilder pada menu pop-up.



- o. Trink tollibor
- 7. Pilih Pribadi.dbf
- 8. Ok
- 9. Pada Available fields, klik 2x pada No_kar, Nama, Alamat



- 10. Klik kontrol option group
- 11. Drag & drop kedalam form.
- 12. Klik kanan mouse pada option group.
- 13. Pilih <u>B</u>uilder pada menu pop-up.
- 14. Klik pada option1 dan ganti menjadi : No, Nama, Alamat
- 15. Klik pada option2 dan ganti menjadi : Nama, Tgl.Lahir



16. Lakukan pengaturan terhadap properti-properti milik form dan objek sesuai dengan table berikut :

Tabel Objek dan property yang perlu disertakan & diatur.

Nama Objek	Properti	Nilai
Form	Caption	Contoh Grid Dengan Kolom
		yang Bervariasi
Grid	AllowHeaderSizing	.F.
	AllowRowSizing	.F.
	Delete Mark	.F.
	Height	144
	Left	24
	Name	GridData
	ReadOnly	.T.
	RecordSourceType	1
	RecordSource	pribadi
	SplitBar	.F.
	Top	24
	Width	325
Option Group	Autosize	.T.
	ButtonCount	2
	Left	24
	Name	Gruppilihandata
	Top	195
	Value	1

17. Listing Program:

Isi untuk form1 procedure Load :

set dele on

use c:\zidan\data\pribadi.dbf set order to tag no_kar

• Isi untuk form1 procedure unLoad :

use

• Isi untuk grup pilihan procedure InteractiveChange:

do case

 $case\ this form. grouppilihandata. value = 1$

thisform.griddata.columncount=3

thisform.griddata.column1.header1.caption="No."

thisform.griddata.column2.header1.caption="Nama" thisform.griddata.column3.header1.caption="Alamat"

thisform.griddata.column1.controlsource="no_kar" thisform.griddata.column2.controlsource="nama"

thisform.griddata.column3.controlsource="alamat"

thisform.griddata.column1.width=60

this form. grid data. column 2. width = 100

this form. gridd at a. column 3. width = 250

 $case\ this form.grouppilihandata.value = 2$

this form.grid data. Column Count = 2

this form. grid data. column 1. header 1. caption = "Nama"

thisform.griddata.column2.header1.caption="Tgl. Lahir"

this form. grid data. column 1. control source = "nama"

 $this form. grid data. column 2. control source = "tgl_lahir"$

thisform.griddata.column1.width=100

thisform.griddata.column2.width=90

endcase

- 18. Jalankan program dengan mengklik icon Run [19.].
- 19. Simpan file dengan nama Grid3.SCX di direktori form.

4. Penggunaan SET FILTER

Contoh 3:

✓ Buatlah form dengan lay-out sbb :



✓ Hasilnya adalah sbb :



- 1. Buat form baru.
- 2. Setelah muncul form kosong, lakukan pengaturan terhadap properti-properti milik form dan kemudian buatlah sejumlah kontrol serta aturlah properti-propertinya sesuai dengan table berikut:

Tabel Objek dan property yang perlu disertakan & diatu	Tabel Obie	k dan r	property	vang perlu	disertakan (& diatur
--	-------------------	---------	----------	------------	--------------	----------

Nama Objek	Properti Nilai		
Form	Caption	Contoh Grid Dengan Kolom	
	yang Bervariasi		
Grid	Name	GridData	
	ReadOnly	.T.	
	RecordSourceType 1		
	RecordSource	pribadi	
Option Group	Autosize .T.		
	ButtonCount	3	
	Name Gruppilihankary		
	Value 1		
	Dengan menggunakan Builder aturlah agar tulisan		
	untuk tombol radio berupa:		
	Semua Karyawan		
	Karyawan Pria		
	Karyawan Wanita		

3. Listing Program:

• Isi untuk form1 procedure Load:

set dele on

use c:\zidan\data\pribadi

set order to tag no_kar

Isi untuk form1 procedure unLoad :

use

Isi untuk grup pilihan procedure InteractiveChange :

do case

case this form.gruppilihankary.value=1

set filter to

case this form.gruppilihankary.value=2

set filter to jns_klmn=.t.

case this form.gruppilihankary.value=3

set filter to jns_klmn=.f.

endcase

thisform.griddata.refresh()

4. Jalankan program dengan mengklik icon Run [].

5. Simpan file dengan nama Grid4.SCX di direktori form.

5. Menampilkan Kotak Cek pada Grid

Contoh 4:

✓ Buatlah form dengan lay-out sbb :

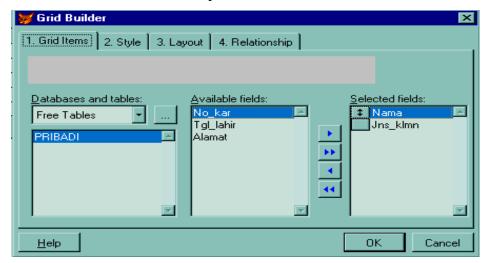


✓ Hasilnya adalah sbb :

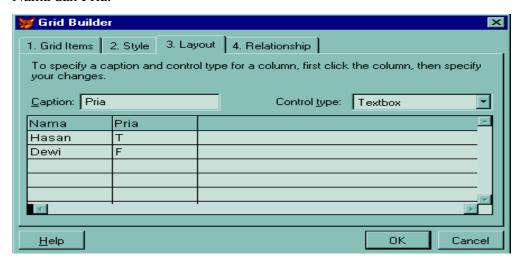


- 1. Buat form baru.
- 2. Klik kontrol grid.
- 3. Drag & drop kedalam form.
- 4. Klik kanan mouse pada grid.
- 5. Pilih <u>Builder pada menu pop-up.</u>

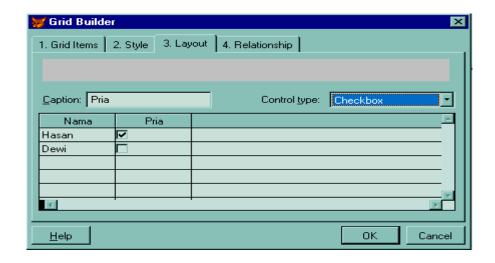
- 6. Klik tombol ...
- 7. Pilih Pribadi.dbf
- 8. Ok
- 9. Pada <u>A</u>vailable fields, klik 2x pada Nama, Jns_klmn



10. Klik tab Layout dan kemudian aturlah pada <u>C</u>aption agar judul kolom berupa Nama dan Pria.

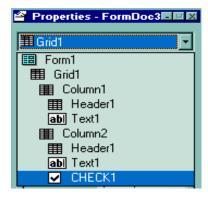


11. Pilihlah kolom Pria dan kemudian pada Control type pilihlah Check Box.



12. OK

13. Pada jendela properti, kliklah Check1 dibawah Column2.



- 14. Ubah properti ReadOnly milik Grid1 menjadi .T. (agar data tidak dapat diubah).
- 15. Jalankan form dengan mengklik icon Run [
- 16. Simpan file dengan nama Grid5.SCX di direktori form

6. Membuat Field Komputasi

Field komputasi adalah field pada grid yang diperoleh melalui suatu ekspresi. Contoh : pemakai dapat menampilkan informasi Pria atau Wanita berdasarkan field jns_klmn.

Contoh 5:

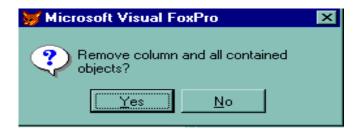
✓ Buatlah form dengan lay-out sbb :



✓ Hasilnya adalah sbb :



- 1. Aktifkan grid5
- 2. Menu File, Save As, isikan Grid6.scx, Ok.
- 3. Ganti caption form1 dengan Contoh Grid dengan Field Komputasi.
- 4. Pada jendela properties, pilihlah column2 pada grid1.
- 5. Klik pada kolom berjudul Pria dan kemudian tekan DEL. Dilayar monitor akan tampil :



- 6. Klik Yes
- 7. Pada jendela properties, pilihlah Grid1 dan ganti ColoumnCount menjadi 2.
- 8. Enter
- 9. Pada jendela properties, pilihlah Header pada Column2 pada Grid1. Gantilah Caption menjadi KELAMIN.
- 10. Isikan pada ControlSource milik Column2:

IIF(pribadi.jns_klmn=.T.,"Pria","Wanita")

- 12. Simpan file dengan nama Grid6.SCX di direktori form.

Bahasa Pemrograman Visual Foxpro 6

DIPLOMA III-AMIK BSI

Pertemuan 27

PEMBUATAN LAPORAN

5. Pengantar Laporan

Laporan biasa digunakan pada aplikasi untuk menampilkan data atau ringkasan data dalam bentuk dokumen tercetak. Visual FoxPro menyediakan beberapa fasilitas untuk membuat laporan :

- a. Quick Report
- b. Report Designer
- c. Report Wizard

Masing-masing fasilitas memiliki keunggulan dan kelemahan sendiri-sendiri. Report Designer adalah yang paling luwes karena dengan fasilitas ini pemakai dapat sepenuhnya mengontrol tata letak laporan. Report Designer inilah yang akan dibahas pada pertemuan berikut ini. Secara garis besar, tahapan dalam pembuatan laporan berupa .

- ✓ Menentukan sumber data, dan
- ✓ Mengatur tata letak laporan.

6. Pembuatan Laporan.

Caranya:

- 1. Aktifkan database PERSONAL.DBC
- 2. Pastikan tabel PRIBADI telah aktif
- 3. Dari Command Window:

CREATE REPORT latihan

Dari menu:

File, New, Report, New File

4. Klik Menu View.

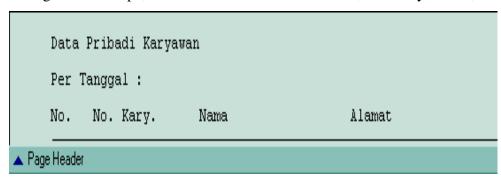
- 5. Pilih Data Environment.
- 6. Klik kanan mouse.
- 7. Pilih **<u>A</u>dd ...**
- 8. Klik double Pribadi.
- 9. Close.
- 10. Tutup jendela Environment.

Langkah 4 – 10 untuk menentukan sumber data.

11. Aktifkan Report Controls: View, Report Controls Toolbar



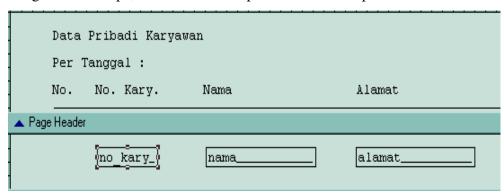
- 12. Klik Label [A] pada report controls.
- 13. Klik di bagian kiri atas Page Header
- 14. Ketikkan : Data Pribadi Karyawan
- 15. Dengan langkah serupa diatas, ketikkan dibawah teks tersebut : Per Tanggal
- 16. Dengan cara serupa, berturut-turut ketikkan teks : No, No. Kary, Nama, Alamat



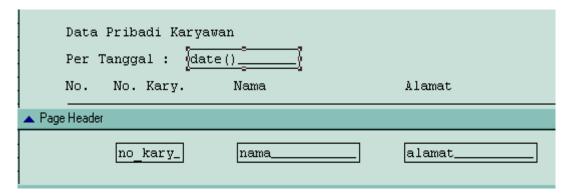
- 18. Drag & drop di bawah No. sampai Alamat
- 19. Klik Field [] pada report controls.
- 20. Klik di bawah Page Header pada kolom No. Kary.



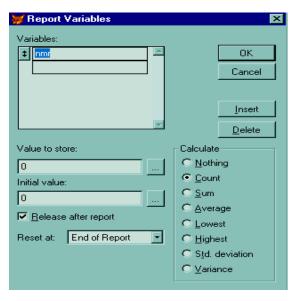
- 21. Klik tombol ____ yang terletak di sebelah kanan kotak teks <u>E</u>xpression.
- 22. Pada kotak dialog Expression Builder, klik ganda field pribadi.no_kary
- 23. Ok
- 24. Ok
- 25. Dengan cara serupa, tambahkan field pribadi.nama dan pribadi.alamat



- 26. Klik Field [] pada report controls.
- 27. Klik di sebelah kanan tulisan Per Tanggal:
- 28. Klik tombol yang terletak di sebelah kanan kotak teks <u>E</u>xpression.
- 29. Ketikkan pada kotak dialog Expression for Field on Report : **DATE**()
- 30. Ok
- 31. Ok



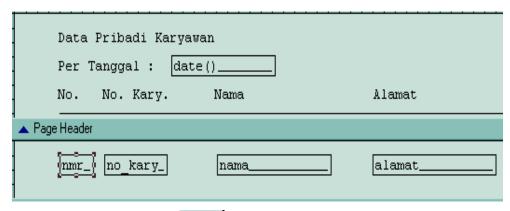
- 32. Klik menu Report
- 33. Pilih Variables



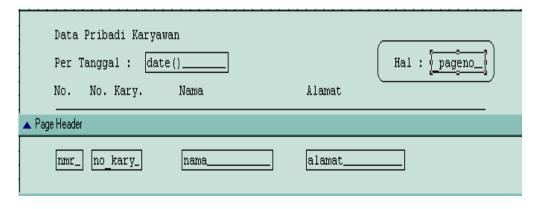
- 34. Ketikkan nama Nmr pada kotak daftar variables
- 35. Klik tombol radio Count
- 36. Klik kotak cek Release after report (untuk menghapus variabel ketika laporan berakhir).
- 37. Ok
- 38. Klik Field [] pada report controls.
- 39. Klik di sebelah bawah kolom No. pada bagian Detail
- 40. Klik tombol ____ yang terletak di sebelah kanan kotak teks <u>E</u>xpression.
- 41. Ketikkan pada kotak dialog Expression for Field on Report : Nmr
- 42. Ok

43. Ok

44. Ubah ukuran field Nmr



- 45. Klik Rounded Rectangle [] pada report controls.
- 46. Drag & drop di kanan atas report
- 47. Buatlah label Hal:
- 48. Klik Field [] pada report controls.
- 49. Klik di sebelah kanan Hal:
- 50. Pada saat muncul kotak dialog Report Expression, ketiklah pada kotak teks Expression: _pageno { _pageno adalah variabel milik Visual FoxPro yang khusus digunakan untuk mencatat nomor halaman }
- 51. Ok



- 52. Simpan laporan: File, Save
- 53. Tampilkan laporan: File, Print Preview

Data Pribadi Kary	awan	
Per Tanggal : 05	-29-2002	Hal: 1
No. No. Kary.	Nama	Alamat
1 111111	Hanif	Jl. Batujajar 7
2 222222	Dewi	Jl. Kramat 5

Bahasa Pemrograman Visual Foxpro 6

DIPLOMA III-AMIK BSI

Pertemuan 28

MENU DESIGNER VISUAL FOXPRO

Menu designer adalah tools yang disediakan oleh Visual FoxPro untuk membuat system menu sesuai dengan system Windows. Menu designer juga dapat dipakai untuk membuat menu shortcut. Extention Menu yaitu **.Mnx** setelah compile menjadi **.Mpr** Menu Visual FoxPro adalah menu Windows bukan object dari FoxPro. Bagian-bagian dari Menu Visual FoxPro, yaitu :

- Menu Bar : level menu paling atas, itemsnya akan tampak saat aplikasi berjalan.

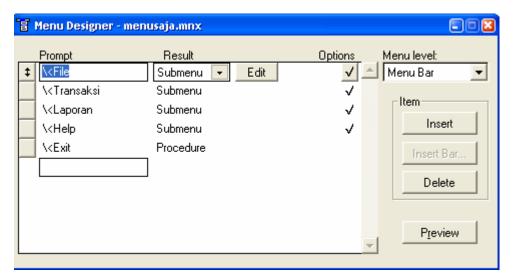
- Menu Pad : level dibawah menu bar (sub menu).

- Menu PopUp : daftar pilihan vertikal didalam menu pad

- Prompt Menu : caption yang ditampilkan pada sebuah menu popup

○ Cara memanggil Menu Designer

- 1. Pilih menu File dari menu bar Visual FoxPro, Pilih New, Pilih Menu atau
- 2. Dengan perintah Create Menu dari Jendela Command atau
- 3. Dari Project Manager, Pilih **Tab Other**, lalu pilih Menus dan Tombol New



Membuat Menu Project

Sistem menu dapat diciptakan dengan fasilitas Menu Designer. Adapun langkah-langkahnya sebagai berikut:

- I. Membuat menu tampilan:
 - 1. Klik tab Other lalu pilih item $\mathbb{E}^{\text{Menus}} > \underline{\text{New}} > \text{Menu.}$ Jendela Menu Designer akan muncul.
 - 2. Ketikkan berturut-turut menu yang akan muncul di menubar sebagai menu utama pada kolom Prompt
 - 3. Setelah menu utama terbentuk, buatlah submenu dari menu utama dengan mengklik tombol Create yang ada disamping kanan kolom Result.
 - 4. Ketikkan berturut-turut di kolom Prompt, menu yang akan muncul sebagai submenu dari menu utama
 - 5. Pada kolom Result pilih Command.
 - 6. Ketikkan pada kotak di sebelah kanan kolom Result kode berikut :
 - DO FORM "c:\nama path\nama form.scx"
 - Kode di atas akan memanggil form aplikasi yang ingin dijadikan sebagai subprogram dari program menu.
 - 7. Sisipkan 1 menu pada menu utama dengan judul "Selesai", dan ketikkan kode berikut:

APPLICATION.ACTIVEFORM.RELEASE()

- 8. Jadikan menu ini sebagai menu teratas, dengan cara memilih menubar <u>V</u>iew > General Option... dan kemudian klik kotak cek Top-Level Form, serta kliklah tombol OK.
- 9. Simpanlah menu ini dengan memilih menubar \underline{F} ile > Save \underline{A} s Beri nama lalu klik OK
- 10. Klik tombol Preview untuk melihat hasil menu yang baru saja dibuat.

II. Membangkitkan kode menu:

1. Klik menubar <u>Menu > Generate</u>. Langkah ini akan memunculkan jendela Generate Menu dimana file menu yang baru saja dibuat akan tampil secara otomatis di kolom Output <u>F</u>ile dengan file ekstensi bernama .MPR.

Klik tombol Generate. Lalu tutuplah jendela Menu Designer

- **⇒** Gabungan Sistem Menu dengan Program Utama
 - 1. Klik tab Code lalu pilih item > New > maka Jendela Program Utama akan muncul.
 - 2. Ketikkan berturut-turut program berikut ini

```
set talk off
set multilocks on
set century on
set date italian
set stat off
```

set defa to c:\Wuri\menus >> mengaktifkan **menu** designer

do menusaja.mpr

>> memanggil menu

_SCREEN.CAPTION="APLIKASI PENJUALAN BARANG " >> nama Aplikasi

```
_screen.picture="j0148788.jpg" >> gambar dimuka windows
on shutdown quit
_screen.windowstate=2
read event
```

- 3. Setelah program utama terbentuk, klik kanan lalu pilih Set Main untuk menentukan bahwa program utama ini akan berjalan pertama kali (program inti).
- 4. Setelah itu coba Run program utamanya

Bahasa Pemrograman Visual Foxpro 6

DIPLOMA III-AMIK BSI

Pertemuan 29

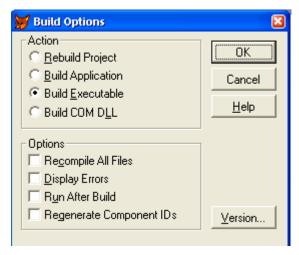
PROSES COMPILE VISUAL FOXPRO

Membuat File .APP dan .EXE

Membuat file .APP dan .EXE adapt dengan mudah dilakukan karena Visual FoxPro mempunyai fasilitas komplit yang pernah ada dan penggunaannya juga mudah.

Langkahnya adalah:

- Pilih Tab Code
- Klik Tombol Build disebelah kanan dari Project Manager, maka akan tampil kotak Build Options seperti berikut ini.



Pilihan-pilihan Action:

- 1. **Rebuild Project**, membuat source code dan memeriksa kesalahan untuk semua file yang ada didalam Project
- 2. **Build Aplication**, melakukan hal yang sama seperti Rebuild Project dengan mejadi sebuah file .APP
- 3. Build Executable, membuat file Project menjadi file extention .EXE
- 4. **Build Com DLL**, membuat DLL atau file system berdasarkan informasi class dari Project.

Pilihan-Pilihan Options:

- 1. **Recompile All File**, jika tidak diaktifkan maka Visual FoxPro hanya akan membangun semua file yang telah dimodifikasi saja.
- 2. **Display Errors**, menampilkan daftar kesalahan pada saat proses pembuatan program berlangsung.
- 3. Run After Build, menjalankan program setelah proses build selesai.
- 4. **Regenerate Component IDs**, membuat ulang Id komponen.

Untuk membentuk suatu file .exe atau mengcompile program didalam Visual FoxPro, langkah yang paling umum dilakukan adalah dengan pilihan berikut ini :

- 1. Pilihan Actions >> **Build Executable**
- 2. Pilihan Options >> Recompile All File dan Display Error
- 3. Stelah pilihan terebut diatas maka kita sudah mempunyai suatu file Aplikasi yang berbasis Windows.