

BAHASA PEMROGRAMAN INFORMIX

Informix-4GL adalah bahasa pemrograman generasi keempat yg memberikan kombinasi dua keunggulan yaitu **prosedural & non prosedural**, sehingga mampu memberikan semua fasilitas yg diperlukan dalam membentuk RDBMS.

Kemampuan informix meliputi :

- Bahasa pemrograman
- Screen-building utility
- Menu-building utility
- Report writer
- Window manager

Informix-4GL didisain secara khusus untuk penulisan program dalam membuat **database relasional** dan memberi fasilitas untuk memanipulasi data yg disimpan dalam database.

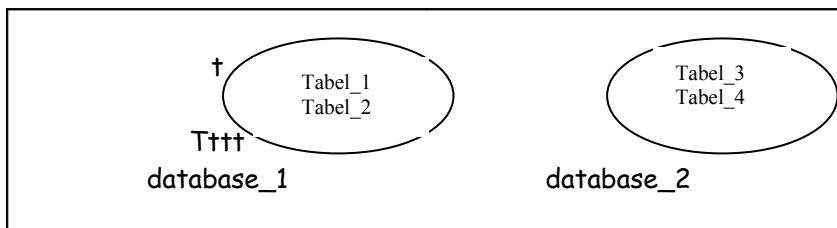
Penyimpanan Data

Informix menyimpan data-data didalam suatu **tabel**.

Tabel adalah kumpulan dari baris & kolom.

Tabel-tabel tsb disimpan dalam suatu **database**.

Suatu database minimal terdiri dari satu tabel.



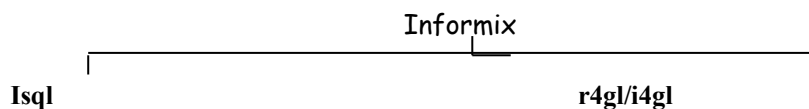
Tabel :

	NPM	Nama	Kota
record/baris 1 →	10197521	Viona	Jakarta
record/baris 2 →	10197214	Irfan	Bogor
	10197687	Benita	Jakarta

← field/attribute

→ data value

↑ ↑ ↑
 Field/kolom 1 kolom2 kolom 3



Informix-SQL merupakan kumpulan lengkap dari manajemen database yg memungkinkan kita untuk melaksanakan tugas dengan jangkauan yg luas.

Dengan informix-sql, kita dapat :

- menyusun & memodifikasi tabel
- menggunakan menu yg disediakan oleh schema editor
- memasukkan & mencari informasi database dgn menggunakan screen form
- mengurutkan, menggabungkan, mengatur & menampilkan data dengan report.
- Menambahkan, mengubah & mencari informasi database menggunakan query language
- Memanfaatkan isql melalui fasilitas khusus, user menu

i4gl (informix-4GL) → C compiler
r4gl (rapid development system)

r4gl memberikan fasilitas koreksi (debugger), program di-compile lebih cepat dari i4gl.
Sedangkan standart i4gl memerlukan compiler C.
r4GL tidak memerlukan compiler C dalam membuat program eksekusi

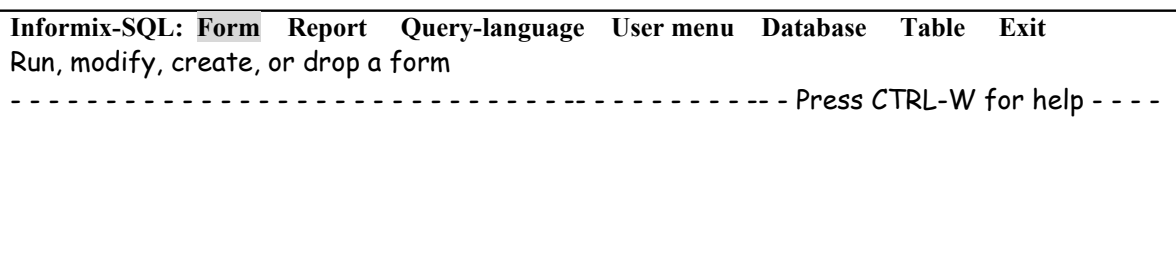
MEMULAI ISQL :

Ketik "isql" pada prompt unix



layar monitor

Kemudian akan muncul menu utama dari isql, sbb.:

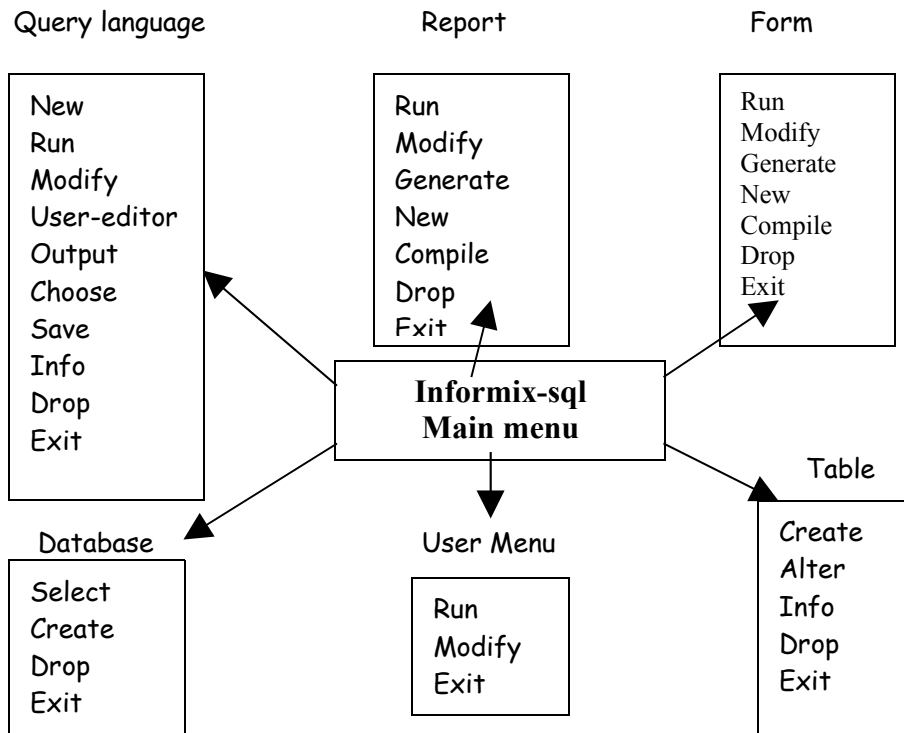


Menu Utama dari isql

Menu isql terdiri dari 6 menu utama seperti diatas.

Menu dapat **dipilih** dengan **menyorot** menu yg diinginkan atau dengan **mengetik huruf pertama** dari menu tsb.

Baris kedua menampilkan **pesan** dari menu yg dipilih / yg disorot.



Gambar. Menu Hierarki ISQL

MENU DATABASE - ISQL

Ada 2 cara untuk membuat database :

- melalui menu utama dengan memilih 'database'
- menggunakan 'sql' dari menu 'query language'

langkah-langkah membuat database, sbb. :

```
Informix-SQL: Form Report Query-language User menu Database Table Exit
Select , create, or drop a database
----- Press CTRL-W for help -----
```

```
DATABASE: Select Create Drop Exit
Select a database to work with.
----- Press CTRL-W for help -----
```

```
DATABASE : Select  Create  Drop  Exit
Create a new database
----- - Press CTRL-W for help -----
```

```
CREATE DATABASE >> filkom ↵
Enter the name you want to assign to the new database, then press Return
----- - Press CTRL-W for help -----
```

```
DATABASE : Select  Create  Drop  Exit
Select a database to work with.
----- filkom ----- - Press CTRL-W for help -----
```

- Select : untuk memilih database yang akan digunakan
- Create : untuk membuat database baru
 - Syarat pemberian nama database :
 - panjang 1-18 karakter
 - dimulai dengan huruf
 - tidak boleh ada spasi
 - tidak boleh menggunakan karakter khusus kecuali '_'
- Drop : untuk menghapus suatu database
- Exit : kembali ke menu utama

MENU TABLE DALAM ISQL

```
Informix-SQL: Form Report Query-language User menu Database Table Exit
Create, Alter, or Drop a database table
----- - Press CTRL-W for help -----
```

Untuk masuk ke menu 'table' tekan tombol 'T' , kemudian monitor akan menampilkan menu table sbb.:

```

TABLE : Create Alter Info Drop Exit
Create a new table
----- filkom ----- Press CTRL-W for help -----

```

- Create : untuk membuat tabel baru
- Alter : untuk mengubah tabel yg telah dibuat
- Info : untuk mengetahui struktur suatu tabel
- Drop : untuk menghapus tabel beserta datanya
- Exit : keluar dari menu tabel & kembali ke menu utama

Jika anda memilih 'create' dalam menu table , maka tampilan berikutnya seperti dibawah ini

```

CREATE TABLE >> t_master ↵
Enter the table nama you wish to create with the schema editor
----- filkom ----- Press CTRL-W for help -----

```

```

CREATE TABLE t_master : Add Modify Drop Screen Exit
Adds columns to the table above the line with the highlight
----- page 1 of 1 ----- filkom ----- Press CTRL-W for help -----

Column name      Type      Length      Index      Nulls


```

```

ADD NAME >> _
Enter column name. Return adds it. INTERRUPT return to CREATE / ALTER menu
----- page 1 of 1 ----- filkom ----- Press CTRL-W for help -----

Column name      Type      Length      Index      Nulls


```

```

ADD TYPE t_master >> Char Number Serial Date Money date-Time Interval
Permits any combination of letters, numbers, and symbols
----- page 1 of 1 ----- filkom ----- Press CTRL-W for help -----

Column name      Type      Length      Index      Nulls
npm              

```

ADD LENGTH >>_
 Enter column length. RETURN adds it
 ----- page 1 of 1 ----- filkom ----- Press CTRL-W for help -----

Column name	Type	Length	Index	Nulls
npm	char	8		

ADD INDEX t_master : Yes No
 Specifies that this columns will NOT have an index
 ----- page 1 of 1 ----- filkom ----- Press CTRL-W for help -----

Column name	Type	Length	Index	Nulls
npm	char	8		Yes

ADD NULLS t_master : Yes No
 Permits null values in this column
 ----- page 1 of 1 ----- filkom ----- Press CTRL-W for help -----

Column name	Type	Length	Index	Nulls
npm	char	8	Yes	N

CREATE TABLE t_master : Add Modify Drop Screen Exit
 Leaves the CREATE TABLE Menu
 ----- page 1 of 1 ----- filkom ----- Press CTRL-W for help -----

Column name	Type	Length	Index	Nulls
npm	char	8	Yes	No
nama	char	25		Yes
alamat	char	40		yes

EXIT t_master : Build-new-table Discard-new-table
 Builds a new table and returns to the Table Menu
 ----- page 1 of 1 ----- filkom ----- Press CTRL-W for help -----

Column name	Type	Length	Index	Nulls
npm	char	8	Yes	No
nama	char	25		Yes
alamat	char	40		yes

Build-new-table : membentuk tabel
Discard-new-table : membatalkan pembentukan tabel

Menu 'ALTER TABLE'

```
ALTER TABLE >> _  
Enter the table name you wish to alter with the schema editor  
----- filkom ----- Press CTRL-W for help -----  
  
t_biaya  
t_coba  
t_master
```

```
ALTER TABLE t_master : Add Modify Drop Screen Exit  
Adds column to the table above the line with the highlight  
-----page 1 of 1----- filkom ----- Press CTRL-W for help -----  
  
Column name      Type      Length      Index      Nulls  
npm              char      8           Yes        No  
nama             char      25          Yes        Yes  
alamat           char      40          Yes        yes
```

Menu 'INFO TABLE'

```
INFO FOR TABLE >> _  
Choose a table with the Arrow Keys, or enter a name, then press Return  
----- filkom ----- Press CTRL-W for help -----  
  
t_biaya  
t_coba  
t_master
```

```
INFO t_master : Columns Indexes Privileges Status Table Exit  
Display columns, names and data types for a table  
----- filkom ----- Press CTRL-W for help -----  
  
Column name      Type      Nulls  
npm              char(8)   No  
nama             char(25)  Yes  
alamat           char(40)  yes
```

MENU QUERY-LANGUAGE dalam ISQL

SQL adalah bahasa query interaktif, yg merupakan sistem dasar komputer untuk pemeliharaan record.

Sebagai sebuah sistem manajemen database SQL berisi program yg handal untuk mengerjakan tugas-tugas manajemen data, yg secara nyata mengurangi jumlah waktu untuk mengorganisasi, menyimpan dan mengakses informasi.

```
Informix-SQL: Form Report Query-language User menu Database Table Exit
Use Informix Structured Language
```

```
----- Press CTRL-W for help -----
```

```
SQL : New Run Modify Use-editor Output Choose save Info Drop Exit
Enter new SQL statement using the SQL editor
```

```
----- -filkom ----- Press CTRL-W for help -----
```

```
NEW : ESC = Done editing CTRL-A = Type over/insert
```

```
CTRL-x = Delete character CTRL-D = Delete rest of line
```

```
----- -filkom ----- Press CTRL-W for help -----
```

```
MODIFY: ESC = Done editing CTRL-A = Type over/insert
```

```
CTRL-x = Delete character CTRL-D = Delete rest of line
```

```
----- -filkom ----- Press CTRL-W for help -----
```

```
OUTPUT : Printer New-file Append-file To-pipe Exit
```

```
Send query results to a printer
```

```
----- -filkom ----- Press CTRL-W for help -----
```

```
CHOOSE >>
```

```
Choose a command file with the Arrow Keys, or enter a name, then press Return
```

```
----- -filkom ----- Press CTRL-W for help -----
```

```
Cek_npm          load_data
```

```
Ins_biodata      unload_dt
```



```

SAVE >>_
Enter the name you want to assign to the command file
-----fildkom----- Press CTRL-W for help -----

```

```

INFO FOR TABLE >>
Choose a table with the Arrow Keys, or enter a name, then press Return
-----fildkom----- Press CTRL-W for help -----
informix.t_biaya
informix.t_coba
infromix.t_master

```

```

DROP COMMAND FILE>>_
Enter the nama of the sql command file you wish to drop from the database
-----fildkom----- Press CTRL-W for help -----
Cek_npm          load_data
Ins_biodata      unload_dt

```

Contoh statement sql

```

NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----fildkom----- Press CTRL-W for help -----

Select npm,nama,telpon,kelas,ipk from t_master where ipk>3.00
Order by kelas,ipk

```

Setelah mengetik statement tsb, tekan tpmbol 'Esc' maka akan tampil menu SQL. Jika ingin menampilkan output dari statement tsb, maka pilih 'Run' pada menu sql, dan hasilnya akan tampak sbb :

```

RUN:  Next  Restart  Exit
Display the next page of query results
-----fildkom----- Press CTRL-W for help -----
  Npm      nama      telpon      kelas      ipk
10197348   Evita Adjani  8727541     4KA02     3,52
10197521   Sheila       4212817     4KA02     3,47
10197634   Indra         7342882     4KA03     3,61
10197867   Hendrawan    8727538     4KA06     3,25
10197491   Febrian      7321455     4KA07     3,28

```

Ada 5 tipe statetement SQL yang digunakan di Informix-SQL

1. DATA DEFINITION :

- | | | |
|-------------------|------------------|---------------------|
| α Create database | α Rename table | α Rename column |
| α Database | α Drop table | α Create index |
| α Close database | α Create view | α Alter index |
| α Drop database | α Drop view | α Drop index |
| α Create table | α Create synonym | α Update statistics |
| α Alter table | α Drop synonym | |

2. DATA MANIPULATION :

- | | |
|---------|---------|
| /Delete | /Update |
| /Insert | /Select |

3. DATA ACCESS :

- | | | |
|----------|----------------|-----------------|
| φ Grant | φ Lock table | φ Set lock mode |
| φ Revoke | φ unlock table | |

4. DATA INTEGRITY :

- β Recover table
- β Start database

5. AUXILIARY :

- | | | |
|----------------|----------|---------------|
| f Check table | f Load | f Output |
| f Repair table | f Unload | f Set explain |
| f Info | | |

Contoh-contoh statement query language

```

NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----filkom----- Press CTRL-W for help -----

SELECT npm,nama,telpon,kelas,ipk FROM t_master WHERE ipk>3.00 ORDER BY kelas,ipk
DESC
    
```

```

NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----filkom----- Press CTRL-W for help -----

SELECT * FROM t_pelanggan
    
```

```
NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----filkom----- Press CTRL-W for help -----
SELECT no_pel,nama,telpon FROM t_pelanggan WHERE no_pel[1]='B'
```

```
NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----filkom----- Press CTRL-W for help -----

SELECT npm,nama,kelas
FROM t_master
WHERE npm[1]='1' and npm[4,5]='97'
ORDER BY kelas,nama
```

```
NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----filkom----- Press CTRL-W for help -----

SELECT npm,nama FROM t_master WHERE npm[1] matches'[1234]'
```

```
NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----filkom----- Press CTRL-W for help -----

Select npm,nama from t_master where npm[1]='1' or npm[1]='2' or npm[1]='3'
or npm[1]='4'
```

```
NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----filkom----- Press CTRL-W for help -----

SELECT npm,nama,kelas FROM t_master WHERE nama matches'*ari*'
```

```
NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----filkom----- Press CTRL-W for help -----

SELECT * FROM t_master WHERE kelas[1] BETWEEN '1' and '3'
```

```
NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----filkom----- Press CTRL-W for help -----
Select * from t_master where kelas[1]='1' or kelas[1]='2' or kelas[1]='3'
```

```
NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----filkom----- Press CTRL-W for help -----
SELECT * FROM t_master WHERE kelas BETWEEN '4KA05' and '4KA06'
```

```
NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----filkom----- Press CTRL-W for help -----
SELECT UNIQUE kelas FROM t_master WHERE kelas[1]='4' ;
SELECT COUNT(*) FROM t_master WHERE kelas='4KA06' ;
SELECT npm,nama,MONTH(tgl_lahir) FROM t_master
```

```
NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----filkom----- Press CTRL-W for help -----
SELECT npm,jml_bayar,tgl_bayar,ket_bayar from t_uang_klh
WHERE npm IN (SELECT npm from t_master WHERE kelas[1,3]='4KA') ;
SELECT t_master.npm,nama,kelas,jml_bayar,tgl_bayar,ket_bayar
FROM t_master,t_uang_klh
WHERE t_master.npm=t_uang_klh.npm ;
SELECT no_pel,nama,alamat
INTO nomor,nama_pel,almt_pel
FROM t_pelanggan ;
```

```
NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----filkom----- Press CTRL-W for help -----
INSERT INTO t_pelanggan
VALUES ('B1504','Bambang','Merpati III') ;
```

```

INSERT INTO t_pelanggan (no_pel,alamat)
VALUES ('C2503','Camar II') ;

INSERT INTO t_master(npm,nama,kelas)
VALUES('10197348','Evita','4KA02') ;

SELECT np,nm_mhs,cls_mhs FROM t_percobaan
INSERT INTO t_master(npm,nama,kelas)

```

```

NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----filkom----- Press CTRL-W for help -----

UPDATE t_pelanggan SET keterangan = null ;

UPDATE t_pelanggan SET kode_pos ='10510'
WHERE alamat = 'Jakarta Pusat' ;

UPDATE t_uang_klh SET ket_bayar='L'
WHERE npm IN (SELECT npm FROM t_master where periode='1') ;

UPDATE t_uang_klh SET ket_bayar='L'
WHERE npm IN
(SELECT npm FROM t_master WHERE periode matches'[123]') ;

UPDATE t_pelanggan SET alamat = 'Akses Kelapa Dua'
WHERE alamat = 'Akses UI'

```

```

NEW :   ESC   = Done editing      CTRL-A = Type over/insert
        CTRL-x = Delete character  CTRL-D = Delete rest of line
-----filkom----- Press CTRL-W for help -----

DELETE FROM t_pelanggan ;
DELETE FROM t_pelanggan WHERE no_pel = 'A1054' ;
DELETE FROM t_master WHERE ipk < 0,50 AND kelas[1]='5' ;
DELETE FROM t_master WHERE ket_lulus='K' ;

CREATE DATABASE filkom ;

CREATE TABLE t_master
( npm          char(8),
  nama         char(25),
  alamat       char(40),
  kelas        char(5) ) ;

ALTER TABLE t_barang ADD (nama_brg char(20)) ;

```

```
RENAME COLUMN t_barang nama_brg TO nama_barang ;
```

```
RENAME TABLE t_master TO t_biodata
```

```
NEW :   ESC   = Done editing      CTRL-A = Type over/insert
```

```
        CTRL-x = Delete character  CTRL-D = Delete rest of line
```

```
-----fildom----- Press CTRL-W for help -----
```

```
--delete from t_pelanggan where telpon is null
```

```
select * from t_pelanggan where telpon is null ;
```

```
select no_pel,nama_pel,kota from t_pelanggan
```

```
where kota matches '[A-J]*'
```

MENU FORM dalam ISQL

Screen form adalah media yg digunakan untuk input/output variabel pada program 4GL, dan juga digunakan sebagai penghubung data secara interaktif untuk menghapus, melihat, modifikasi atau menghapus data.

Pembuatan form ada 2 cara :

- *Generate* : sistem informix yg membuat form
- *New* : programmer yg membuat sendiri

```
Informix-SQL: Form Report Query-language User menu Database Table Exit
Run, modify, create, or drop a form
----- Press CTRL-W for help -----
```

```
FORM : Run Modify Generate New Compile Drop Exit
Use a form to enter data or query a database
----- filkom----- Press CTRL-W for help -----
```

Run : untuk menjalankan form yg telah dicompile / generate
Modify : mengubah bentuk form yg telah dibuat / digenerate
Generate : membuat form baru dgn bantuan sistem informix
New : membuat form baru tanpa bantuan sistem informix
Compile: mengcompile form yg telah dibuat/dimodifikasi sehingga dapat dijalankan
Drop : untuk menghapus form

Langkah-langkah pembuatan form dengan menggunakan 'Generate'

```
GENERATE FORM >> f_master
Enter the name you want to assign to then form, then press Return
----- filkom----- Press CTRL-W for help -----
```

```
CHOOSE TABLE >>
Choose the table to be used in the default form
----- filkom----- Press CTRL-W for help -----

informix.t_biaya
informix.t_coba
informix.t_master
```

```
GENERATE FORM : Table-selection-compile  Select-more-tables  Exit
Continue creating a default form with the selected tables
----- filkom----- Press CTRL-W for help -----
```

Setelah memilih 'Table-select-compile', kemudian akan kembali menu Form, ini berarti sudah tercipta sebuah form dengan nama 'f_master', jika ingin melihat tampilan dari form tsb, maka pilih 'Run' pd menu Form.

Langkah-langkah pembuatan form dengan menggunakan 'New'

```
NEW FORM>> f_master2 ↵
Enter the name you want to assign to the form, then press Return
----- filkom----- Press CTRL-W for help -----
```

```
USE-EDITOR >> vi ↵
Enter editor name, (RETURN only for default editor)
----- filkom----- Press CTRL-W for help -----
```

```
~ _
~
~      'buat sendiri program yang anda inginkan'
~      harus mengetahui penggunaan editor 'vi'
~
~
"tmp/f_master2.per [new file]
```

Setelah program selesai dibuat, maka tampilan selanjutnya sbb.:

```
NEW FORM : Compile  Save-and-exit  Discard-and-exit
Compile the form specification.
----- filkom----- Press CTRL-W for help -----
```

Form yang telah dibuat dengan '**Generate**' atau '**New**', dapat ditampilkan dengan memilih '**Run**' pada menu **Form**.

```
RUN FORM>> _
```


Choose a form with Arrow Keys, or enter a name, then press Return
----- filkom----- Press CTRL-W for help -----

f_biaya
f_coba
f_master

PERFORM : Query Next Previous View Add Update Remove Table ...
Searches the active database table

1 : t_master table

Npm []
Nama []
Alamat []
Kota []
Telpon []
Kelas []
Ipk []

Jika anda ingin mencari data atau mengubah data maka pilihlah 'query'

QUERY : ESCAPE queries. INTERRUPT discards query. ARROW keys move cursor.
Searches the active database table

1 : t_master table

Npm []
Nama []
Alamat []
Kota [Jakarta]
Telpon []
Kelas []
Ipk []

Contoh : mencari data mahasiswa yg alamatnya di 'Jakarta'

Setelah memasukkan kondisi seperti diatas, kemudian tekan tombol 'Esc'

Tampilan sama dengan atas, yg berbeda cuma pojok kiri bawah

Searching

Jika datanya ketemu, maka tampilan selanjutnya

PERFORM : Query Next Previous View Add Update Remove Table ...
Searches the active database table

1 : t_master table

Npm [10197348]
Nama [Yudisthira]
Alamat [Salemba No.33]
Kota [Jakarta]
Telpon [4212817]
Kelas [4KA01]
Ipk [2,65]

15 row(s) found

Beberapa contoh mencari data dengan menggunakan 'query' pada menu 'Run-Form (PERFORM)'

←

QUERY : ESCAPE queries. INTERRUPT discards query. ARROW keys move cursor.
Searches the active database table

1 : t_master table

Npm [???97*]
Nama []
Alamat []
Kota []
Telpon []
Kelas []
Ipk [>3,00]

↑

QUERY : ESCAPE queries. INTERRUPT discards query. ARROW keys move cursor.
Searches the active database table

1 : t_master table

Npm [1?196*]
Nama [*ika* Nasution]
Alamat []
Kota []
Telpon []
Kelas []
Ipk []

QUERY : ESCAPE queries. INTERRUPT discards query. ARROW keys move cursor.
Searches the active database table

1 : t_master table

Npm []
Nama []
Alamat []
Kota [Depok]
Telpon []
Kelas [4KA06]
Ipk []

↓QUERY : ESCAPE queries. INTERRUPT discards query. ARROW keys move cursor.
Searches the active database table

1 : t_master table

Npm []
Nama []
Alamat []
Kota []
Telpon [8727541]
Kelas []
Ipk []

Tampilan penggunaan 'Add' pada menu 'Run-Form (PERFORM)'

ADD : ESCAPE adds new data. INTERRUPT discards it. ARROW keys move cursor
Adds new data to the active database table

1 : t_master table

Npm [_]
Nama []
Alamat []

Masukkan data , kemudian tekan tombol 'Esc'

PERFORM : Query Next Previous View **Add** Update Remove Table
Adds new data to the active database table

1 : t_master table

Npm [10197521]
Nama [Evita]
Alamat [Akses UI No.53]

Row added

Tampilan penggunaan 'Update' pada menu 'Run-Form (PERFORM)'

UPDATE : ESCAPE changes data. INTERRUPT discard changes
Changes this row in the active database table.

1 : t_master table

Npm [10197521]
Nama [Evita]
Alamat [Akses UI No.53]

Ubahlah data diatas sesuai yg diinginkan, kemudian tekan tombol 'Esc'

PERFORM : Query Next Previous View Add Update Remove Table
Changes data to the active database table

1 : t_master table

Npm [10197521]
Nama [Evita Adjani]
Alamat [Akses UI No.53]

This row has been changed

Tampilan penggunaan 'Remove' pada menu 'Run-Form (PERFORM)'

REMOVE : Yes No
Removes this row from the active table

1 : t_master table

Npm [10197521]

PERFORM : Query Next Previous View Add Update Remove Table
Delete a row from the active database table

1 : t_master table

Npm []

Row deleted

Tampilan penggunaan 'Compile' pada menu 'Form'

FORM : Run Modify Generate New Compile Drop Exit
Compile the form specification

----- filkom----- Press CTRL-W for help -----

```
COMPILE FORM >> _
Choose a form with the ARROW Keys, or enter a name, then press Return
----- filkom----- Press CTRL-W for help -----

f_biaya
f_coba
f_master
```

Tampilan penggunaan 'Drop' pada menu 'Form'

```
FORM : Run  Modify  Generate  New  Compile  Drop  Exit
Drop a form
----- filkom----- Press CTRL-W for help -----
```

```
DROP FORM >> _
Choose a form with the ARROW Keys, or enter a name, then press Return
----- filkom----- Press CTRL-W for help -----

f_biaya
f_coba
f_master
```

```
CONFIRM : No  Yes
No, I do not want to drop it
----- filkom----- Press CTRL-W for help -----

f_biaya
f_coba
f_master
```

MODIFY FORM

```
FORM : Run  Modify  Generate  New  Compile  Drop  Exit
Modify a form
----- filkom----- Press CTRL-W for help -----

f_biaya
f_coba
f_master
```

MODIFY FORM >>_

Choose a form with the ARROW Keys, or enter a name, then press Return

----- filkom----- Press CTRL-W for help -----

f_biaya
f_coba
f_master

Database filkom

Screen size 24 by 80

```
{  
npm      [f010 ]  
nama     [f020      ]  
alamat  [f030      ]  
}
```

end

tables

t_master

attributes

f010 = t_master.npm;

f020 = t_master.nama;

f030 = t_master.alamat;

end

tekan tombol 'Esc' kemudian ketik ':wq'

Tampilan program diatas
seperti tampilan sebelumnya

```
Npm      [      ]  
Nama     [      ]  
Alamat  [      ]
```

Ubahlah tampilan disamping tsb. menjadi seperti
dibawah ini.

BIODATA MAHASISWA

```
Nomor Pokok Mhs. = [      ]  
Nama Mahasiswa  = [      ]  
Alamat Mahasiswa = [      ]
```

Untuk membuat tampilan seperti yang diinginkan, maka program awal harus diubah sbb.:

```
Database filkom
Screen size 24 by 80
{
  \gp-----q\g
  \g\g      BIODATA MAHASISWA      \g\g
  \g\g      \g-----\g            \g\g
  \g\g      \g\g                    \g\g
  \g\g      \g\g                    \g\g
  \g\g      Nomor Pokok Mhs. = [f010  ]      \g\g
  \g\g      Nama Mahasiswa = [f020          ]      \g\g
  \g\g      Alamat Mahasiswa = [f030          ] \g\g
  \g\g      \g\g                    \g\g
  \g\g      \g\g                    \g\g
  \gb-----d\g
}
end
tables
t_master
attributes
f010 = t_master.npm,required;
f020 = t_master.nama,autonext;
f030 = t_master.alamat;
end

tekan tombol 'Esc' kemudian ketik ':wq'
```

FORMAT UMUM MODIFIKASI FORM

```
DATABASE section
SCREEN section
{
  [text] [field_tag] [char_graphics]
  .....
}
END
TABLES table names
ATTRIBUTES
  field_tag = table.column [,type];
  .....
```

END

DATABASE section :

Mengidentifikasi database yg digunakan

Jika tidak memakai database, maka ditulis 'DATABASE FORM ONLY'

SCREEN section :

Untuk menggambarkan bagaimana form akan ditampilkan di monitor.

Default adalah 24 baris, 80 kolom.

[text] :

kata / kalimat yang akan ditampilkan di monitor

[field_tag] :

untuk menunjukkan bagaimana data akan ditampilkan di monitor,

field yang akan ditampilkan menggunakan tanda [], masing-masing field

b'hubungan dgn [field_tag] yg diidentifikasi di ATTRIBUTES section.

[char_graphics] :

yang dapat digunakan antara lain :

<u>char</u>		<u>tampilan di layar</u>
p	↑	┌
q	↑	┐
b	↑	└
d	↑	┘
	↑	
-	↑	—

Supaya tiap karakter bersambung,
penulisannya diapit '\g'

TABLES :

Digunakan untuk mengidentifikasi seluruh nama tabel yg dipakai dalam spesifikasi form.

ATTRIBUTES :

Menggambarkan bagaimana bentuk dari masing-masing field yang akan ditampilkan di monitor.

TYPE :

Argumen yang dapat digunakan pada tiap field, antara lain :

autonext	Kursor otomatis pindah ke field berikutnya jika karakter untuk field tsb sudah penuh
downshift	Semua karakter dikonversi ke huruf kecil
upshift	Semua karakter dikonversi ke huruf besar
comments	Perform akan menampilkan pesan dibawah layar ketika kursor pindah ke field yg diberi argumen comments. Contoh : FOO1 = t_master.alamat,comments='Alamat harus lengkap';

include	Menentukan nilai-nilai yg diperbolehkan masuk dalam field ini. Contoh : F002 = t_master.agama,include=('1','2','3','4','5'); F003 = t_master.jns_kel,include=('1','2');
default	Memberikan nilai dasar bila field tidak diisi Contoh : F004 = t_master.ket,default='T';
noentry	Mencegah pemasukan data
picture	Menspesikasikan pola data masukan A → huruf , # → numerik , x → alfanumerik Contoh : F005 = t_krs.kd_mk,picture ='AA-#####'; F006 = t_biaya.jml,picture ='Rp ##.###.###,##';
required	Memaksa pemasukan nilai pada field tsb. Contoh : F007 = t_master.kode_pos,required;
reverse	Mengganti warna karakter kebalikan dengan warna background.
right	Agar data rata kanan
verify	Memaksa pemasukan nilai sebanyak dua kali
zerofill	Agar field rata kanan dan dimulai dengan angka 0, berguna untuk data bertipe numerik

MENU REPORT dalam ISQL

```
Informix-SQL: Form Report Query-language User menu Database Table Exit
Run , modify , generate, new , compile , drop
----- Press CTRL-W for help -----
```

```
REPORT : Run Modify Generate New Compile Drop Exit
Run a report
```

Run : Run a report
Modify : modify a report specification
Generate : generate a default report
New : create a new report specification
Compile : compile a report
Drop : drop a report from the database

Ada 2 cara membuat report, yaitu dengan menggunakan 'Generate' atau 'New'

Jika pilih 'Generate' ⇒ pada menu REPORT, ketik tombol 'G' (huruf pertama dari Generate)

maka tampilan di layar akan berubah seperti berikut ini :

```
Generate Report >> (ketik nama report)
Enter the name you want to assign to the report, then press Return
----- filkom ----- Press Ctrl W - for Help -----
```

Contoh : nama reportnya adalah 'r_coba'

```
Choose TABLE >> (ketik nama table atau pilih table-table dibawah)
Choose the table to be used in the default report
----- filkom ----- Press Ctrl W - for Help -----
```

```
informix.t_biaya
informix.t_master
informix.t_uang_klh
```

Contoh : nama table yang dipilih adalah 't_master'

Setelah anda memilih table, maka tampilan layar akan kembali ke menu REPORT, sbb.:

```
REPORT : Run Modify Generate New Compile Drop Exit
```

```
Run a report
```

```
----- filkom ----- Press Ctrl W - for Help -----
```

Pilih 'Run' untuk menampilkan hasilnya, kemudian layar akan berubah sbb.:

```
Run REPORT >> (ketik nama report atau pilih report2 dibawah)
```

```
Choose a report with Arrow Keys, or enter a name
```

```
----- filkom ----- Press Ctrl W - for Help -----
```

```
r_awal
```

```
r_coba
```

```
lap_akhir
```

Akan tampil output sbb.:

```
Run REPORT >> r_coba
```

```
Choose a report with Arrow Keys, or enter a name
```

```
----- filkom ----- Press Ctrl W - for Help -----
```

```
The reading of the database will now .....
```

```
.....
```

npm	nama	alamat
10197234	Niken Anwar	Margonda Raya
10197575	Irwan	Salemba
10197871	Vera Anita	Akses UI

Output tersebut diatas dibuat oleh informix, jika anda ingin melihat programnya atau ingin mengubah tampilan outputnya, maka pada menu REPORT, pilihlah 'Modify'

```
MODIFY REPORT >> _
```

```
Choose a report with the Arrow Keys, or enter a name, then press Return
```

```
----- filkom ----- Press Ctrl W - for Help -----
```

```
r_awal
```

```
r_coba
```

```
lap_akhir
```

Setelah memilih nama report, muncul editor 'vi' yang menampilkan program sbb.:

```

Database filkom end
Select npm,
      nama,
      alamat
From informix.t_master end
Format every row end

```

Program diatas dapat dimodifikasi (menggunakan editor 'vi', pelajari tentang editor 'vi') seperti dibawah ini :

Sintaks REPORT spect

```

DATABASE
  Nama database  mutlak ada
END

DEFINE
  VARIABLE var_name data_type  tidak mutlak
END

INPUT
  PROMPT FOR  tidak mutlak
END

OUTPUT
  REPORT TO ('file.name'| Printer)
  LEFT MARGIN
  RIGHT MARGIN
  TOP MARGIN
  BOTTOM MARGIN
  PAGE LENGTH
END  tidak mutlak

SELECT
  Statement query language  mutlak ada
END

FORMAT
  PAGE HEADER control block
  PAGE TRAILER control block
  FIRST PAGE HEADER control block
  ON EVERY ROW control block
  ON LAST ROW control block
  BEFORE GROUP OF control block
  AFTER GROUP OF control block
END  mutlak ada

```

PAGE HEADER : Pencetakan judul pada tiap halaman
PAGE TRAILER : Dicitak pada baris terakhir tiap halaman
FIRST PAGE HEADER : Pencetakan judul hanya pada halaman pertama saja
ON EVERY ROW : Mencetak setiap baris informasi yg dikirim menggunakan looping
ON LAST ROW : Dicitak setelah seluruh informasi selesai dicetak
BEFORE GROUP OF / AFTER GROUP : Dicitak sebelum / sesudah pengelompokkan

```

Database filkom end
Output Report to '/dev/lp9'
End
Select npm,nama,alamat from informix.t_master Order by npm
End
Format
  Page header
    Print column 20,"BIODATA MAHASISWA"
    Skip 2 lines
    Print '-----'
    Print '   N P M       N a m a           A l a m a t   '
    Print '-----'
  On every row
    Print column 2,npm,column 15,nama,column 35, alamat
    Print '-----'
End

```

Setelah selesai diubah programnya maka tampilan berikutnya adalah

```

MODIFY REPORT : Compile Save-and-exit Discard-and-exit
Compile the report specification
----- filkom ----- Press Ctrl W - for Help -----

```

Jika anda ingin melihat hasil yang telah di'compile' , maka pilih menu 'Run' pada Report, dan akan tampil outputnya sbb.:

```

Run REPORT >> r_coba
Choose a report with Arrow Keys, or enter a name
----- filkom ----- Press Ctrl W - for Help -----

The reading of the database will now .....
.....
.....

BIODATA MAHASISWA

```

N P M	N a m a	Alamat
10197234	Niken Anwar	Margonda Raya
10197575	Zulliandra	Salemba
10197871	Vera Anita	Akses UI

Database filkom end

Select kd_mk,nm_mk,sks,jenis **from** t_krs

Order by jenis,nm_mk

End

Format

Page header

Print column 40,'MATA KULIAH'

Print column 30,'FAKULTAS KOMPUTER'

Print column 30,'-----'

Skip 2 lines

Print column 70,'Halaman : ',pageno using '###'

Print '-----'

Print 'Kode MK Nama Mata Kuliah Jenis SKS'

Print '-----'

Skip 1 line

On every row

Print kd_mk,column 15,nm_mk,column 55,jenis,column 65,sks

After group of jenis

Skip 2 lines

On last row

Print '-----'

Print column 45,'Total Matakuliah',

column 60,count using '###'

Page trailer

Print 'Laporan Mata Kuliah'

End

Output

MATA KULIAH FAKULTAS KOMPUTER -----

Halaman : 1

Kode MK	Nama Mata Kuliah	Jenis	SKS
KK-011317	Teori Bahasa & Otomata	W	3
KK-011314	Sistem Operasi	W	3
KK-011308	Analisis & Perancangan Sistem	UU	3
KK-011305	Pengembangan Sistem Informasi	UU	3
KK-011345	Pemrograman <i>Generasi Keempat</i>	P	3

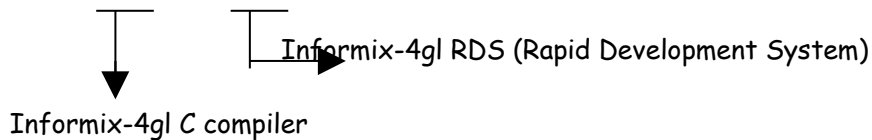
Total Mata Kuliah = 5

Laporan Mata Kuliah

PEMROGRAMAN INFORMIX

Memulai informix-4gl

Ketik i4gl atau r4gl pada shell prompt



/u/audry/komputer> **r4gl** ↵

maka akan tampil menu sbb :

```
INFORMIX-4GL : Module Form Program Query-language Exit
Create , modify , or run individual 4gl program modules
----- Press CTRL-W for help -----
```

Module : bekerja pad program module Informix-4gl
Form : bekerja pada screen form
Program : spesifikasi komponen dari program multi-module
Query language : menggunakan SQL interaktif
Exit : kembali ke shell prompt

Menu MODIFY Dalam Module 4GL

```
Module : Modify New Compile Program_Compile Run Debug Exit
Change an existing 4gl program module
----- Press CTRL-W for help -----
```


- Modify** : untuk memodifikasi module 4gl yang sudah ada
- New** : untuk membuat module 4gl yang baru
- Compile**: untuk mengkompile module 4gl yang stand_alone tanpa melalui pilihan modify.
- Program_compile** : mengijinkan kita mengkompile dan menggabung module.
Pilihan ini berguna jika kita mempunyai satu module ari sekelompok program yang kompleks & ingin menguji dengan cara mengkompilanya dengan module lain.
- Run** : menjalankan module program
- Debug** : menggunakan Informix-4gl interaktif debugger untuk menganalisa program.

```

MODIFY MODULE >>
Choose a module with Arrow Keys or enter a name, and press RETURN
----- Press CTRL-W for help -----
Prog_blanko
Prog_dns
Prog_krs

```

```

USE-EDITOR >> vi
Enter editor name. (Return only for default editor)
----- Press CTRL-W for help -----

```

Kemudian muncul program yang sudah pernah dibuat,sbb.:

```

Main
  Display "saat ini kami sedang belajar Informix-4gl" at 5,10
  Display "mohon jangan diganggu" at 6,15
End main

:wq

```

```

MODIFY MODULE : Compile Save-and-exit Discard-and-exit
Compile the 4gl module specification
----- Press CTRL-W for help -----

```

```

COMPILE MODULE : Object Runnable Exit
Create an object file (.4go suffix)
----- Press CTRL-W for help -----

```



```
COMPILE MODULE : Object Runnable Exit
Create an runnable program (.4gi suffix)
----- Press CTRL-W for help -----
```

Setelah sukses mengkompilasi, tampilan berikutnya sbb.:

```
MODIFY MODULE : Compile Save-and-exit Discard-and-exit
Save the 4GL module and return to MODULE Menu
----- Press CTRL-W for help -----
```

Menu NEW Dalam Module 4GL

```
Module ; Modify New Compile Program_Compile Run Debug Exit
Create a new 4GL program module
----- Press CTRL-W for help -----
```

```
NEW MODULE >>_
Enter the name you want to assign to the module, and press RETURN
----- Press CTRL-W for help -----
```

```
~
~      ketik program anda
~
~
```

Dalam setiap modul program 4GL harus terdapat **main program**.
Catatan Kuliah : Pemrograman Bahasa Generasi Ke-4

Dalam main program harus ada statement **MAIN**

```
Main
.....
....statement.....
.....
End main
```

MAIN → menyatakan awal blok program .

END MAIN → menyatakan akhir blok program.

Contoh :

```
Main
CREATE DATABASE filkom
CREATE TABLE t_matakuliah
(kd_mk      char(8),
 nm_mk      char(30),
 sks        smallint,
 jenis      char(1))
INSERT INTO t_matakuliah
VALUES ("DU-14201","Agama",2,"W")
```

End main

```
Main

display 'Universitas Gunadarma' at 6,7
display 'Jl. Akses UI Kelapa Dua' at 7,7
attribute (reverse,blink)

end main
```

Output :

```
Main
Define
  nm  char(25),
  ket char(30)

Let ket = 'mahasiswa Universitas Gunadarma'
Prompt 'Nama anda  :' for nm
  Display 'Mhs. yang bernama ',nm,ket at 10,5
  Display 'Mhs. yang bernama ',nm,' adalah',ket at 12,5
End main
```

Output :

```
Main
Define nm char(25),
```

```
    kls char(5)

    prompt 'Masukkan nama anda : ' for nm
    prompt 'Masukkan kelas anda : ' for kls

    display 'Saya yang bernama : ',nm
    display 'Sekarang duduk di kelas ',kls
    display 'Sedang belajar Informix'

End main
```

Vi Editor

Vi editor (visual editor) adalah editor klasik pada 'unix' yg dikembangkan di California (Barkeley) yg diciptakan oleh Bill Joy.

Vi editor dikatakan editor layar karena text yg disunting ditampilkan pada layar.

Modus 'vi' :

- modus masukan / penyisipan
 untuk melakukan pengetikan teks
- modus perintah
 untuk melakukan perintah yg diinginkan, agar dapat berpindah dari modus masukan ke modus perintah atau sebaliknya, tekan tombol (Esc).
- modus baris akhir
 berkaitan dengan kemampuan 'vi' yg mendukung operasi berbasis baris yg bersifat kompatibel.
 Untuk berpindah ke modus ini, didahului dengan menekan tombol (Esc) : ,
 (mis. Untuk simpan teks ⇒ : wq)

Memulai vi :

- tentukan nama file
- berikan perintah

contoh : /u/informix> vi latihan1.txt ↵

Keluar dari vi :

- : q ↵ : keluar dari vi & tidak menyimpan hasil perubahan
- : wq ↵ : menyimpan ke file dan keluar dari vi
- : x ↵ : menyimpan ke file dan keluar dari

Menambahkan teks :

- i text (Esc) : menyisipkan bbrp karakter di sebelah kiri posisi kursor
- I text (Esc) : menyisipkan bbrp karakter di awal baris
- a text (Esc) : menyisipkan karakter di sebelah kanan posisi kursor
- A text (Esc) : menyisipkan karakter di akhir baris
- o text (Esc) : menyisipkan 1 baris kosong di bawah posisi kursor

- O text (Esc) : menyisipkan 1 baris kosong di atas posisi kursor

Menggerakkan kursor :

<ctrl> f : gerakkan kursor 1 layar ke bawah
<ctrl> b : gerakkan kursor 1 layar ke atas
+ , <enter> : gerakkan kursor ke bawah 1 baris
- : gerakkan kursor ke atas 1 baris

Menghapus text :

x : menghapus karakter pada posisi kursor
X : menghapus karakter di kiri posisi kursor
dw : menghapus dari posisi kursor sampai akhir kata
db : menghapus dari awal kata hingga sebelum posisi kursor
dd : menghapus 1 baris dimana kursor berada
d0 (nol) : menghapus dari awal baris sampai sebelum posisi kursor
contoh :
3x : menghapus 3 karakter
5dd : menghapus 5 baris

Pengubahan text :

cw : mengubah sampai akhir kata
cb : mengubah dari awal kata sampai posisi sebelum kursor

Mencari string :

/ string: mencari string pertama yg cocok dgn pola setelah kursor
? string : mencari string pertama yg cocok dgn pola sebelum kursor
n : mencari string selanjutnya

Mencari dan mengganti string :

:s/hari/saat : mengganti kata 'hari' yg pertama ditemui menjadi 'saat'
:s/hari/saat/g : mengganti 'hari' yg berada pd satu brs menjadi 'saat'
:s/hari//saat/g: mengganti semua kata 'hari' menjadi kata 'saat'

\$ cat contoh ↵

```
hari itu hari minggu
hari yang cerah ceria
hari, andi dan rani tertawa riang
hari itulah hari ulang tahun hari
```

\$ vi contoh ↵

```
hari itu hari minggu
hari yang cerah ceria
hari, andi dan rani tertawa riang
hari itulah hari ulang tahun hari

<esc> :/hari/saat
```

→

```
saat itu hari minggu
hari yang cerah ceria
hari, andi dan rani tertawa riang
hari itulah hari ulang tahun hari
```

<esc> :wq

hari itu hari minggu
hari yang cerah ceria
hari, andi dan rani tertawa riang
hari itulah hari ulang tahun hari

<esc> :s/hari/saat/g

→

saat itu **saat** minggu
hari yang cerah ceria
hari, andi dan rani tertawa riang
hari itulah hari ulang tahun hari

hari itu hari minggu
hari yang cerah ceria
hari, andi dan rani tertawa riang
hari itulah hari ulang tahun hari

<esc> :s/hari//saat/g

→

saat itu saat minggu
saat yang cerah ceria
saat, andi dan rani tertawa riang
saat itulah saat ulang tahun saat

Program dengan menggunakan DATABASE

```
Database filkom          #contoh program cari data
Main
  Define nm  char(25), kls char(5)
  SELECT nama,kelas INTO nm,kls
  FROM t_master
  WHERE npm='10197477'

  Display 'Nama : ',nm at 5,9
  Display 'Kelas : ',kls at 6,9
End main
```

```
Database filkom
Main
Define nm like t_master.nama, kls like t_master.kelas,
  m_npm like t_master.npm

prompt 'Masukkan NPM yang dicari : ' for m_npm

select nama,kelas into nm,kls from t_master where npm=m_npm

display nm clipped,'/',m_npm,'/',kls at 10,12
end main
```

Contoh program INSERT

```
# statement yg diawali tanda '#' tidak diakses
# Ini adalah contoh untuk memasukkan data kedalam table
# Program ini dibuat oleh : Rizky
# Tanggal pembuatan : 24 /11/2000
#-----

Database filkom    # kalau pindah fakultas,ganti database
Main
  INSERT INTO t_matakuliah (kd_mk,nm_mk,sks,jenis)
  VALUES ('DK-14202','Fisika',2,'W')
```

```
End main
```

```
{ contoh kedua program insert data  
dibuat oleh .....  
selain menggunakan tanda '#' boleh juga menggunakan tanda{' }'  
}
```

```
Database filkom
```

```
Main
```

```
Define kode_mk char(8), nama_mk char(25),  
jsks smallint, jns char(1)
```

```
let kode_mk = 'DK-14202' #contoh penggunaan LET
```

```
let nama_mk = 'fisika'
```

```
let jsks = 2
```

```
let jns = 'W'
```

```
INSERT INTO t_matakuliah(kd_mk,nm_mk,sks,jenis)  
VALUES (kode_mk,nama_mk,jsks,jns)
```

```
End main
```

```
#contoh ketiga program insert data  
# bandingkan format penulisan define
```

```
database filkom
```

```
main
```

```
define kode_mk char(8),nama_mk char(25),jsks smallint,  
jns char(1)
```

```
prompt 'masukkan kode matakuliah = ' for kode_mk
```

```
prompt 'masukkan nama matakuliah = ' for nama_mk
```

```
prompt 'masukkan jumlah sks = ' for jsks
```

```
prompt 'masukkan jenis matakuliah = ' for jns
```

```
#dibawah ini adalah statement 'sql'
```

```
# penulisan statement sql-nya sama dengan contoh kedua
```

```
INSERT INTO t_matakuliah (kd_mk,nm_mk,sks,jenis)  
VALUES (kode_mk,nama_mk,jsks,jns)
```

```
end main
```

Contoh Program DELETE


```
Database ekonomi
Main
  Define np like t_master.npm
  Prompt 'masukkan npm anda : ' for np

  DELETE FROM t_master WHERE npm=np
End main
```

```
Database filkom
Main
  Delete from t_master where kelas[1]='4' and ket_lulus='W'
End main
```

Contoh Program UPDATE

```
Database sastra
Main
  Define
    np like t_master.npm,
    almt like t_master.alamat

  prompt 'masukkan npm anda   : ' for np
  prompt 'masukkan alamat baru : ' for almt

#perintah sql
  UPDATE t_master set alamat = almt WHERE npm = np

End main
```

```
Database sastra
Main

  UPDATE t_master set kota='Depok' WHERE kota='Cimanggis'

End main
```

Contoh penggunaan FOR END FOR

```
-- membuat program dengan menggunakan FOR ... END FOR
-- kegunaan tanda '--' sama dengan tanda '#'

Database filkom
```

```

main
  define np char(8),
  nama char(25), kls char(5),i smallint

FOR i = 1 to 5
  prompt 'ketik npm anda  = ' for np
  prompt 'ketik nama anda = ' for nama
  prompt 'ketik kelas anda = ' for kls

  INSERT INTO t_master (npm,nama_mhs,kelas)
    VALUES (np,nama,cls)

  Display 'N P M   : ',np at 6,5
  Display 'Nama   : ',nama at 7,5 attribute(blink)
  Display 'Kelas : ',cls at 8,5
  Sleep 2
  Clear screen

END FOR

display 'selesai dech .....by.....' At 15,10

end main

```

{contoh penggunaan **WHILEEND WHILE** }

```

database filkom
main
  define np char(8), nama char(25), kls char(5), jawab char(1)

let jawab='y'
WHILE jawab='y' or jawab='Y' #boleh ditulis jawab matches'[Yy]'
  Prompt 'ketik npm anda  = ' for np
  Prompt 'ketik nama anda = ' for nama
  Prompt 'ketik kelas anda = ' for kls
  insert into t_master (npm,nama_mhs,kelas) values (np,nama,cls)
  prompt 'ingin tambah data lagi [y/t] / ' for jawab
  clear screen
END WHILE
End main

```

RECORD

Record merupakan kumpulan dari program variable yang kemungkinannya berbeda tipe data yang satu dengan lainnya.

Sebelum memakai suatu record , kita harus mendefinisikan terlebih dahulu pada DEFINE.

Catatan Kuliah : Pemrograman Bahasa Generasi Ke-4

```

record_name

    RECORD
        Var_list data_type [,.....]
    END RECORD

```

```

Define
    rec_mk RECORD
        kd_mk char(8), nm_mk char(25),
        sks smallint, jenis char(1)
    END RECORD
Main
    Select kd_mk,nm_mk,sks,jenis INTO rec_mk.*
    from t_matakuliah where kd_mk = 'TI-17203'

#-----
#sql diatas bisa ditulis seperti dibawah ini
{ Select kd_mk,nm_mk,sks into rec_mk.kd_mk THRU rec_mk.sks
From t_matakuliah where kd_mk='TI-17203' }
#-----
    display 'Nama Matakuliah : ',rec_mk.nm_mk
    display 'Kode Matakuliah : ',rec_mk.kd_mk
end main

```

Jika kita mendefinisikan record sama seperti pada suatu table (seperti contoh diatas), maka penulisannya dapat disederhanakan menjadi :

```

record_name RECORD LIKE table_name.*

```

```

Define
    rec_mk RECORD LIKE t_mk.*
main
    .....
end main

```

PERINTAH SELECT DENGAN CURSOR

- Mendeklarasikan cursor untuk mewakili perintah **SELECT**
- Mendapatkan row dengan **FOREACH**

- Mendapatkan row dengan **OPEN** , **FETCH** dan **CLOSE**

Mendeklarasikan cursor untuk select

sintaks : **DECLARE** nama_cursor [scroll] **CURSOR FOR**
SELECT statement

```
DECLARE k1 CURSOR FOR
Select * from t_master
Where kelas[1,2] = '4K'
```

```
DECLARE kur CURSOR FOR
Select npm,nama from t_master
Where nama[1]='R'
```

Mendapatkan row dengan FOREACH

Statement 'foreach' menyebabkan urutan statement dikerjakan satu persatu.

sintaks : **FOREACH** nama_kursor [**INTO** var]
 statement.....

 [CONTINUE FOREACH]

 [EXIT FOREACH]

 END FOREACH

- Nama kursor di **FOREACH** harus sudah pernah didefinisikan pada statement **DECLARE**
- **INTO** digunakan untuk menampung hasil query setiap row
- **CONTINUE FOREACH** atau **EXIT FOREACH** digunakan untuk interupsi pada kondisi tertentu
- Diakhiri dengan **END FOREACH**

```
# contoh penggunaan FOREACH

database filkom
main
  define rec_mk RECORD LIKE t_matakuliah.*

declare k1 cursor for
select * from t_matakuliah where kd_mk[1,2]='DU"

foreach k1 INTO rec_mk.*
  display 'Kode Matakuliah : ',rec_mk.kd_mk
  display 'Nama Matakuliah : ',rec_mk.nm_mk
```

```
display 'Sks           ':',rec_mk.sks
end foreach
end main
```

CONTINUE FOREACH

Statement IF dapat digunakan bersama statement CONTINUE FOREACH untuk menghentikan urutan statement yang ada pada FOREACH loop, dan melanjutkan pembacaan baris berikutnya.

```
#contoh penggunaan statement 'continue foreach'
database filkom
main
  define rec_mhs RECORD LIKE t_master.*

  declare k2 cursor for
    select * from t_master

  foreach k2 into rec_mhs.*
    if rec_mhs.kota = 'Jakarta' then
      continue foreach
    end if
    display 'N P M   ':',rec_mhs.npm
    display 'Nama   ':',rec_mhs.nama
    display 'Kota   ':',rec_mhs.kota
    display ' '
  end foreach

end main
```

EXIT FOREACH

Statement IF dapat digunakan bersama statement EXIT FOREACH untuk menghentikan urutan statement yang ada pada 'foreach loop' , serta keluar dari 'foreach loop'

```
# contoh penggunaan 'exit foreach'
database filkom
main
  define rec_mhs RECORD LIKE t_master.* ,jw char(1)
```

```

declare k3 cursor for
select * from t_master

foreach k3 into rec_mhs.*
    display 'N P M   : ',rec_mhs.npm
    display 'Nama    : ',rec_mhs.nama
    display 'Alamat  : ',rec_mhs.alamat

    prompt 'Ingin lihat data berikutnya ?[y/t] : ' for jw
    if jw matches'[Tt]' then
        exit foreach
    end if
end foreach

end main

```

Mendapatkan row dengan OPEN , FETCH dan CLOSE

Statement **foreach** diatas, sebenarnya mengkombinasikan beberapa fungsi statement tunggal :

- melakukan OPEN cursor yang terkait dengan statement SELECT
- Secara berulang melakukan FETCH (mengambil) baris, dan menjalankan sejumlah statement yang ada pada FOREACH loop.
- Melakukan CLOSE cursor, setelah seluruh baris terbaca seluruhnya.

OPEN

Untuk melakukan aktif set dari statement SELECT yang dikaitkan dengan suatu cursor.

```

DECLARE kur CURSOR FOR
    SELECT * FROM t_matakuliah WHERE jenis = 'P'
OPEN kur

```

Informix-4GL akan mencari baris sesuai statement diatas dan meletakkan cursor pada posisi awal, sebelum baris pertama.

DK-14301	Pemrog.Gene.4	3	P
TI-17204	Sistem Pakar	2	P
TI-18201	Automata	2	P

FETCH

Untuk menggerakkan cursor pada baris berikutnya dalam suatu aktif set, dan mengambil informasi pada baris tersebut.

```
#contoh penggunaan statement fetch
```

```
FETCH kur INTO rec_mk.*
```

Output dari statement diatas, sbb.

DK-14301	Pemrog.Gene.4	3	P	← cursor
TI-17204	Sistem Pakar	2	P	
TI-18201	Automata	2	P	

Jika FETCH dijalankan sekali lagi, maka cursor akan berpindah ke baris kedua.

Bila cursor berada setelah baris terakhir, maka tidak ada lagi informasi yang akan diambil. Pada saat yang sama Informix akan mengisi variabel status dengan informasi NOT FOUND, sebagai tanda bahwa cursor sudah berada di 'luar batas'.

DK-14301	Pemrog.Gene.4	3	P	
TI-17204	Sistem Pakar	2	P	
TI-18201	Automata	2	P	
Status = NOT FOUND			cursor	←

Karena kita tidak dapat memperkirakan berapa jumlah baris dalam aktif set, maka kita harus selalu memeriksa isi variabel status, segera setelah statement FETCH.

Contoh :

```
FETCH kur INTO rec_mk.*
IF status = notfound then
  Message 'Tidak ada data'
ELSE
  display 'Kode Matakuliah :',rec_mk.kd_mk
  Display 'Nama Matakuliah :',rec_mk.nm_mk
END IF
```

CLOSE

=====

Jika sudah tidak membutuhkan informasi atas baris-baris yang diperoleh dari statement SELECT, maka gunakan statement CLOSE atas kursor yang terkait.

```
#contoh penggunaan statement Close

CLOSE kur
```

Dapat menggunakan WHILE loop untuk pengulangan pembacaan (FETCH) suatu baris dan memeriksa variabel status selama seluruh baris yang ada dalam aktif set diproses.

Sintaks :

WHILE ekspresi Boolean

.....
.....

[CONTINUE WHILE]

.....

.....

[EXIT WHILE]

END WHILE

#contoh 1

Database filkom

Define r_mk RECORD LIKE t_mk.*,jawab char(1)

Main

DECLARE kurs CURSOR FOR

Select * from t_mk where nm_mk[1]='A'

OPEN kurs

Let jawab='y'

WHILE jawab matches'[Yy]'

FETCH kurs INTO r_mk.*

IF status = notfound then

Message 'Data tidak ada'

EXIT WHILE

END IF

Display 'Nama Matakuliah : ',r_mk.nm_mk

Display 'Kode Matakuliah : ',r_mk.kd_mk

Prompt 'Lihat data selanjutnya ? [y/t]' for jawab

ENDWHILE

CLOSE kurs

End main

#contoh 2, untuk memastikan suatu aktif set itu kosong

Database filkom

Define r_mk RECORD LIKE t_mk.*,jawab char(1)

Main

DECLARE kurs SCROLL CURSOR FOR


```
Select * from t_mk where nm_mk[1]='B'
```

OPEN kurs

```
FETCH FIRST kurs INTO r_mk.*
```

```
  If status = notfound then
```

```
    Message 'Tidak ada data pada aktif set'
```

```
  Else
```

```
    Message 'Minimal ada satu data pada aktif set'
```

```
  End if
```

```
CLOSE kurs
```

```
End main
```

NEXT :menandakan baris berikutnya setelah baris aktif (active set)

PREVIOUS :menandakan baris sebelumnya dari baris aktif

FIRST : menandakan baris pertama pada baris aktif

LAST : menandakan baris terakhir pada baris aktif

CURRENT : menandakan pada baris yg aktif pada active list

RELATIVE m : menandakan baris ke-m relatif pada kursor yg aktif

ABSOLUTE n : menandakan baris ke-n, n bias integer atau variabel dari program

ARRAY

Array merupakan kumpulan record dengan tipe yang sama

Sintaks :

```
array_name ARRAY[n] OF RECORD  
      Var_list data_type[.....]  
END RECORD
```

n = bilangan integer, merupakan jumlah record yg dapat ditampung oleh array tersebut.

```
#contoh  
ar_krs ARRAY[12] OF RECORD  
      npm1 like t_nilai.npm,  
      nama1 like t_nilai.nama,  
      kd_mk1 like t_nilai.kd_mk,  
      nm_mk1 like t_nilai.nm_mk  
END RECORD
```

```
#contoh  
ar_mhs ARRAY[150] OF RECORD LIKE t_master.*
```

```
#contoh program lengkap tentang ARRAY  
Database filkom
```

```

Main
Define ar_mhs ARRAY[150] OF RECORD LIKE t_master.* , i,j smallint
DECLARE k1 CURSOR FOR
SELECT * from t_master where kelas[1]='4'

Let i=1
FOREACH k1 INTO ar_mhs[i].* Let i = i + 1 END FOREACH

Let j = i - 1
FOR i = 1 to j Display 'Data ke-',i,' ',ar_mhs[i].* END FOR
End main

```

PENGUNAAN FUNCTION

Mendefinisikan function

Function adalah satu routine yang merupakan bagian dari modul program 4GL. Function diletakkan diluar routine MAIN

```

main
.....
.....
end main

FUNCTION nama_function( )
.....
.....
END FUNCTION

```

#contoh penggalan program yg menggunakan function

```

function jum_mhs( )
  define jum integer,
        kls char(5)

  prompt 'menghitung jumlah mahasiswa kelas ' for kls

  # perintah sql untuk menghitung
  select count(*) into jum
  from t_master
  where kelas = kls

  display 'Banyaknya mahasiswa kelas ',kls,'=', jum
end function

```

Memanggil function

Untuk menggunakan function yang telah didefinisikan, dapat dilakukan dengan memanggil function tersebut dari dalam MAIN atau dari dalam FUNCTION lain. Perintah yg diperlukan untuk memanggil function adalah 'CALL'

```
Main
.....
.....
    CALL nama_function( )
.....
End main
```

```
# contoh program untuk memanggil function
```

```
main
    call berita( )
end main

function berita( )
    display 'lagi belajar function'
end function
```

Pengertian variable global, variable module & variable lokal

Routine adalah seperti MAIN, REPORT dan FUNCTION dapat mempunyai **tiga jenis tingkatan variable** terhadap data yaitu **global**, **module** dan **local**.

Variable global

Variable yang dapat dikenal dan diterima diseluruh program

Variable module

Variable yang dikenal disemua routine yang terletak didalam program 4gl yang sama

Variable local

Variable ini hanya berlaku dan dimengerti didalam routine yang mendefinisikannya

Skema lingkup data variable

```
Globals
    Define var A      modul 1

Main
    Define var B      modul 2
End main
Define var C
```

```

Function x
  Define var D      modul 3
Function y
  Define var E

```

lokasi	Nama variabel	type	Scope
Globals	Var A	Global	Main Function x Function y
Main	Var B	Local	Main
Module	Var C	Module	Function x Function y
Function x	Var D	Local	Function x
Function y	Var E	Local	Function y

Contoh :

<pre> Globals Define a integer End globals Main Let a = 0 Call ganti() Display a End main Function ganti() Let a = 99 End function #outputnya = 99 </pre>	<pre> Globals Define a integer End globals Main Let a = 0 Call ganti() Display a End main Function ganti() Define a integer Let a = 99 End function # outputnya = 0 </pre>
--	---

Parameter

Jenis variable atau nilai yang terletak antara tanda kurung () pada perintah FUNCTION dan berhubungan dengan perintah CALL.

Parameter ini digunakan untuk melewati / mengirim nilai/informasi dari MAIN ke FUNCTION tanpa mempengaruhi nilainya.

PASSING VALUE (melewatkan/mengirim nilai melalui function)

Untuk mengirim/melewatkan nilai melalui function, digunakan **parameter** pada function yang terletak antara tanda kurung ().

Variable yang menerima nilai/informasi disebut **variable argument**.

Parameter harus didefinisikan juga dengan define.

```

Function tampil (nama,alamat)
  Define nama char(25), alamat char(30)
  .....
  .....
End function

```

```

Function lihat (rec_mhs)          #baca catatan ttg record
  Define rec_mhs record like t_master.*
  .....
  .....
End function

```

```

Main
  Define nama,kota char(30)  #2 var bertipe sama
  Let nama = 'audi'
  Let kota = 'Bandung'
  Call lihat(nama,kota)      # passing value
End main

Function lihat(nm,kt)        # argument value
  Define nm,kt char(30)
  Display 'nama = ',nm at 7,10
  Display 'kota = ',kt at 8,10
End function

```

RETURN VALUE

Mengembalikan nilai dari function kembali ke routine yang memanggilnya.

```

Main
  Define juml integer

  Call hitung( ) RETURNING juml
  Display 'jumlah = ',juml at 10,10

End main

Function hitung( )
  Define jml integer
  Select count(*) into jml from t_master
  RETURN jml
End function

```

Penggunaan PASSING VALUE & RETURN VALUE

```

Database filkom
Main
  Define npm char(8), nama char(25),

```

Alamat char(30), kota char(15)

Prompt 'tampilkan data mahasiswa , npm = ' for npm

Call cari(npm) **returning** nama,alamat,kota

Display 'N P M = ',npm at 15,5

Display 'Nama = ',nama at 16,5

Display 'Alamat = ',alamat at 17,5

Display 'Kota = ',kota at 18,5

End main

Function cari(np)

Define np char(8), nm char(25),
alm char(30), kt char(15)

```
select nama,alamat,kota  
into nm , alm, kt  
from t_master  
where npm = np
```

#perintah sql diatas bisa ditulis dalam satu baris seperti dibawah ini

#select nama,alamat,kota into nm,alm,kt from t_master where npm=np

```
return nm, alm, kt  
end function
```