

# Perancangan *Database* pada Interbase Server (Studi Kasus: Sistem Informasi Sirkulasi Perpustakaan)

**Riyanto**

*mohriyan@gmail.com*

*http://www.masto.co.cc*

## ***Lisensi Dokumen:***

*Copyright © 2003-2007 IlmuKomputer.Com*

*Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.*

Sebagaimana MySQL, Interbase juga memiliki kemampuan mengelola data skala *enterprise*. Meskipun penggunaanya tidak sebanyak MySQL, tetapi tidak sedikit masyarakat TI, termasuk perusahaan yang menggunakan Interbase. Tutorial kali ini akan membahas perancangan *database* untuk Sistem Informasi Sirkulasi Perpustakaan menggunakan berbagai objek *database* yang didukung Interbase.

Tutorial ini tidak hanya dapat dimanfaatkan untuk pengguna Interbase, tapi juga FireBird. *Database* FireBird memiliki arsitektur yang sangat mirip dengan Interbase, sehingga tidak heran bila FireBird sering disebut sebagai “kloning”-nya Interbase. FireBird juga sering dikenal dengan versi *open source*-nya Interbase.

## **Pendahuluan**

Interbase adalah sebuah *relational database management system* (RDBMS) yang mampu melakukan pemrosesan transaksi dengan cepat dan menggunakan data secara bersama baik dalam lingkungan *single user* ataupun *multi user*.

Interbase menggunakan sebuah *data definition language* (DDL) untuk mendefinisikan berbagai objek *database* yang berisi informasi mengenai struktur *database* beserta datanya. Objek-objek ini sering juga disebut sebagai “*metadata*”.

### **A. Sintaks untuk Membuat *Database***

```
CREATE {DATABASE | SCHEMA} 'filespec'  
[USER 'username' [PASSWORD 'password']]  
[PAGE_SIZE [=] int]  
[LENGTH [=] int [PAGE[S]]]  
[DEFAULT CHARACTER SET charset]  
[<secondary_file>;]
```

```
<secondary_file> = FILE 'filespec' [<fileinfo>] [  
<secondary_file>]  
<fileinfo> = [LENGTH [=] int [PAGE[S]] | STARTING [AT [PAGE]] int ]  
[<fileinfo>]
```

**contoh:**

```
CREATE DATABASE 'LA.gdb'  
USER 'SYSDBA' PASSWORD 'masterkey';
```

## B. Sintaks untuk Membuat *Domain*

```
CREATE DOMAIN domain [AS] <datatype>  
[DEFAULT {literal | NULL | USER}]  
[NOT NULL] [CHECK (<dom_search_condition>)]  
[COLLATE collation];  
  
<datatype> =  
{SMALLINT|INTEGER|FLOAT|DOUBLE PRECISION}[<array_dim>]  
| {DATE|TIME|TIMESTAMP}[<array_dim>]  
| {DECIMAL | NUMERIC} [(precision [, scale])] [<array_dim>]  
| {CHAR | CHARACTER | CHARACTER VARYING | VARCHAR} [(int)]  
  [<array_dim>] [CHARACTER SET charname]  
| {NCHAR | NATIONAL CHARACTER | NATIONAL CHAR}  
  [VARYING] [(int)] [<array_dim>]  
| BLOB [SUB_TYPE {int | subtype_name}] [SEGMENT SIZE int]  
  [CHARACTER SET charname]  
| BLOB [(seglen [, subtype])]  
<array_dim> = [[x:]y [, [x:]y ...]]  
<dom_search_condition> = {  
VALUE <operator> value  
| VALUE [NOT] BETWEEN value AND value  
| VALUE [NOT] LIKE value [ESCAPE value]  
| VALUE [NOT] IN (value [, value ...])  
| VALUE IS [NOT] NULL  
| VALUE [NOT] CONTAINING value  
| VALUE [NOT] STARTING [WITH] value  
| (<dom_search_condition>)  
| NOT <dom_search_condition>  
| <dom_search_condition> OR <dom_search_condition>  
| <dom_search_condition> AND <dom_search_condition>  
}  
<operator> = {= | < | > | <= | >= | !< | !> | <> | !=}
```

**contoh:**

```
/* Domain definition */  
CREATE DOMAIN dCALLNO AS VARCHAR(20);  
CREATE DOMAIN dNOANGGOTA AS INTEGER  
CHECK (VALUE > 0);  
CREATE DOMAIN dBIAYA AS NUMERIC (10, 2)  
CHECK (VALUE >= 0);
```

## C. Sintaks untuk Membuat *Table*

```
CREATE TABLE table [EXTERNAL [FILE] 'filespec']  
(<col_def> [, <col_def> | <tconstraint> ...]);  
<col_def> = col {<datatype> | COMPUTED [BY] (<expr>) | domain}  
[DEFAULT {literal | NULL | USER}]  
[NOT NULL]  
[<col_constraint>]  
[COLLATE collation]  
  
<datatype> =  
{SMALLINT | INTEGER | FLOAT | DOUBLE PRECISION}[<array_dim>]  
| {DATE | TIME | TIMESTAMP}[<array_dim>]
```

```

| {DECIMAL | NUMERIC} [(precision [, scale])] [<array_dim>]
| {CHAR | CHARACTER | CHARACTER VARYING | VARCHAR} [(int)]
    [<array_dim>] [CHARACTER SET charname]
| {NCHAR | NATIONAL CHARACTER | NATIONAL CHAR}
    [VARYING] [(int)] [<array_dim>]
| BLOB [SUB_TYPE {int | subtype_name}] [SEGMENT SIZE int]
    [CHARACTER SET charname]
| BLOB [(seglen [, subtype])]<array_dim> = [[x:]y [, [x:]y ...]]
<expr> = A valid SQL expression that results in a single value.

<col_constraint> = [CONSTRAINT constraint]
{ UNIQUE
| PRIMARY KEY
| REFERENCES other_table [(other_col [, other_col ...])]
    [ON DELETE {NO ACTION|CASCADE|SET DEFAULT|SET NULL}]
    [ON UPDATE {NO ACTION|CASCADE|SET DEFAULT|SET NULL}]
| CHECK (<search_condition>)}

<tconstraint> = [CONSTRAINT constraint]
{{PRIMARY KEY | UNIQUE} (col [, col ...])
| FOREIGN KEY (col [, col ...]) REFERENCES other_table
    [ON DELETE {NO ACTION|CASCADE|SET DEFAULT|SET NULL}]
    [ON UPDATE {NO ACTION|CASCADE|SET DEFAULT|SET NULL}]

| CHECK (<search_condition>)}

<search_condition> = <val> <operator> {<val> | (<select_one>)}
| <val> [NOT] BETWEEN <val> AND <val>
| <val> [NOT] LIKE <val> [ESCAPE <val>]
| <val> [NOT] IN (<val> [, <val> ...] | <select_list>)
| <val> IS [NOT] NULL
| <val> {>= | <=}
| <val> [NOT] {= | < | >}
| {ALL | SOME | ANY} (<select_list>)
| EXISTS (<select_expr>)

| SINGULAR (<select_expr>)
| <val> [NOT] CONTAINING <val>
| <val> [NOT] STARTING [WITH] <val>
| (<search_condition>)
| NOT <search_condition>
| <search_condition> OR <search_condition>
| <search_condition> AND <search_condition>
<val> = { col [<array_dim>] | :variable
| <constant> | <expr> | <function>
| udf ([<val> [, <val> ...]])
| NULL | USER | RDB$DB_KEY | ? }

[COLLATE collation]
<constant> = num | 'string' | charsetname 'string'

<function> = COUNT (* | [ALL] <val> | DISTINCT <val>)
| SUM ([ALL] <val> | DISTINCT <val>)
| AVG ([ALL] <val> | DISTINCT <val>)
| MAX ([ALL] <val> | DISTINCT <val>)
| MIN ([ALL] <val> | DISTINCT <val>)
| CAST (<val> AS <datatype>)
| UPPER (<val>)

| GEN_ID (generator, <val>)
<operator> = {= | < | > | <= | >= | !< | !> | <> | !=}
<select_one> = SELECT on a single column; returns exactly one value.
<select_list> = SELECT on a single column; returns zero or more values.
<select_expr> = SELECT on a list of values; returns zero or more values.

```

**Contoh:**

```
CREATE TABLE "ANGGOTA"  
(  
  "NO_ANGGOTA" "DNOANGGOTA" NOT NULL,  
  "NAMA" VARCHAR(25) NOT NULL,  
  "KATEGORI_KEANGGOTAAN" VARCHAR(25) NOT NULL,  
  "ALAMAT" VARCHAR(60),  
  "KOTA" VARCHAR(25) NOT NULL,  
  "TGL_AKHIR_KEANGGOTAAN" DATE NOT NULL,  
  PRIMARY KEY ("NO_ANGGOTA")  
);
```

**D. Sintaks untuk Membuat View**

```
CREATE VIEW name [(view_col [, view_col ...])]  
AS <select> [WITH CHECK OPTION];
```

**Contoh:**

```
/*view: BORROWED_BOOKS, owner:SYSDBA */  
CREATE VIEW BORROWED_BOOKS As  
  SELECT S.CALL_NO, K.JUDUL, S.NO_ANGGOTA, A.NAMA  
  FROM SIRKULASI S, ANGGOTA A, KATALOG K  
  WHERE (A.NO_ANGGOTA=S.NO_ANGGOTA) AND (K.CALL_NO=S.CALL_NO) ;
```

**Persiapan Data**

Berikut disajikan cara pembuatan tabel dan relasi antar tabel, serta contoh data (*record*) yang akan digunakan untuk pemrosesan *query* pada bahasan berikutnya.

**A. Pembuatan Tabel-tabel**

▪ **Tabel Tjenis Buku:**

```
CREATE TABLE TJENIS_BUKU  
(  
  DCC VARCHAR(10) NOT NULL,  
  JENIS_BUKU VARCHAR(15),  
  PRIMARY KEY (DCC)  
);
```

▪ **Tabel Tdaftar\_Buku :**

```
CREATE TABLE TDAFTAR_BUKU  
(  
  KODE_BUKU VARCHAR(4) NOT NULL,  
  ISBN VARCHAR(10) NOT NULL,  
  PRIMARY KEY (KODE_BUKU)  
);
```

▪ **Tabel Tpengarang :**

```
CREATE TABLE TPENGARANG  
(  
  KODE_PENGARANG VARCHAR(2) NOT NULL,  
  NAMA_PENGARANG VARCHAR(25),  
  PRIMARY KEY (KODE_PENGARANG)  
);
```

▪ **Tabel Tpenerbit :**

```
CREATE TABLE TPENERBIT  
(  
  KODE_PENERBIT VARCHAR(2) NOT NULL,  
  NAMA_PENERBIT VARCHAR(25),  
  PRIMARY KEY (KODE_PENERBIT)  
);
```

▪ **Tabel Tinfo\_Buku :**

```
CREATE TABLE TINFO_BUKU
(
  ISBN VARCHAR(10) NOT NULL,
  DCC VARCHAR(10),
  JUDUL VARCHAR(20),
  KODE_PENGARANG VARCHAR(2),
  KODE_PENERBIT VARCHAR(2),
  TH_TERBIT VARCHAR(4),
  PRIMARY KEY (ISBN)
);
```

▪ **Tabel Tprodi :**

```
CREATE TABLE TPRODI
(
  KODE_PRODI VARCHAR(2) NOT NULL,
  NAMA_PRODI VARCHAR(20),
  JURUSAN VARCHAR(10),
  FAKULTAS VARCHAR(10),
  PRIMARY KEY (KODE_PRODI)
);
```

▪ **Tabel Tanggota :**

```
CREATE TABLE TANGGOTA
(
  NO_ANGGOTA VARCHAR(4) NOT NULL,
  NIM VARCHAR(20),
  NAMA VARCHAR(25),
  ALAMAT VARCHAR(50),
  JENIS_KELAMIN VARCHAR(1),
  ANGKATAN VARCHAR(2),
  KODE_PRODI VARCHAR(2),
  PRIMARY KEY (NO_ANGGOTA)
);
```

▪ **Tabel Tsirkulasi**

```
CREATE TABLE TSIRKULASI
(
  KODE_BUKU VARCHAR(4),
  NO_ANGGOTA VARCHAR(4),
  TGL_PINJAM DATE,
  TGL_HRS_KEMBALI DATE,
  TGL_KEMBALI DATE
);
```

**B. Pembuatan Foreign Key**

▪ **Foreign Key pada Tabel TInfo\_Buku**

```
ALTER TABLE TINFO_BUKU ADD CONSTRAINT FK_DCC FOREIGN KEY (DCC)
REFERENCES TJENIS_BUKU (DCC)
ON UPDATE SET DEFAULT ON DELETE SET NULL;
ALTER TABLE TINFO_BUKU ADD CONSTRAINT FK_PENERBIT FOREIGN KEY (KODE_PENERBIT)
REFERENCES TPENERBIT (KODE_PENERBIT)
ON UPDATE SET DEFAULT ON DELETE SET DEFAULT;
ALTER TABLE TINFO_BUKU ADD CONSTRAINT FK_PENGARANG FOREIGN KEY
(KODE_PENGARANG)
REFERENCES TPENGARANG (KODE_PENGARANG)
ON UPDATE SET DEFAULT ON DELETE SET DEFAULT;
```

▪ **Foreign Key Pada Tabel TAnggota**

```
ALTER TABLE TANGGOTA ADD CONSTRAINT FK_PRODI FOREIGN KEY (KODE_PRODI)
REFERENCES TPRODI (KODE_PRODI)
ON UPDATE CASCADE ON DELETE CASCADE;
```

### Foreign Key Pada Tabel TSirkulasi

```
ALTER TABLE TSIRKULASI ADD CONSTRAINT FK_ANGGOTA FOREIGN KEY (NO_ANGGOTA)
REFERENCES TANGGOTA (NO_ANGGOTA)
ON UPDATE CASCADE ON DELETE SET DEFAULT;
ALTER TABLE TSIRKULASI ADD CONSTRAINT FK_BUKU FOREIGN KEY (KODE_BUKU)
REFERENCES TDAFTAR_BUKU (KODE_BUKU)
ON UPDATE CASCADE ON DELETE SET DEFAULT;
```

## C. Pengisian Record pada Tabel

### Tabel Tjenis\_Buku

```
Insert into Tjenis_Buku Values ('1','Komputer');
Insert into Tjenis_Buku Values ('2','Bisnis');
Insert into Tjenis_Buku Values ('3','Agama');
Insert into Tjenis_Buku Values ('4','Masakan');
Insert into Tjenis_Buku Values ('5','Psikologi');
Commit;
```

Output :

DCC	JENIS_BUKU
1	Komputer
2	Bisnis
3	Agama
4	Masakan
5	Psikologi

### Tabel TDaftar\_Buku

```
Insert into TDaftar_Buku Values ('1','11-111-222');
Insert into TDaftar_Buku Values ('10','11-111-221');
Insert into TDaftar_Buku Values ('11','11-111-220');
Insert into TDaftar_Buku Values ('12','11-111-231');
Insert into TDaftar_Buku Values ('2','11-111-223');
Insert into TDaftar_Buku Values ('3','11-111-224');
Insert into TDaftar_Buku Values ('4','11-111-225');
Insert into TDaftar_Buku Values ('5','11-111-226');
Insert into TDaftar_Buku Values ('6','11-111-227');
Insert into TDaftar_Buku Values ('7','11-111-228');
Insert into TDaftar_Buku Values ('8','11-111-229');
Insert into TDaftar_Buku Values ('9','11-111-230');
Commit;
```

Output:

KODE_BUKU	ISBN
1	11-111-222
10	11-111-221
11	11-111-220
12	11-111-231
2	11-111-223
3	11-111-224
4	11-111-225
5	11-111-226
6	11-111-227
7	11-111-228
8	11-111-229
9	11-111-230

#### ▪ Tabel TPengarang

```
Insert into TPengarang Values ('1','Antony Pranata');
Insert into TPengarang Values ('2','Yahya Kurniawan');
Insert into TPengarang Values ('3','Onno W Purbo');
Insert into TPengarang Values ('4','Abdul Kadir');
Insert into TPengarang Values ('5','Alusius Yoga Phd');
Insert into TPengarang Values ('6','Ambaldy Juardi');
Insert into TPengarang Values ('7','Kenneth Cragg');
Insert into TPengarang Values ('8','Yuniarti');
Insert into TPengarang Values ('9','Ferry Santoso');
Commit;
```

#### Output:

	KODE_PENGARANG	NAMA_PENGARANG
▶	1	Antony Pranata
	2	Yahya Kurniawan
	3	Onno W Purbo
	4	Abdul Kadir
	5	Alusius Yoga Phd
	6	Ambaldy Juardi
	7	Kenneth Cragg
	8	Yuniarti
	9	Ferry Santoso

#### ▪ Tabel TPenerbit

```
Insert into TPenerbit Values ('1','Elek Media Komputindo');
Insert into TPenerbit Values ('2','Andi Offset');
Insert into TPenerbit Values ('3','Toha Putra');
Insert into TPenerbit Values ('4','Erlangga');
Insert into TPenerbit Values ('5','Eksekutif Press');
Insert into TPenerbit Values ('6','Gava Media');
Commit;
```

#### Output:

	KODE_PENERBIT	NAMA_PENERBIT
▶	1	Elek Media Komputindo
	2	Andi Offset
	3	Toha Putra
	4	Erlangga
	5	Eksekutif Press
	6	Gava Media

#### ▪ Tabel Tpenerbit

```
Insert into TProdi Values ('1','Ilkom','Matematika','MIPA');
Insert into TProdi Values ('2','Komsis','Matematika','MIPA');
Insert into TProdi Values ('3','Geofisika','Fisika','MIPA');
Insert into TProdi Values ('4','Kimia','Kimia','MIPA');
Insert into TProdi Values ('5','Statistika','Matematika','MIPA');
Insert into TProdi Values ('6','Fisika','Fisika','MIPA');
Insert into TProdi Values ('7','Matematika','Matematika','MIPA');
Commit;
```

**Output :**

	KODE_PRODI	NAMA_PRODI	JURUSAN	FAKULTAS
▶	1	Ilkom	Matematika	MIPA
	2	Komsi	Matematika	MIPA
	3	Geofisika	Fisika	MIPA
	4	Kimia	Kimia	MIPA
	5	Statistika	Matematika	MIPA
	6	Fisika	Fisika	MIPA
	7	Matematika	Matematika	MIPA

▪ **Tabel Tinfo\_Buku**

```

Insert into Tinfo_Buku Values('11-111-220','5','Psikologi Remaja',
'5','5','2000');
Insert into Tinfo_Buku Values('11-111-221','5','Etika Psikologi',
'5','5','2001');
Insert into Tinfo_Buku Values('11-111-222','1','Pemrograman Delphi',
'1','1','2003');
Insert into Tinfo_Buku Values('11-111-223','1','Visual Basic','1','1','2000');
Insert into Tinfo_Buku Values('11-111-224','1','Web dengan PHP',
'1','2','2003');
Insert into Tinfo_Buku Values('11-111-225','1','Database Interbase',
'3','2','1999');
Insert into Tinfo_Buku Values('11-111-226','1','Macromedia MX',
'4','6','1998');
Insert into Tinfo_Buku Values('11-111-227','2','E-Commerce','2','1','2004');
Insert into Tinfo_Buku Values('11-111-228','2','Kiat Cari Uang',
'6','4','2002');
Insert into Tinfo_Buku Values('11-111-229','3','Kearifan Sang Sufi',
'7','3','2002');
Insert into Tinfo_Buku Values('11-111-231','5','Filsafat Aristoteles',
'9','3','1999');
Commit;
```

**Output:**

	ISBN	DCC	JUDUL	KODE_PENGARANG	KODE_PENERBIT	TH_TERBIT
▶	11-111-220	5	Psikologi Remaja	5	5	2000
	11-111-221	5	Etika Psikologi	5	5	2001
	11-111-222	1	Pemrograman Delphi	1	1	2003
	11-111-223	1	Visual Basic	1	1	2000
	11-111-224	1	Web dengan PHP	1	2	2003
	11-111-225	1	Database Interbase	3	2	1999
	11-111-226	1	Macromedia MX	4	6	1998
	11-111-227	2	E-Commerce	2	1	2004
	11-111-228	2	Kiat Cari Uang	6	4	2002
	11-111-229	3	Kearifan Sang Sufi	7	3	2002
	11-111-231	5	Filsafat Aristoteles	9	3	1999

▪ **Tabel Tanggota :**

```

Insert into Tanggota Values('1','1390','Riyanto','Pati','0','02','1');
Insert into Tanggota Values('2','1391','Sitepu','Medan','0','00','1');
Insert into Tanggota
Values('3','1380','Kartiningrum','Tangerang','1','01','2');
Insert into Tanggota
Values('4','1382','Dimas Chandra','Jakarta','0','02','3');
```



```
Insert into Tanggota
Values ('5','1406','Dimas Mukhlas','Yogyakarta','0','99','4');
Insert into Tanggota
Values ('6','1408','Dian Kridawati','Yogyakarta','1','03','4');
Insert into Tanggota
Values ('7','1472','Anang Susilo','Yogyakarta','0','02','4');
Commit;
```

### Output:

	NO_ANGGOTA	NIM	NAMA	ALAMAT	JENIS_KELAMIN	ANGKATAN	KODE_PRODI
▶	1	1390	Riyanto	Pati	0	02	1
	2	1391	Sitepu	Medan	0	00	1
	3	1380	Kartiningrum	Tangerang	1	01	2
	4	1382	Dimas Chandra	Jakarta	0	02	3
	5	1406	Dimas Mukhlas	Yogyakarta	0	99	4
	6	1408	Dian Kridawati	Yogyakarta	1	03	4
	7	1472	Anang Susilo	Yogyakarta	0	02	4

### ▪ Tabel TSirkulasi

```
Insert into Tsirkulasi Values ('6','2','02/06/2003','07/06/2003','05/06/2003');
Insert into Tsirkulasi Values ('2','1','10/06/2003','15/06/2003','16/06/2003');
Insert into Tsirkulasi Values ('4','1','07/05/2003','12/05/2003','10/05/2003');
Insert into Tsirkulasi Values ('3','2','08/05/2003','13/05/2003','14/05/2003');
Insert into Tsirkulasi Values ('1','3','14/06/2003','19/06/2003','20/06/2003');
Insert into Tsirkulasi Values ('5','4','05/05/2003','10/05/2003','12/05/2003');
Insert into Tsirkulasi Values ('9','5','24/05/2003','29/05/2003','30/05/2003');
Insert into Tsirkulasi Values ('6','6','05/05/2003','10/05/2003','10/05/2003');
Insert into Tsirkulasi Values ('8','7','13/05/2003','18/05/2003','17/05/2003');
Insert into Tsirkulasi Values ('7','7','20/05/2003','25/05/2003','24/05/2003');
Insert into Tsirkulasi Values ('6','7','02/06/2003','07/06/2003','17/06/2003');
Insert into Tsirkulasi Values ('1','5','14/06/2003','19/06/2003','18/06/2003');
Insert into Tsirkulasi Values ('1','6','14/06/2003','19/06/2003','17/06/2003');
Insert into Tsirkulasi Values ('1','4','14/06/2003','19/06/2003','18/06/2003');
Commit;
```

### Output:

	KODE_BUKU	NO_ANGGOTA	TGL_PINJAM	TGL_HRS_KEMBALI	TGL_KEMBALI
▶	6	2	02/06/2003	07/06/2003	05/06/2003
	2	1	10/06/2003	15/06/2003	16/06/2003
	4	1	07/05/2003	12/05/2003	10/05/2003
	3	2	08/05/2003	13/05/2003	14/05/2003
	1	3	14/06/2003	19/06/2003	20/06/2003
	5	4	05/05/2003	10/05/2003	12/05/2003
	9	5	24/05/2003	29/05/2003	30/05/2003
	6	6	05/05/2003	10/05/2003	10/05/2003
	8	7	13/05/2003	18/05/2003	17/05/2003
	7	7	20/05/2003	25/05/2003	24/05/2003
	6	7	02/06/2003	07/06/2003	17/06/2003
	1	5	14/06/2003	19/06/2003	18/06/2003
	1	6	14/06/2003	19/06/2003	17/06/2003
	1	4	14/06/2003	19/06/2003	18/06/2003

## Pemrosesan *Query*

Bahasan berikut akan menyajikan berbagai implementasi *query* yang sering digunakan pada sistem informasi sirkulasi perpustakaan. Penyajian bahasan berupa contoh berbagai jenis permasalahan dan bagaimana penyelesaiannya dengan perintah SQL.

### 1. Menampilkan buku yang paling sering dipinjam (buku terlaris).

#### Sintaks SQL :

```
/* membuat view vSirkul */

Create View vSirkul(Kode_Buku, Jumlah) As
Select kode_buku, count(*) As jumlah From tsirkulasi
Group by kode_buku;

/* mencari buku terlaris dengan memanfaatkan view diatas (Sintak_1.sql)*/

Select Max(t.jumlah), i.judul, p.nama_pengarang, pt.nama_penerbit
From vSirkul t, tinfo_buku i, tpengarang p,
tpenerbit pt, tdaftar_buku d
Where t.kode_buku=d.kode_buku
And i.kode_pengarang=p.kode_pengarang
And i.kode_penerbit=pt.kode_penerbit And d.isbn=i.isbn
And t.jumlah>=all(Select jumlah From vSirkul)
Group by i.judul, p.nama_pengarang, pt.nama_penerbit;
```

#### Output :

	MAX	JUDUL	NAMA_PENGARANG	NAMA_PENERBIT
▶	4	Pemrograman Delphi	Antony Pranata	Elek Media Komputindo

### 2. Menampilkan jumlah buku yang dipinjam pada tanggal tertentu

#### Sintaks SQL : Sintak\_2.sql)

```
Select distinct tgl_pinjam, kode_buku, count(*) As jml_per_tgl
From tsirkulasi
Group by tgl_pinjam, kode_buku;
```

#### Output :

	TGL_PINJAM	KODE_BUKU	JML_PER_TGL
▶	05/05/2003	5	1
	05/05/2003	6	1
	07/05/2003	4	1
	08/05/2003	3	1
	13/05/2003	8	1
	20/05/2003	7	1
	24/05/2003	9	1
	02/06/2003	6	2
	10/06/2003	2	1
	14/06/2003	1	4

### 3. Menampilkan data anggota serta lama meminjam buku tertentu.

#### Sintaks SQL : (Sintak\_3.sql)

```
Select no_anggota, kode_buku, sum(tgl_kembali - tgl_pinjam)As
lama_pinjam
```

```
From tsirkulasi
Group by no_anggota, kode_buku, tgl_kembali;
```

**Output :**

NO_ANGGOTA	KODE_BUKU	LAMA_PINJAM
1	2	6
1	4	3
2	3	6
2	6	3
3	1	6
4	1	4
4	5	7
5	1	4
5	9	6
6	1	3
6	6	5

4. Menampilkan data anggota yang terlambat mengembalikan dan lama keterlambatan (Dalam hal ini maksimal lama meminjam adalah 5 hari).

**Sintaks SQL :**

```
/* membuat view vLamaPinjam */

Create view vLamaPinjam(no_anggota,kode_buku, lama_pinjam) As
Select no_anggota, kode_buku, sum(tgl_kembali - tgl_pinjam)As
lama_pinjam
From tsirkulasi
Group by no_anggota, kode_buku, tgl_kembali;

/* mencari Daftar Anggota yang terlambat mengembalikan buku dengan
memanfaatkan view diatas (Sintak_4.sql)*/

Select no_anggota, tAnggota>Nama, kode_buku, lama_pinjam,
sum(lama_pinjam-5)As lama_terlambat
From vLamaPinjam, tAnggota
Where vLamaPinjam.no_Anggota=tAnggota.no_Anggota
And (lama_pinjam-5)>0
Group by no_anggota, nama, kode_buku, lama_pinjam;
```

**Output :**

	NO_ANGGOTA	NAMA	KODE_BUKU	LAMA_PINJAM	LAMA_TERLAMBAT
▶	1	Riyanto	2	6	1
	2	Sitepu	3	6	1
	3	Kartiningrum	1	6	1
	4	Dimas Chandra	5	7	2
	5	Dimas Mukhlas	9	6	1
	7	Anang Susilo	6	15	10

5. Menampilkan Kode\_buku dan jumlahnya yang dipinjam pada bulan tertentu

**Sintaks SQL :** (Sintak\_5.sql)

```
Select tgl_pinjam, kode_buku, count(*) As jumlah
From tsirkulasi
```

```
Where tgl_pinjam between '06/01/2003' And '06/30/2003'
Group by tgl_pinjam, kode_buku;
```

#### Output :

	TGL_PINJAM	KODE_BUKU	JUMLAH
▶	02/06/2003	6	2
	10/06/2003	2	1
	14/06/2003	1	4

### 6. Menampilkan Data anggota yang paling aktif

#### Sintaks SQL :

```
/* membuat view vAnggota */

Create view vAnggota(no_anggota, jumlah) As
Select no_anggota, count(*) As jumlah From tsirkulasi
Group by no_anggota;

/* mencari Daftar Anggota Paling Aktif dengan memanfaatkan view diatas
(Sintak_6.sql) */

Select ta.nama, tpr>Nama_Prodi, tpr.Jurusan, tpr.Fakultas,
ta.angkatan, Max(va.jumlah)As Banyak_minjam
From vAnggota va, anggota ta, tprodi tpr
Where va.no_anggota=ta.no_anggota And ta.kode_prodi=tpr.kode_prodi
And va.jumlah>=all(Select jumlah From vAnggota)
Group by ta.nama, ta.angkatan,tpr>Nama_Prodi,
tpr.Jurusan, tpr.Fakultas;
```

#### Output :

	NAMA	NAMA_PRODI	JURUSAN	FAKULTAS	ANGKATAN	BANYAK_MINJAM
▶	Anang Susilo	Kimia	Kimia	MIPA	02	3

### 7. Menampilkan jumlah anggota berdasarkan jenis kelamin

#### Cara 1:

#### Sintaks SQL : (Sintak\_7.sql)

```
Select jenis_kelamin, count(*) As Jumlah From tAnggota
Group by jenis_kelamin;
```

#### Output :

	JENIS_KELAMIN	JUMLAH
▶	0	5
	1	2

#### Cara 2:

#### Sintaks SQL :

```
/* membuat view vPria */

Create View VPria(Jumlah_Pria) As
```

```

Select count(*) As jumlah_Pria From anggota
Where jenis_kelamin='0';

/* membuat view vWanita */

Create View VPria(Jumlah_Wanita) As
Select count(*) As jumlah_Wanita From anggota
Where jenis_kelamin='1';

/* mencari buku terlaris dengan memanfaatkan vPria dan vWanita
(Sintak_8.sql)*/

Select * From vwanita, vpria;

```

#### Output :

	JUMLAH_WANITA	JUMLAH_PRIA
▶	2	5

### 8. Menampilkan Detail buku-buku yang berkategori dan penerbit tertentu.

#### Sintaks SQL : (Sintak\_9.sql)

```

Select tInfo_Buku.Judul, tPengarang>Nama_Pengarang, tPenerbit>Nama_Penerbit,
tInfo_Buku.Th_Terbit, tJenis_Buku.Jenis_Buku
From tInfo_Buku, tPengarang, tPenerbit, tJenis_Buku, tDaftar_Buku
Where tJenis_buku.DCC=tInfo_Buku.DCC
And tInfo_Buku.ISBN=tDaftar_Buku.ISBN
And tInfo_Buku.Kode_Pengarang=tPengarang.Kode_Pengarang
And tInfo_Buku.Kode_Penerbit=tPenerbit.Kode_Penerbit
And tJenis_Buku.Jenis_Buku='Komputer'
And tPenerbit>Nama_Penerbit='Elek Media Komputindo'

Order by Th_Terbit;

```

#### Output :

	JUDUL	NAMA_PENGARANG	NAMA_PENERBIT	TH_TERBIT	JENIS_BUKU
▶	Visual Basic	Antony Pranata	Elek Media Komputindo	2000	Komputer
	Pemrograman Delphi	Antony Pranata	Elek Media Komputindo	2003	Komputer

### 9. Menampilkan Daftar Buku-buku Terbaru yang ada (Terbit pada tahun termuda/terbesar)

#### Sintaks SQL : (Sintak\_10.sql)

```

SELECT ISBN, Judul, Nama_Pengarang, Th_terbit
FROM tPengarang INNER JOIN tInfo_buku ON tPengarang.Kode_Pengarang =
tInfo_buku.Kode_Pengarang
Where Th_terbit>=all(Select Th_terbit From tInfo_buku);

```

#### Output :

	ISBN	JUDUL	NAMA_PENGARANG	TH_TERBIT
▶	11-111-227	E-Commerce	Yahya Kurniawan	2004

## 10. Menampilkan Pengarang yang mengarang lebih dari satu buku

### Sintaks SQL :

```
/* membuat view vPengarang */

Create View vPengarang (Kode_Pengarang, Jumlah) As
Select Kode_Pengarang, Count(*) As Jumlah
From tInfo_buku Group by Kode_Pengarang;

/* mencari Pengarang yang mengarang lebih dari satu buku dengan memanfaatkan
view vPengarang (Sintak_11.sql)*/

Select Kode_Pengarang, Nama_Pengarang
From vPengarang, tPengarang
Where vPengarang.Kode_Pengarang=tPengarang.Kode_Pengarang
And vPengarang.Jumlah>1 ;
```

### Output:

	KODE_PENGARANG	NAMA_PENGARANG
▶	1	Antony Pranata
	5	Alusius Yoga Phd

## 11. Menampilkan Informasi Buku, pengarang, penerbit, Tahun Terbit dan Kategori Buku yang dipinjam oleh Anggota tertentu.

### Sintaks SQL : (Sintak\_12.sql)

```
Select tAnggota.no_Anggota, tAnggota>Nama, tInfo_Buku.Judul,
tPengarang>Nama_Pengarang, tPenerbit>Nama_Penerbit,
tInfo_Buku.Th_Terbit, tJenis_Buku.Jenis_Buku
From tAnggota, tInfo_Buku, tPengarang, tPenerbit, tJenis_Buku,
tDaftar_Buku, tSirkulasi
Where tAnggota.no_Anggota=tSirkulasi.no_Anggota
And tSirkulasi.Kode_Buku=tDaftar_Buku.Kode_Buku
And tJenis_buku.DCC=tInfo_Buku.DCC
And tInfo_Buku.ISBN=tDaftar_Buku.ISBN
And tInfo_Buku.Kode_Pengarang=tPengarang.Kode_Pengarang
And tInfo_Buku.Kode_Penerbit=tPenerbit.Kode_Penerbit
And tAnggota.no_Anggota='1';
```

### Output :

	NO_ANGGOTA	NAMA	JUDUL	NAMA_PENGARANG	NAMA_PENERBIT	TH_TERBIT	JENIS_BUKU
▶	1	Riyanto	Visual Basic	Antony Pranata	Elek Media Komputindo	2000	Komputer
	1	Riyanto	Database Interbase	Onno W Purbo	Andi Offset	1999	Komputer

## 12. Menampilkan Daftar Jenis (Kategori) Buku yang belum ada

### Sintaks SQL : (Sintak\_13.sql)

```
Select DCC, Jenis_Buku
FROM tJenis_Buku
where not DCC in(select DCC From tInfo_Buku);
```

**Output :**

	DCC	JENIS_BUKU
►	4	Masakan

### 13. Menampilkan Daftar Anggota Dari Masing-Masing Program Studi Tertentu

```

Create View vIlKom (IlKom) as
Select Nama As IlKom From Tanggota Where Kode_Prodi='1';

Create View vKoms (Koms) as
Select Nama As Koms From Tanggota Where Kode_Prodi='2';

Create View vGeoFisika (GeoFisika) as
Select Nama As GeoFisika From Tanggota Where Kode_Prodi='3';

Create View vKimia (Kimia) As
Select Nama As Kimia From Tanggota Where Kode_Prodi='4';

```

Contoh 1:

**Sintaks SQL :** (Sintak\_14.sql)

```

select Distinct *
from (vKoms full outer join vIlkom on vKoms.Koms=vIlkom.Ilkom);

```

**Output :**

KOMS	ILKOM
► Kartiningrum	<null>
<null>	Riyanto
<null>	Sitepu

Contoh 2:

**Sintaks SQL :** (Sintak\_15.sql)

```

select *
from (vKoms full outer join vIlkom on vKoms.Koms=vIlkom.Ilkom);

```

**Output :**

KOMS	ILKOM
► <null>	Riyanto
<null>	Sitepu
Kartiningrum	<null>

Pemberian Contoh 1 dan 2 diatas dimaksudkan untuk membedakan antara disertakannya perintah “Distinct” dengan tanpa “Distinct”.

Dari kedua output diatas dapat disimpulkan bahwa pemberian perintah “Distinct” menyebabkan kolom sebelah kiri menjadi prioritas, dengan lain kata data yang ada pada kolom kiri ditampilkan terlebih dahulu.

Contoh 3:

**Sintaks SQL :** (Sintak\_16.sql)

```

select Distinct *

```

```
from (vGeofisika full outer join vKimia on
vGeofisika.Geofisika=vKimia.Kimia);
```

#### Output :

GEOFISIKA	KIMIA
► Dimas Chandra	<null>
<null>	Anang Susilo
<null>	Dian Kridawati
<null>	Dimas Mukhlas

#### 14. Menampilkan Daftar Anggota Dari Penggabungan Masing-Masing Program Studi Tertentu yang Telah dicari. (Penerapan Penggabungan View Dalam View).

```
Create View Komsilkom (Komsil, Ilkom) As
select Distinct *
from (vKomsil full outer join vIlkom on vKomsil.Komsil=vIlkom.Ilkom);

create view Geo_Kimia (Geofisika, Kimia) As
select Distinct *
from (vGeofisika full outer join vKimia on
vGeofisika.Geofisika=vKimia.Kimia);
```

#### Sintaks SQL : (Sintak\_17.sql)

```
Select Distinct * from Komsilkom full outer join Geo_Kimia
on Komsilkom.Komsil=Geo_Kimia.Kimia;
```

#### Output :

KOMSI	ILKOM	GEOFISIKA	KIMIA
► Kartiningrum	<null>	<null>	<null>
<null>	Riyanto	<null>	<null>
<null>	Sitepu	<null>	<null>
<null>	<null>	Dimas Chandra	<null>
<null>	<null>	<null>	Anang Susilo
<null>	<null>	<null>	Dian Kridawati
<null>	<null>	<null>	Dimas Mukhlas

### Kesimpulan

Berdasarkan materi yang disajikan dapat ditarik kesimpulan sebagai berikut:

- Sebagaimana di Microsoft Access yang mendukung *query*, demikian juga Interbase yang mendukung *view*.
- Interbase juga mendukung *sub-view* (*view* di dalam *view*) yang memungkinkan kita mencari informasi kompleks.
- Interbase mendukung lebih dari satu perintah SQL, yaitu dengan dibatasi tanda titik koma (;).
- Kelebihan lain, kapasitas memori yang diperlukan Interbase relatif kecil dan fitur visual yang *user friendly*.
- Merupakan RDMS yang *powerfull*, karena mampu melakukan pemrosesan transaksi dengan cepat dan menggunakan data secara bersama baik dalam lingkungan *singe user* ataupun *multi user*.



## Daftar Pustaka

Manual Interbase Server 6.5

## Biografi Penulis



**Riyanto**, Lahir di Pati, 20 Juli 1982. Menyelesaikan S1 Ilmu Komputer UGM, Yogyakarta pada tahun 2007. Saat ini mengelola *software house* yang bergerak di bidang Sistem Informasi dan *Networking*. Beberapa karyanya telah diterbitkan sebagai buku, diantaranya:

- Pengembangan Aplikasi Manajemen *Database* dengan Java 2 (SE/ME/EE). Penerbit Gava Media: 2008.
- Pengembangan Aplikasi Sistem Informasi Geografis Berbasis *Desktop* dan *Web*. Penerbit Gava Media: 2009.
- Membuat Sendiri Sistem Informasi Penjualan dengan PHP

dan MySQL (Studi Kasus Aplikasi Mini Market Integrasi *Barcode Reader*). Penerbit Gava Media: 2010.

- Sistem Informasi Geografis Berbasis *Mobile*. Penerbit Gava Media: 2010.
- Membuat Sendiri Aplikasi *Mobile GIS* Platform Java ME, BlackBerry, dan Android. (Proses Cetak). Penerbit Andi Offset: 2010.

Penulis juga aktif sebagai *IT Trainer* di beberapa lembaga pelatihan di Yogyakarta. Informasi lebih lanjut tentang penulis dapat Anda temukan di <http://www.masto.co.cc>.