Modul 5

Manipulasi dan Retrieve Data(Bagian II)

Tujuan :

1. Mencari dan menampilkan data dengan perintah select
2. Mengkombinasikan perintah select dengan perintah lainnya

Dasar teori

a. Memberikan nama lain pada kolom

```
SELECT namakolomlama AS namakolombaru FROM namatabel;
```

Berikut ini perintah untuk memberikan nama lain pada kolom jenis menjadi jenis_film pada tabel jenisfilm :

```
SELECT JENIS AS TYPE FROM JENISFILM;
```

b. Menggunakan alias untuk nama tabel

```
SELECT namalias.jenis, namalias.harga FROM namatabel namalias;
```

Berikut ini perintah untuk memberikan alias pada tabel jenisfilm :

```
SELECT J.JENIS, J.HARGA FROM JENISFILM J;
```

c. Menampilkan data lebih dari dua tabel

```
SELECT * FROM namatabel1, namatabel2, namatabel-n;
```

Sebagai contoh buat table baru berikut :

Kd_Supplier	NamaSupplier	AlamatSupplier	TelponSupplier
012228	CV.Super Makmur	JL.Kemakmuran No. 12	0541-7009673
021123	Cv.Makmur indah	JL.rotan No.201	0541-290150
021200	Cv.indah maksud	JL.berkat No.202	0541-292150
022323	Cv.Cyber Sequad	JL.Indah Permata No.	0541-290189
022345	Cv.Cita bakal	JL.Permata No.01	0541-290190
023000	Cv.gokil maju	JL.kurna No.132	0541-291113
023111	Cv.Uncle Grup	JL.P.Suryanata No.11	0541-290123
023436	Cv.Maju Mundur	JL.soepratman No.101	0541-290367

d. Nested Queries / Subquery (IN, NOT IN, EXISTS, NOT EXISTS

Subquery berarti query di dalam query. Dengan menggunakan subquery, hasil dari query akan menjadi bagian dari query di atasnya. Subquery terletak di dalam klausa WHERE atau HAVING. Pada klausa WHERE,subquery digunakan untuk memilih baris-baris tertentu yang kemudian digunakan oleh query.

Sedangkan pada klausa HAVING, subquery digunakan untuk memilih kelompok baris yang kemudian digunakan oleh query.

Contoh 1: perintah untuk menampilkan data pada tabel jenisfilm yang mana data pada kolom jenis-nya tercantum pada tabel film menggunakan IN :

```
SELECT * FROM JENISFILM WHERE JENIS IN (SELECT JENIS FROM FILM);
```

atau menggunakan EXISTS

```
SELECT * FROM JENISFILM WHERE EXISTS (SELECT * FROM FILM WHERE
HARGA > 2000);
```

Pada contoh di atas : `SELECT JENIS FROM FILM` disebut subquery, sedangkan: SELECT * FROM JENISFILM berkedudukan sebagai query. Perhatikan,terdapat data jenis dan harga pada tabel jenisfilm yang tidak ditampilkan. Hal ini disebabkan data pada kolom jenis tidak terdapat pada kolom jenis di tabel film.

Contoh 2: perintah untuk menampilkan data pada tabel jenisfilm yang mana data pada kolom jenis-nya tidak tercantum pada tabel film menggunakan NOT IN :

```
SELECT * FROM JENISFILM WHERE JENIS NOT IN (SELECT JENIS FROM
FILM);
```

atau menggunakan NOT EXISTS

```
SELECT * FROM JENISFILM WHERE NOT EXISTS (SELECT * FROM FILM
WHERE HARGA > 2000);
```

e. Operator comparison ANY dan ALL

Operator ANY digunakan berkaitan dengan subquery. Operator ini menghasilkan TRUE (benar) jika paling tidak salah satu perbandingan dengan hasil subquery menghasilkan nilai TRUE. Ilustrasinya jika: Gaji > ANY (S)

Jika subquery S menghasilkan G1, G2,..,Gn, maka kondisi di atas identik dengan:

(gaji > G1) OR (gaji > G2) OR ... OR (gaji > Gn)

Contoh : perintah untuk menampilkan semua data jenisfilm yang harganya bukan yang terkecil:

```
SELECT * FROM JENISFILM WHERE HARGA > ANY (SELECT HARGA FROM JENISFILM);
```

Operator ALL digunakan untuk melakukan perbandingan dengan subquery. Kondisi dengan ALL menghasilkan nilai TRUE (benar) jika subquery tidak menghasilkan apapun atau jika perbandingan menghasilkan TRUE untuk setiap nilai query terhadap hasil subquery.

Contoh : perintah untuk menampilkan data jenisfilm yang harganya paling tinggi:

```
SELECT * FROM JENISFILM WHERE HARGA >= ALL (SELECT HARGA FROM JENISFILM);
```

f. Sintak ORDER BY

Klausa ORDER BY digunakan untuk mengurutkan data berdasarkan kolom tertentu sesuai dengan tipe data yang dimiliki. Contoh : perintah untuk mengurutkan data film berdasarkan kolom judul:

```
SELECT * FROM FILM ORDER BY JUDUL;
```

atau tambahkan ASC untuk pengurutan secara ascending (menaik) :

```
SELECT * FROM FILM ORDER BY JUDUL ASC;
```

atau tambahkan DESC untuk pengurutan secara descending (menurun) :

```
SELECT * FROM FILM ORDER BY JUDUL DESC;
```

g. Sintak DISTINCT

Distinct adalah kata kunci ini untuk menghilangan duplikasi. Sebagai Contoh, buat sebuah tabel pelanggan yang berisi nama dan kota asal dengan beberapa record isi dan beberapa kota asal yang sama. Kemudian ketikkan perintah berikut:

```
SELECT DISTINCT KOTA FROM PELANGGAN;
```

Dengan perintah di atas maka nama kota yang sama hanya akan ditampilkan satu saja.

h. UNION, INTERSECT dan EXCEPT

UNION merupakan operator yang digunakan untuk menggabungkan hasil query, dengan ketentuan jumlah, nama dan tipe kolom dari masing-masing tabel yang akan ditampilkan datanya harus sama. Berikut ini perintah untuk memperoleh data pada tabel film dimana jenisnya action dan horor:

```
SELECT JENIS, JUDUL FROM FILM WHERE JENIS = 'ACTION' UNION SELECT
JENIS, JUDUL FROM FILM WHERE JENIS = 'HOROR';
```

Perintah di atas identik dengan :

```
SELECT JENIS, JUDUL FROM FILM WHERE JENIS = 'ACTION' OR JENIS ='HOROR';
```

Namun tidak semua penggabungan dapat dilakukan dengan OR, yaitu jika bekerja pada dua tabel atau lebih.

INTERSECT merupakan operator yang digunakan untuk memperoleh data dari dua buah query dimana data yang ditampilkan adalah yang memenuhi kedua query tersebut dengan ketentuan jumlah, nama dan tipe kolom dari masing-masing tabel yang akan ditampilkan datanya harus sama.

```
SELECT * FROM namatabel1 INTERSECT SELECT * FROM namatabel2;
```

Pada MySQL tidak terdapat operator INTERSECT namun sebagai gantinya dapat menggunakan operator IN seperti contoh 1 pada bagian Nested Queries.

EXCEPT / Set Difference merupakan operator yang digunakan untuk memperoleh data dari dua buah query dimana data yang ditampilkan adalah data yang ada pada hasil query 1 dan tidak terdapat pada data dari hasil query 2 dengan ketentuan jumlah, nama dan tipe kolom dari masing-masing tabel yang akan ditampilkan datanya harus sama.

```
SELECT * FROM namatabel1 EXCEPT SELECT * FROM namatabel2;
```

Pada MySQL tidak terdapat operator EXCEPT namun sebagai gantinya dapat menggunakan operator NOT IN seperti contoh 2 pada bagian Nested Queries.

Praktik

1. Buat tabel Barang sebagai berikut :

Field	Type	Null	Key	Default	Extra
KodeBarang	varchar(8)	NO	PRI		
NamaBarang	varchar(20)	NO			
MerkBarang	varchar(20)	NO			
StokBarang	int(5)	YES		NULL	
TglBeli	date	YES		NULL	
HargaBeli	int(12)	YES		NULL	
HargaJual	int(12)	YES		NULL	

2. Isi data tabel Seperti berikut dan tampilkan semua kolom !

```
mysql> select*from barang;
+------------+---------------+------------+------------+------------+-----------+-----------+
| KodeBarang | NamaBarang    | MerkBarang | StokBarang | TglBeli    | HargaBeli | HargaJual |
+------------+---------------+------------+------------+------------+-----------+-----------+
| A01        | Monitor       | Samsung    |          5 | 2006-04-04 |    450000 |    550000 |
| A02        | Monitor       | LG         |          6 | 2006-04-04 |    500000 |    600000 |
| A03        | Monitor       | Toshiba    |          6 | 2006-02-22 |    475000 |    575000 |
| B01        | Charger       | Toshiba    |         12 | 2006-02-22 |     90000 |    115000 |
| B02        | Charger       | Acer       |         12 | 2005-02-20 |    100000 |    135000 |
| B03        | Charger       | Axio       |          6 | 2004-05-20 |     80000 |    110000 |
| C01        | Flashdisk 4Gb | Toshiba    |          4 | 2006-04-22 |     40000 |     60000 |
| C02        | Flashdisk 4Gb | Kingstone  |          8 | 2008-02-05 |     35000 |     50000 |
| D01        | Printer       | Samsung    |          3 | 2007-07-07 |    900000 |   1200000 |
+------------+---------------+------------+------------+------------+-----------+-----------+
```

3. Tampilkan kolom KodeBarang, NamaBarang dan MerkBarang yang Kondisi StokBarang nya 6!
4. Tampilkan kolom NamaBarang, MerkBarang, HargaBeli, HargaJual dan sebuah kolom baru yaitu HargaJual-HargaBeli yang berisi Keuntungan dari HargaJual di kurang HargaBeli !
5. Ubah HargaJual menjadi NULL untuk KodeBarang = D01.Kemudian lakukan kembali percobaan 5.
6. Seperti percobaan 4,tampilkan kolom NamaBarang, MerkBarang, HargaBeli,HargaJual dan sebuah kolom baru (gunakan alias) yaitu Keuntungan yang terdiri dari HargaBeli dan HargaJual!
7. Tambahkan record baru dengan value : E01,LCD, LG, 2, 2005-09-01, 1000000, NULL dan Tampilkan kolom yang telah di tambahkan saja dengan kondisi KodeBarang E01!
8. Untuk KodeBarang yang ber-kode A03 dan B02 ubah Merkbarang menjadi Asus !
9. Sekarang tampilkan kolom MerkBarang saja !
10. .Menampilkan MerkBarang saja tapi datanya tidak ada yang sama !
11. Tampilkan Harga Jual Barang antara 550.000 – 100.000 !
12. Tampilkan tabel barang yang terurut berdasarkan MerkBarang (dari a ke z)!
13. Tampilkan tabel barang yang diurutkan berdasarkan MerkBarang dengan urutan terbalik (dari z ke a)!