

BAB V

PENGENALAN OPERATOR DASAR

4.1 Materi

4.1.1 AS

AS biasa digunakan untuk menampilkan label kolom dengan nama lain sehingga yang akan muncul dalam hasil query bukan nama asli kolom, tetapi nama yang mungkin lebih sesuai dan mudah dimengerti. AS digunakan setelah nama kolom yang akan diganti yang kemudian diikuti dengan nama penggantinya. Berikut struktur querynya :

select namakolom as namakolompengganti from namatabel;

Contoh :

```
MySQL [mariadbml]> select * from baranghapus;
+-----+-----+
| id_bar | nama_bar |
+-----+-----+
| 10    | kotakPensilBiru |
| 11    | BukuGambar      |
| 12    | Penggaris       |
+-----+-----+
3 rows in set (0.00 sec)

MySQL [mariadbml]> select id_bar as kode_brg, nama_bar as nama_brg from baranghapus;;
+-----+-----+
| kode_brg | nama_brg |
+-----+-----+
| 10      | kotakPensilBiru |
| 11      | BukuGambar      |
| 12      | Penggaris       |
+-----+-----+
3 rows in set (0.00 sec)
```

4.1.2 AND dan OR

Pada bagian sebelumnya kita menggunakan anak kalimat atau sintaks *WHERE* hanya pada konteks yang sederhana. Berikut ini kita akan mencoba menggunakan *WHERE* untuk konteks yang lebih kompleks lagi, anak kalimat *where* yang kompleks akan bekerja dengan baik dengan menggunakan kata *AND* dan *OR*. SQL menggunakan standar logika *boolean three-valued* seperti pada tabel berikut ;

Kondisi A	Kondisi B	A AND B	A OR B
True	True	True	True
True	False	False	True
True	Null	Null	True
False	False	False	False
False	Null	False	Null
Null	Null	Null	Null

A	NOT A
True	False
False	True
Null	Null

Berikut struktur SQL untuk penampilan data :

select * from namatabel where namakolom = 'pencarian 1' and namakolom = 'pencarian 2';

Atau

select * from namatabel where namakolom = 'pencarian 1' or namakolom = 'pencarian 2';

Contoh :

```
MySQL [praktikumdb]> select*from barang;
+-----+-----+-----+
| id_bar | nama_bar | id_mac |
+-----+-----+-----+
| 1 | Pensil | 10001 |
| 2 | Penghapus | 10002 |
| 3 | bolpoin | 10003 |
| 4 | Buku Kotak | 10004 |
| 5 | Buku Garis | 10005 |
+-----+-----+-----+
5 rows in set (0.36 sec)

MySQL [praktikumdb]> select * from barang where id_bar=1 AND nama_bar='pensil';
+-----+-----+-----+
| id_bar | nama_bar | id_mac |
+-----+-----+-----+
| 1 | Pensil | 10001 |
+-----+-----+-----+
1 row in set (0.11 sec)

MySQL [praktikumdb]> select * from barang where id_bar=1 OR nama_bar='bolpoin';
+-----+-----+-----+
| id_bar | nama_bar | id_mac |
+-----+-----+-----+
| 1 | Pensil | 10001 |
| 3 | bolpoin | 10003 |
+-----+-----+-----+
2 rows in set (0.04 sec)
```

4.1.3 BETWEEN

BETWEEN digunakan untuk menentukan lebar nilai dari nilai terendah dan nilai tertinggi. Pada *BETWEEN* menggunakan operator pembandingan seperti pada tabel berikut;

Operator	Penjelasan
<	Kurang dari
>	Lebih dari
<=	Kurang dari atau sama dengan
>=	Lebih dari atau sama dengan
=	Sama dengan
< > atau !=	Tidak sama dengan

Logika operator *BETWEEN* sebagai berikut :

- ◆ Nilai a dalam formula "a **BETWEEN** x **AND** y" indentik dengan "a >= x **AND** a <= y"

- ◆ "a NOT BETWEEN x AND y" identik dengan "a < x OR a > y"

Berikut Struktur yang digunakan :

select * from nama_tabel where nama_kolom between 'nilai_awal' and 'nilai_akhir';

4.1.4 IN dan NOT IN

Operator IN berguna melakukan pencocokan dengan salah satu yang ada pada suatu daftar nilai.

Berikut Struktur yang digunakan

Select * from nama_tabel where nama_kolom in (kata_kunci1, kata_kunci2, kata_kunci3, kata_kunci4);

Contoh : menampilkan id macam barang yang memiliki id 10001 dan 10002

```
MySQL [praktikumDB]> select * from macam_barang where id_mac in(10001,10002);
+-----+-----+-----+
id_mac | nama_mac | guna_mac |
+-----+-----+-----+
10001 | Alat Tulis | menulis |
10002 | Penghapus | menulis |
+-----+-----+-----+
rows in set (0.00 sec)
```

4.2.5 LIKE

LIKE digunakan ketika kita ingin mencari sebuah data yang hanya diwakili oleh salah satu atau lebih hurufnya saja. Misalkan kita ingin mencari nama yang huruf awalnya A, maka kita harus menggunakan LIKE.

Berikut adalah tabel contoh kasus penggunaan LIKE :

Kasus	Operator
Diawali dengan huruf A	LIKE 'A%'
Diakhiri dengan huruf A	LIKE '%A'
Huruf A pada posisi ke dua	LIKE '_A%'
Diawali dengan huruf A dan mengandung huruf l	LIKE 'A%l%'
Tidak diawali dengan huruf A	NOT LIKE 'A%'

Berikut struktur SQL untuk pencarian data menggunakan LIKE :

select * from nama_tabel where nama_kolom like 'operator';

Contoh : Mencari data barang yang mempunyai nama dengan huruf akhiran s

```

MySQL [praktikumDB]> select * from barang;
+-----+-----+-----+
| id_bar | nama_bar | id_mac |
+-----+-----+-----+
| 1 | Pensil | 10001 |
| 2 | Penghapus | 10002 |
| 3 | bolpoin | 10003 |
| 4 | Buku Kotak | 10004 |
| 5 | Buku Garis | 10005 |
+-----+-----+-----+
5 rows in set (0.00 sec)

MySQL [praktikumDB]> select * from barang where nama_bar like '%s';
+-----+-----+-----+
| id_bar | nama_bar | id_mac |
+-----+-----+-----+
| 2 | Penghapus | 10002 |
| 5 | Buku Garis | 10005 |
+-----+-----+-----+
2 rows in set (0.00 sec)

MySQL [praktikumDB]>

```

4.2.6 DISTINCT

DISTINCT sering kali diperlukan untuk mengembalikan hasil dari sebuah *query* dengan tidak terdapat duplikasi, artinya pada hasil outputnya tidak terjadi kesamaan data meskipun pada data sesungguhnya sangat mungkin banyak duplikasi. Hal ini juga berlaku jika data yang ada dalam bentuk angka. Berikut struktur perintah dasar SQL :

select distinct nama_kolom from nama_tabel;

Contoh :

```

for the right syntax to use near 'from macam_barang' at line 1
MySQL [praktikumDB]> select * from macam_barang;
+-----+-----+-----+
| id_mac | nama_mac | guna_mac |
+-----+-----+-----+
| 10001 | Alat Tulis | menulis |
| 10002 | Penghapus | menulis |
| 10003 | Alat Tulis | menulis |
| 10004 | Buku | menulis |
| 10005 | Buku | menulis |
+-----+-----+-----+
5 rows in set (0.00 sec)

MySQL [praktikumDB]> select distinct nama_mac from macam_barang;
+-----+
| nama_mac |
+-----+
| Alat Tulis |
| Penghapus |
| Buku |
+-----+
3 rows in set (0.04 sec)

MySQL [praktikumDB]>

```

4.1.7 LIMIT

LIMIT digunakan untuk membatasi jumlah output dari query berdasarkan jumlah row bukan karena kondisi seperti WHERE. Bisa dikatakan LIMIT adalah untuk menentukan jumlah baris yang akan ditampilkan yang dihitung dari baris pertama, Berikut struktur perintah dasar SQL :

select * from nama_tabel limit 2;

Contoh :

```

MySQL [praktikumDB]> select * from macam_barang;
+-----+-----+-----+
| id_mac | nama_mac | guna_mac |
+-----+-----+-----+
| 10001  | Alat Tulis | menulis  |
| 10002  | Penghapus | menulis  |
| 10003  | Alat Tulis | menulis  |
| 10004  | Buku      | menulis  |
| 10005  | Buku      | menulis  |
+-----+-----+-----+
5 rows in set (0.00 sec)

MySQL [praktikumDB]> select * from macam_barang limit 2;
+-----+-----+-----+
| id_mac | nama_mac | guna_mac |
+-----+-----+-----+
| 10001  | Alat Tulis | menulis  |
| 10002  | Penghapus | menulis  |
+-----+-----+-----+
2 rows in set (0.00 sec)

```

4.1.8 UNION, EXCEPT dan INTERSECT

Hasil dari dua buah query dapat dikombinasikan dengan menggunakan UNION, EXCEPT atau INTERSECT. UNION digunakan untuk menggabungkan hasil dua buah query menjadi satu kolom. Berikut struktur query untuk UNION :

select nama_kolom from nama_tabel union select nama_kolom from nama_tabel;

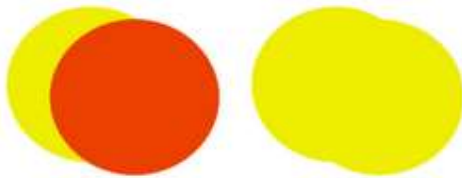
atau

select nama_kolom from nama_tabel union all select nama_kolom from nama_tabel;

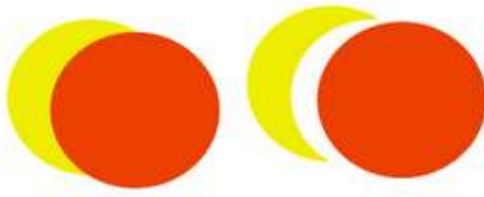
EXCEPT digunakan untuk menampilkan hanya query pertama saja, sedangkan hasil query kedua dan yang sama dengan hasil query kedua tidak akan ditampilkan. Berikut struktur query penggunaan EXCEPT :

Perintah INTERSECT hanya akan menampilkan seluruh isi dari data yang memiliki kesamaan diantara hasil kedua query tersebut. Berikut struktur query penggunaan INTERSECT :

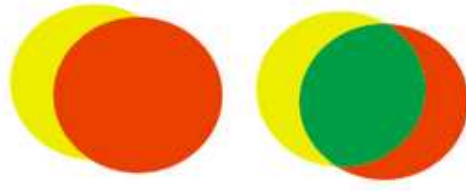
Dari uraian materi tentang UNION, EXCEPT dan INTERSECT dapat digambarkan sebagai berikut :



Gambar diatas menyatakan UNION yaitu gabungan dari kedua buah object.



Sedangkan gambar kedua menyatakan EXCEPT dimana object yang diambil hanya yang kuning. Sehingga object yang merah atau object kuning yang sama dengan object merah tidak diambil.



Gambar ke tiga menyatakan INTERSECT yaitu mengambil object yang berwarna hijau atau object memiliki kandungan dari object kuning dan merah.

Contoh :

```
MySQL [praktikumDB]> select * from mk_bio;
+-----+
| nama_mk |
+-----+
| genetika |
| anatomi |
| komputer |
+-----+
3 rows in set (0.00 sec)

MySQL [praktikumDB]> select * from mk_informatika;
+-----+
| nama_mk |
+-----+
| programming |
| logika |
| komputer |
+-----+
3 rows in set (0.00 sec)

MySQL [praktikumDB]> select * from mk_bio union select * from mk_informatika;
+-----+
| nama_mk |
+-----+
| genetika |
| anatomi |
| komputer |
| programming |
| logika |
+-----+
5 rows in set (0.00 sec)
```

Tugas Praktikum

Persiapan praktek : Gunakan tabel mahasiswa dan fakultas pada pertemuan sebelumnya. Tambahkan kolom gender di tabel mahasiswa kemudian update data mahasiswa. Insert data mahasiswa hingga 7 baris.

1. Berdasarkan tabel sebelumnya yaitu bab 4, tampilkan data mahasiswa dengan nama dari kolomnya berubah menjadi nomor_identitas, nama, alamat, nama_fakultas tanpa harus merubah nama kolom secara permanen.
2. Tampilkan nama mahasiswa fakultas saintek yang berjenis kelamin perempuan. Kemudian nama mahasiswa fakultas saintek atau berjenis kelamin perempuan
3. Pada tabel mahasiswa terdapat 7 baris data mahasiswa. Tampilkan data mahasiswa dari nomor 3 sampai dengan 5, kemudian tampilkan juga data mahasiswa yang bukan dari nomor 3 sampai dengan 5 menggunakan between dan tanpa menggunakan between.
4. Tampilkan mahasiswa dengan nomor 2,3,5 menggunakan operator IN.
5. Tampilkan semua data mahasiswa yang namanya mempunyai awalan huruf "i".

6. Tampilkan semua data mahasiswa yang namanya mempunyai akhiran "a" dan huruf ke-3 dari nama tersebut terdapat huruf "n".
7. Tampilkan nomor fakultas di tabel mahasiswa dengan nomor harus tidak ada yang sama dari baris pertama sampai akhir.
8. Tampilkan data mahasiswa 3 baris saja.
9. Buatlah tabel dengan nama organ_dalam dan organ_luar sebagai berikut :
 - a. Tabel organ_dalam

Nomor	Nama
1	Jantung
2	Paru-paru
3	Tenggorokan

- b. Tabel organ_luar

Nomor	Nama
1	Hidung
2	Mata
3	Tenggorokan

Tampilkan data dari kedua tabel tersebut menggunakan UNION, EXCEPT dan INTERSECT

Next week : SUB QUERY dan INDEKS