

Pertemuan 3

Pengambilan data dan Operator dalam Query

Dalam menuliskan perintah query, penggunaan operator sangat penting peranannya. Operator-operator ini digunakan untuk berbagai alasan, termasuk untuk membatasi query yang ditampilkan dan juga sebagai syarat dalam menggabungkan beberapa tabel.

Adapun beberapa jenis operator yang digunakan dalam query adalah:

1. Operator Aritmetika

Merupakan operator yang digunakan untuk operasi matematika, yakni: *, /, +, dan -. Operator ini dapat digunakan terhadap field yang akan dikalkulasi nilainya.

2. Operator Perbandingan

Merupakan operator yang umumnya digunakan dalam matematika untuk menentukan besar-kecil sebuah nilai.

Yang termasuk dalam operator ini adalah:

Operator	Arti
=	Sama dengan
>	Lebih besar dari
>=	Lebih besar atau sama dengan
<	Kurang dari
<=	Kurang dari atau sama dengan
<>	Tidak sama dengan
!=	Tidak sama dengan
^=	Tidak sama dengan
BETWEEN ... AND ...	Suatu kondisi yang berada di antara 2 value
IN(set nilai)	Yang cocok dengan salah satu yang terdapat dalam set nilai
LIKE	Yang cocok dengan pola karakter tertentu
IS NULL	Jika value-nya merupakan merupakan nilai null

Catatan:

- BETWEEN ... AND ... digunakan untuk menampilkan baris berdasarkan *range* sebuah nilai yang terletak di samping kiri-kanan kata AND. Jika tidak memenuhi yang berada di dalam *range* dimaksud, dapat menggunakan NOT BETWEEN ... AND ... Value yang menjadi nilai dari query ini wajib dalam bentuk numerik
- IN (set) digunakan untuk menunjukkan satu kumpulan himpunan yang memenuhi. Jika tidak memenuhi yang berada dalam himpunan tersebut, digunakan NOT IN (set nilai). Value di dalam set nilai dapat berupa nilai numerik maupun string.
- LIKE merupakan sebuah kondisi untuk melakukan pencarian wildcard terhadap sebuah string dimana:
 - Tanda % mewakili banyak karakter
 - Tanda _ mewakili satu karakter
- IS NULL digunakan untuk mencari data yang tidak ada isinya (kosong). Jika mencari yang isinya tidak kosong, dapat digunakan klausa IS NOT NULL.

3. Operator Logika

Operator ini merupakan operator penting yang terutama digunakan ketika pengguna basis data memiliki lebih dari satu syarat, atau juga dapat digunakan pada saat melakukan join beberapa tabel.

Operator	Arti
AND	Menghasilkan TRUE apabila kedua komponen benar
OR	Menghasilkan TRUE apabila salah satu komponen benar
NOT	Menghasilkan TRUE apabila kondisinya false

Hak Tata Urutan Operator

Dalam mengerjakan sebuah query, sebuah operator memiliki tata urutan pengerjaan, dimana sebuah operator yang posisinya lebih tinggi akan dikerjakan terlebih dahulu dari operator yang ada di bawahnya. Adapun urutannya dapat dilihat pada tabel berikut.

Urutan Pengerjaan	Operator
1	Operator aritmetika
2	Operator penggabungan
3	Operator perbandingan
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	Kondisi logika NOT
7	Kondisi logika AND
8	Kondisi logika OR

Tanda kurung (dan) dapat digunakan untuk memberikan hak pengerjaan lebih tinggi.

ALIAS

Alias merupakan penamaan ulang dari sebuah kolom ataupun tabel. Penulisan alias pada kolom/field dapat dengan menambahkan kata AS setelah nama field atau cukup dengan menambahkan sebuah spasi setelah field tersebut. Dalam sebuah field yang dibuatkan alias-nya, apabila alias tersebut menginginkan adanya perbedaan huruf besar-kecil atau spasi, alias tersebut dapat dituliskan dengan menggunakan tanda petik.

PRAKTIK

1. Tambahkan tabel matakuliah dalam basis data AKADEMIK yang telah Anda buat pada pertemuan sebelumnya sesuai dengan skema pada pertemuan ke-2.
2. Tambahkan data-data berikut ini ke dalam tabel matakuliah tersebut.

Kode_mkul	Nama_matakuliah	SKS	Semester	Status
TI01	SBD	3	2	W
TI02	Algoritma dan Pemrograman	3	1	W
SI01	SQA	2	3	P
SI02	Pengantar Bisnis	3	4	W
SI03	Teori Organisasi	2	4	P
SI04	Matematika Dasar	3	1	W
TK01	Dasar Teknik Listrik	3	1	W
TK02	Fisika	3	3	W

TK03	Administrasi Jaringan	2	5	P	
KA01	Akuntansi Pengantar	3	1	W	
KA02	Hukum Pajak	2	2	W	
KA03	Perpajakan	2	3	W	
MI01	Matematika Bisnis	2	2	P	
MI02	Pemrograman Berorientasi	2	2	W	
	Obyek 1				
MI03	Statistika Dasar	2	3	P	
+-----+-----+-----+-----+-----+					

3. Tampilkan data dosen yang jurusannya selain TI

```
SELECT * FROM dosen WHERE Jurusan <> 'TI';
atau
SELECT * FROM dosen WHERE not Jurusan = 'TI';
```

4. Tampilkan NIM, Nama_mhs dan Kota mahasiswa yang kotanya Klaten dan Solo

```
SELECT NIM, Nama_mhs, Kota FROM Mahasiswa WHERE Kota='Klaten' OR
Kota = 'Solo';
atau
SELECT NIM, Nama_mhs, Kota FROM Mahasiswa WHERE Kota
IN('Klaten', 'Solo');
```

5. Tampilkan NIM, Nama_mhs dan Kota mahasiswa yang kotanya selain Klaten dan Solo

```
SELECT NIM, Nama_mhs, Kota FROM Mahasiswa WHERE Kota NOT
IN('Klaten', 'Solo');
```

6. Tampilkan dosen yang tanggal lahirnya antara tanggal '1970-01-01' dan '1980-12-30'

```
SELECT * FROM Dosen WHERE Tgl_lahir>='1970-01-01' AND Tgl_lahir
<= '1980-12-30';
atau
SELECT * FROM Dosen WHERE Tgl_lahir BETWEEN '1970-01-01' AND
'1980-12-30';
```

7. Tampilkan nama dosen, Jurusan dan pendidikan terakhir yang mempunyai nama dengan huruf depannya 'S'

```
SELECT Nama_dosen, Jurusan, Pendidikan_terakhir FROM dosen WHERE
Nama_dosen LIKE 'S%';
```

8. Tampilkan nama mahasiswa untuk nama mahasiswa yang tidak mengandung unsur huruf "a"

```
SELECT Nama_mhs FROM mahasiswa WHERE Nama_mhs NOT LIKE '%a%';
```

9. Tampilkan nama mahasiswa yang pada bagian belakang namanya terdapat huruf 'a'

10. Tampilkan data jurusan yang nama_jurusan-nya mengandung kata 'informa'

11. Tampilkan semua kode dan mata kuliah yang mengandung kata 'Dasar' di dalamnya.

12. Tampilkan semua mata kuliah dengan status 'P'.

13. Tampilkan tempat lahir dosen dimana tempat lahir yang sama hanya ditampilkan satu kali saja. Ketikkan perintah di bawah ini dengan dan tanpa kata DISTINCT dan amati bedanya.

```
SELECT DISTINCT Tempat_lahir FROM dosen ORDER BY tempat_lahir;
```

14. Tampilkan tempat lahir dosen jurusan TI dimana tempat lahir yang sama hanya ditampilkan satu kali saja. Ketikkan perintah di bawah ini dengan dan tanpa kata **DISTINCT** dan amati bedanya.

```
SELECT DISTINCT nama_mhs, kota FROM mahasiswa WHERE  
Kota='Yogya';
```

15. Ketikkan query di bawah, amati hasilnya, jelaskan

```
SELECT Nama_matakuliah FROM matakuliah WHERE semester > ANY  
(SELECT semester FROM matakuliah WHERE Semester > 2);
```

Modifikasi perintah tersebut menjadi

```
SELECT Nama_matakuliah FROM matakuliah WHERE semester < ANY  
(SELECT semester FROM matakuliah WHERE Semester > 2);
```

Soal no 15 disebut dengan subquery. Subquery akan dibahas pada pertemuan lain.