

PERKULIAHAN KE 11

BAHASA QUERY TERAPAN LANJUTAN

1. *Join*

Menurut (Ladjamudin, 2004) menyampaikan bahwa Join merupakan operasi yang digunakan untuk menggabungkan dua tabel atau lebih dengan hasil berupa gabungan dari kolom-kolom yang berasal dari tabel-tabel tersebut. Contoh, diketahui ada dua tabel yaitu Tabel Mahasiswa dan Tabel Nilai, sebagai berikut :

Tabel Mahasiswa			Tabel Nilai			
NIM	NAMA	ALAMAT	NIM	KD_MK	MID	FINAL
10296832	Nurhayati	Jakarta	10296832	KK021	60	75
10296126	Astuti	Jakarta	10296126	KD132	70	90
31296500	Budi	Depok	31296500	KK021	55	40
41296525	Prananigrum	Bogor	41296525	KU122	90	80
50096487	Pipit	Bekasi	21196353	KU122	75	75
21196353	Quraish	Bogor	50095487	KD132	80	0
10296001	Fintri	Depok				
21198002	Julizar	Jakarta				

Ada beberapa tipe Join, yaitu :

a. INNER JOIN

Menggabungkan dua tabel dimana diantara dua tabel datanya bersesuaian.

Jika ada Query sebagai berikut :

Nilai INNER JOIN Mahasiswa ON Nilai.NIM = Mahasiswa.NIM;

Maka Hasil gabungannya adalah :

NIM	KD_MK	MID	FINAL	NIM	NAMA	ALAMAT
10296832	KK021	60	75	10296832	Nurhayati	Jakarta
10296126	KD132	70	90	10296126	Astuti	Jakarta
31296500	KK021	55	40	31296500	Budi	Depok
41296525	KU122	90	80	41296525	Prananigrum	Bogor
21196353	KU122	75	75	21196353	Quraish	Bogor
50096487	KD132	80	0	50096487	Pipit	Bekasi

Kemudian jika diberikan query sebagai berikut:

```
SELECT Nilai.NIM, Mahasiswa.NAMA, Nilai.KD_MK, Nilai.MID
FROM Nilai INNER JOIN Mahasiswa ON Nilai.NIM = Mahasiswa.NIM;
```

Maka Hasilnya adalah :

NIM	NAMA	KD_MK	MID
10296832	Nurhayati	KK021	60
10296126	Astuti	KD132	70
31296500	Budi	KK021	55
41296525	Prananigrum	KU122	90
21196353	Quraish	KU122	75
50095487	Pipit	KD132	80

b. LEFT JOIN atau LEFT OUTER JOIN

Menggabungkan dua tabel dimana diantara dua tabel datanya bersesuaian dan juga semua record pada tabel sebelah kiri.

Contoh : Jika ada query sebagai berikut :

```
Mahasiswa LEFT OUTER JOIN Nilai ON Nilai.NIM = Mahasiswa.NIM
```

Maka Hasil gabungannya adalah sebagai berikut :

NIM	NAMA	ALAMAT	NIM	KD_MK	MID	FINAL
10296832	Nurhayati	Jakarta	10296832	KK021	60	75
10296126	Astuti	Jakarta	10296126	KD132	70	90
31296500	Budi	Depok	31296500	KK021	55	40
41296525	Prananigrum	Bogor	41296525	KU122	90	80
50096487	Pipit	Bekasi	50096487	KD132	80	0
21196353	Quraish	Bogor	21196353	KU122	75	75
10296001	Fintri	Depok	NULL	NULL	NULL	NULL
21198002	Julizar	Jakarta	NULL	NULL	NULL	NULL

Kemudian jika diberikan query sebagai berikut :

```
SELECT Mahasiswa.NIM, Mahasiswa.NAMA, Nilai.KD_MK, Nilai.MID
FROM Mahasiswa LEFT OUTER JOIN Nilai ON Nilai.NIM =
Mahasiswa.NIM; Maka Hasilnya :
```

NIM	NAMA	KD_MK	MID
10296832	Nurhayati	KK021	60
10296126	Astuti	KD132	70
31296500	Budi	KK021	55
41296525	Prananigrum	KU122	90
50096487	Pipit	KD132	80
21196353	Quraish	KU122	75
10296001	Fintri	NULL	NULL
21198002	Julizar	NULL	NULL

c. RIGHT JOIN atau RIGHT OUTER JOIN

Menggabungkan dua tabel dimana diantara dua tabel datanya bersesuaian dan juga semua record pada tabel sebelah kanan.

Contoh, jika ada query seperti dibawah ini :

Nilai RIGHT OUTER JOIN Mahasiswa ON Nilai.NIM = Mahasiswa.NIM;

Maka Hasilnya :

NIM	KD_MK	MID	FINAL	NIM	NAMA	ALAMAT
10296832	KK021	60	75	10296832	Nurhayati	Jakarta
10296126	KD132	70	90	10296126	Astuti	Jakarta
31296500	KK021	55	40	31296500	Budi	Depok
41296525	KU122	90	80	41296525	Prananigrum	Bogor
50096487	KD132	80	0	50096487	Pipit	Bekasi
21196353	KU122	75	75	21196353	Quraish	Bogor
NULL	NULL	NULL	NULL	10296001	Fintri	Depok
NULL	NULL	NULL	NULL	21198002	Julizar	Jakarta

Kenudian, jika diberikan query sebagai berikut :

```
SELECT Mahasiswa.NIM, Mahasiswa.NAMA, Nilai.KD_MK, Nilai.MID
FROM Nilai RIGHT OUTER JOIN Mahasiswa ON Nilai.NIM =
Mahasiswa.NIM;
```

Maka Hasilnya :

NIM	NAMA	KD_MK	MID
10296832	Nurhayati	KK021	60
10296126	Astuti	KD132	70
31296500	Budi	KK021	55
41296525	Prananigrum	KU122	90
50096487	Pipit	KD132	80
21196353	Quraish	KU122	75
10296001	Fintri	NULL	NULL
21198002	Julizar	NULL	NULL

2. Fungsi Agregat

Disamping menampilkan nilai-nilai atribut yang ada didalam tabel, sering pula ada kebutuhan untuk menampilkan data-data agregasi, seperti banyaknya record, total nilai suatu atribut, rata-rata nilai atribut, nilai atribut terbesar ataupun nilai atribut terkecil. Berikut yang termasuk fungsi agregasi :

a. COUNT

Digunakan untuk menghitung jumlah atau untuk mendapatkan nilai banyaknya record hasil query.

Contoh : Menghitung jumlah record mahasiswa dari tabel.

Tabel Mahasiswa

NIM	NAMA	ALAMAT
10296832	Nurhayati	Jakarta
10296126	Astuti	Jakarta
31296500	Budi	Depok
41296525	Prananigrum	Bogor
50096487	Pipit	Bekasi
21196353	Quraish	Bogor
10296001	Fintri	Depok
21198002	Julizar	Jakarta

Query : SELECT COUNT(*) FROM MAHASISWA;

Hasilnya: 8

b. SUM

Digunakan untuk menghitung total dari kolom yang mempunyai tipe data numerik.

Contoh : Menghitung total sks dari tabel MATAKULIAH.

KD_MK	NAMA_MK	SKS
KK021	Sistem Basis Data	2
KD132	Sistem Informasi Manajemen	3
KU122	Pancasila	2

Query : SELECT SUM(SKS) AS 'TOTAL SKS' FROM MATAKULIAH;

Hasilnya ;

Total SKS
7

c. AVG

Digunakan untuk menghitung rata-rata dari data dalam sebuah kolom.

Contoh : Menghitung Nilai Rata-rata Nilai Final dari Tabel Nilai

NIM	KD_MK	MID	FINAL
10296832	KK021	60	75
10296126	KD132	70	90
31296500	KK021	55	40
41296525	KU122	90	80
21196353	KU122	75	75
50095487	KD132	80	0

Query : SELECT AVG(FINAL) AS 'FINAL' FROM Nilai;

Hasilnya :

FINAL
60

d. MIN

Digunakan untuk menghitung nilai minimal dalam sebuah kolom.

Contoh : Menghitung Minimal Nilai Final dari Tabel Nilai

Query : SELECT MIN(FINAL) FROM Nilai;

Hasilnya :

FINAL
0

e. MAX

Digunakan untuk menghitung nilai maksimum dalam sebuah kolom

Contoh : Menghitung maksimal nilai MID dari tabel Nilai

Query : SELECT MAX(MID) FROM Nilai;

Hasilnya :

MID
90

3. SubQuery

a. Pengenalan Subquery

Menurut (Ladjamudin, 2004), subquery berarti query didalam query. Dengan menggunakan subquery, maka hasil dari query akan menjadi bagian dari query diatasnya. Subquery terletak didalam klausa WHERE atau HAVING. Pada klausa WHERE, subquery digunakan untuk memilih baris-baris tertentu, yang kemudian digunakan untuk query. Sedangkan pada klausa HAVING, subquery digunakan untuk memilih kelompok baris, yang kemudian digunakan oleh query.

Tabel Pengarang

KD_PENG	NAMA	KOTA
P01	Ashadi	Yogyakarta
P02	Rian	Yogyakarta
P03	Suadi Marwan	Bandung
P04	Siti Halimah	Solo
P05	Amir Hamzah	Kudus
P06	Suparman	Jakarta
P07	Jaja	Bandung
P08	Sama	Bogor

Tabel Buku

KD_BUKU	JUDUL	KD_PENG
B01	Pengantar Basis Data	P02
B02	Pemrograman C++	P01
B03	Sistem Pakar	P03
B04	Visual C++	P05
B05	Pemrograman Pascal	P01
B06	Pemrograman Delphi	P07

Sebagai contoh, diinginkan untuk menampilkan daftar kode pengarang dan nama pengarang (berdasarkan tabel Pengarang) yang kode pengarangnya tercantum pada tabel Buku. Maka query nya :

```
SELECT KD_PENG, NAMA FROM Pengarang WHERE KD_PENG IN
(SELECT KD_PENG FROM BUKU);
```

Pada query diatas, SELECT KD_PENG FROM BUKU disebut dengan subquery, sedangkan SELECT KD_PENG, NAMA berkedudukan sebagai query.

Hasil dari query diatas :

KD_PENG	NAMA
P01	Ashadi
P02	Rian
P03	Suadi Marwan
P05	Amir Hamzah
P07	Jaja

b. Aturan Membuat Subquery

Aturan-aturan untuk membuat subquery, yaitu

- 1) Klausa Order By tidak boleh digunakan di subquery, Order By hanya dapat digunakan di pernyataan Select luar.
- 2) Klausa subquery Select harus berisi satu nama kolom tunggal atau ekspresi kecuali untuk subquery-subquery menggunakan kata kunci EXIST

3) Secara default nama kolom di subquery mengacu ke nama tabel di klausa FROM dari subquery tersebut.

4) Saat subquery adalah salah satu dua operan dilibatkan di perbandingan, subquery harus muncul di sisi kanan perbandingan.

c. Penggunaan ANY dan ALL

Digunakan berkaitan dengan subquery. Jika subquery diawali kata kunci ANY, syaratnya akan bernilai TRUE jika dipenuhi sedikitnya satu nilai yang dihasilkan subquery tersebut atau dapat pula dikatakan menghasilkan TRUE kalau paling tidak salah satu perbandingan dengan hasil subquery menghasilkan nilai TRUE.

Contoh :

Menampilkan semua nim yang nilai mid nya bukan yang terkecil

```
SELECT nim FROM Nilai WHERE MID > ANY ( SELECT MID FROM
Nilai);
```

Sedangkan jika subquery diawali kata kunci ALL, maka syarat hanya akan bernilai TRUE jika dipenuhi semua nilai yang dihasilkan subquery itu.

Contoh :

```
SELECT NIM FROM Nilai WHERE MID >= ALL (SELECT MID FROM
Nilai);
```

d. Penggunaan EXIST dan NOT EXIST

EXIST akan mengirim nilai TRUE jika dan hanya jika terdapat sedikitnya satu baris di tabel hasil yang dikirim oleh subquery dan EXIST mengirim

nilai FALSE jika subquery mengirim tabel kosong. Untuk NOT EXIST kebalikan dari EXIST.

Contoh :

```
SELECT NIM, NAMA FROM Mahasiswa WHERE EXISTS (SELECT *
FROM Nilai WHERE NIM=Nilai.NIM);
```

e. Contoh Subquery

- 1) Ambil nilai mid dan final dari mahasiswa yang bernama Astuti.

```
SELECT MID, FINAL FROM NILAI WHERE NIM=( SELECT NIM
FROM MAHASISWA WHERE NAMA='Astuti')
```

- 2) Ambil nilai kode matakuliah, mid dan final dari mahasiswa yang

```
tinggal di jakarta. SELECT KD_MK, MID, FINAL FROM NILAI
WHERE NIM IN(SELECT NIM FROM MAHASISWA WHERE
ALAMAT = 'Jakarta')
```

- 3) Ambil nama-nama mahasiswa yang mengikuti ujian. SELECT NAMA

```
FROM MAHASISWA WHERE EXISTS (SELECT NIM FROM
NILAI WHERE NILAI.NIM= MAHASISWA.NIM)
```

- 4) Ambil nama-nama mahasiswa yang tidak mengikuti ujian. SELECT

```
NAMA FROM MAHASISWA WHERE NOT EXISTS (SELECT
NIM FROM NILAI WHERE NILAI.NIM= MAHASISWA.NIM)
```