

Pertemuan 8

OUTER JOIN DAN KLAUSA HAVING

OUTER JOIN digunakan untuk menampilkan data yang tidak memiliki padanan dengan isi dari tabel lain yang dilakukan JOIN padanya. Jika tidak ada padanan, isi record akan bernilai null. Namun jika dilakukan agregasi, maka yang ditampilkan adalah nilai 0.

Ketika menggunakan OUTER JOIN, programmer dapat menambahkan klausa LEFT atau RIGHT pada JOIN yang telah terbentuk. Harap diperhatikan bahwa jika menggunakan klausa LEFT, maka data yang akan ditampilkan semuanya berada di tempatkan paling kiri sementara data yang kemungkinan memiliki nilai null ditempatkan setelahnya. Jika klausa RIGHT yang digunakan, maka penyebutan tabel yang akan ditampilkan semuanya disebut paling terakhir (kanan) sementara tabel-tabel yang memiliki nilai null ditempatkan di sebelah kiri.

HAVING merupakan sebuah klausa yang berfungsi khusus untuk menampung data hasil agregasi. Semua syarat yang lain dapat diletakkan pada klausa WHERE, terkecuali syarat yang mengandung fungsi agregasi.

PRAKTIK

1. Buatlah query untuk menampilkan banyaknya mahasiswa yang berasal dari tiap-tiap daerah.

```
SELECT kota, COUNT(kota)
FROM mahasiswa
GROUP BY kota
```

2. Modifikasi soal tersebut untuk menampilkan yang banyak mahasiswanya lebih dari dua orang.

```
SELECT kota, COUNT(kota)
FROM mahasiswa
GROUP BY kota
HAVING COUNT(kota) > 2
```

3. Buatlah query untuk menampilkan banyaknya dosen yang lahir di tahun yang sama sehingga hasilnya sebagai berikut. (Petunjuk: ekstrak dulu tahun dari tanggal lahir dan kelompokkan berdasarkan tahunnya; soal ini tidak ada HAVING)

tahun	banyak
1958	1
1960	1
1975	1
1978	2
1979	1
1980	1
1982	1
1985	1

4. Buatlah query untuk menampilkan nama matakuliah beserta banyaknya matakuliah yang **telah** diambil oleh mahasiswa, hanya untuk yang jumlah mahasiswanya lebih dari 1 orang.

```
SELECT nama_matakuliah, COUNT(krs.kode_mkul)
FROM matakuliah mk, krs
WHERE mk.kode_mkul = krs.Kode_mkul
GROUP BY nama_matakuliah
HAVING COUNT(krs.kode_mkul) > 1
```

Atau

```
SELECT nama_matakuliah, COUNT(krs.kode_mkul)
FROM matakuliah mk
      JOIN krs ON mk.kode_mkul = krs.Kode_mkul
GROUP BY nama_matakuliah
HAVING COUNT(krs.kode_mkul) > 1
```

5. Modifikasi soal tersebut untuk menampilkan nama dari **semua** matakuliah, baik yang sudah pernah diambil atau tidak, dan tampilkan banyak kelasnya masing-masing. Urutkan matakuliahnya berdasarkan nama mata kuliahnya. Perhatikan bedanya dengan soal nomor 4.

```
SELECT nama_matakuliah, COUNT(krs.kode_mkul)
FROM matakuliah mk
      LEFT JOIN krs ON mk.kode_mkul = krs.Kode_mkul
GROUP BY nama_matakuliah
ORDER BY Nama_matakuliah
```

6. Buatlah query untuk menampilkan nama jurusan dan banyaknya mahasiswa pada tiap-tiap jurusan tersebut, hanya untuk mahasiswa yang lebih dari atau sama 2 orang.

```
SELECT nama_jurusan, COUNT(m.nim)
FROM mahasiswa m, jurusan j
WHERE m.kode_jur=j.kode_jur
GROUP BY nama_jurusan
HAVING COUNT(m.nim) >=2
```

7. Tampilkan nama-nama matakuliah yang sudah ada di tabel transkrip.

```
SELECT Nama_matakuliah, COUNT(t.nilai)
FROM matakuliah mk
      JOIN transkrip t ON mk.Kode_mkul = t.kode_mkuliah
GROUP BY Nama_matakuliah
ORDER BY Nama_matakuliah
```

8. Modifikasi query nomor 7 dan tampilkan hanya matakuliah yang muncul lebih dari sekali.

```
SELECT Nama_matakuliah, COUNT(t.nilai)
FROM matakuliah mk
```

```
        JOIN transkrip t ON mk.Kode_mkul = t.kode_mkuliah  
GROUP BY Nama_matakuliah  
HAVING COUNT(t.nilai) > 1  
ORDER BY Nama_matakuliah
```

9. Modifikasi query nomor 7 dan tampilkan juga mata kuliah yang tidak terdapat di dalam tabel transkrip.

```
SELECT Nama_matakuliah, COUNT(t.nilai)  
FROM matakuliah mk  
    LEFT JOIN transkrip t ON mk.Kode_mkul = t.kode_mkuliah  
GROUP BY Nama_matakuliah  
ORDER BY Nama_matakuliah
```

10. Buatlah query yang menampilkan daftar nama semua dosen beserta banyaknya matakuliah yang telah diampu olehnya. (termasuk dosen yang sama sekali belum pernah mengampu mata kuliah). Urutkan nama-nama dosen berdasarkan abjad.