

Pertemuan 5

Fungsi Banyak Baris

Fungsi ini juga disebut dengan fungsi agregasi/pengelompokan.

Group functions atau disebut juga fungsi pengelompokan atau fungsi banyak baris (*multiple-row function*) adalah fungsi yang dilakukan pada sekelompok baris untuk memberikan satu hasil untuk setiap satu kelompok, lain halnya dengan fungsi satu baris (*single-row function*).

Ada beberapa tipe fungsi pengelompokan yang paling banyak digunakan adalah:

1. **AVG** ([DISTINCT|ALL] *n*), menghitung rata-rata dari semua data yang di-query (*n*), tidak termasuk yang bernilai null.
2. **COUNT** ({* | [DISTINCT|ALL] *ekspr*}), mencacah jumlah baris, dimana *ekspr* merupakan sesuatu nilai bukan null. Jika hendak digunakan untuk mencacah semua baris gunakan tanda *, maka yang terhitung adalah semua baris yang muncul lebih dari 1 kali dan juga baris yang null.
3. **MAX** ([DISTINCT|ALL] *ekspr*), nilai maksimum dari *ekspr*, tidak termasuk nilai null.
4. **MIN** ([DISTINCT|ALL] *ekspr*), nilai minimum dari *ekspr*, tidak termasuk nilai null.
5. **SUM** ([DISTINCT|ALL] *n*), menghitung jumlah dari *n*, tidak termasuk nilai null.

Mengelompokkan Data

Hingga saat ini, semua *group function* memperlakukan tabel sebagai satu kumpulan informasi yang sangat besar. Ada saatnya informasi tersebut dipecah atas kelompok-kelompok yang lebih kecil.

Perhatikan sintaks cara penggunaan fungsi pengelompokan seperti yang tertulis di bawah ini.

```
SELECT [kolom, ] group_function(kolom), ...  
FROM nama_tabel  
[WHERE kondisi]  
[GROUP BY ekspresi_group_by]  
[ORDER BY kolom];
```

Petunjuk untuk melakukan **GROUP BY** adalah:

- Semua kolom yang terdapat pada klausa **SELECT** harus ditulis di bagian klausa **GROUP BY**, KECUALI kolom yang dikenakan fungsi pengelompokan.
- Akan tetapi, yang terdapat pada klausa **GROUP BY** tidak harus berada di klausa **SELECT**.

Selain itu, **GROUP BY** digunakan pada saat user ingin mencampurkan antara *individual item* dengan fungsi pengelompokan dalam satu perintah **SELECT**.

Jika hal ini terjadi namun klausa **GROUP BY** tidak ditulis, maka akan muncul error. Error ini disebut juga dengan *illegal queries*.

Kerjakan dan analisislah semua bagian praktik berikut ini dengan menggunakan skema yang diperoleh pada **Pertemuan 2**.

PRAKTIK

1. Buat query untuk menampilkan banyaknya mahasiswa dari semua jurusan.
SELECT COUNT(*)
FROM MAHASISWA;
2. Buat query untuk menampilkan banyaknya mahasiswa yang berasal dari kota 'Yogya' atau 'Yogyakarta'
SELECT COUNT(*)
FROM MAHASISWA
WHERE kota = 'Yogya' OR kota = 'Yogyakarta';
3. Buat query untuk menampilkan banyaknya dosen dalam perguruan tinggi tersebut.
SELECT COUNT(*)
FROM dosen;
4. Buatlah query untuk menampilkan banyak jurusan pada perguruan tinggi tersebut
SELECT COUNT(*)
FROM jurusan;
5. Buatlah query untuk menampilkan banyaknya mahasiswa laki-laki di perguruan tinggi tersebut.
SELECT COUNT(*)
FROM mahasiswa
WHERE jenis_kelamin = 'P';
6. Buatlah query untuk menampilkan banyaknya mahasiswa perempuan di perguruan tinggi tersebut.
SELECT COUNT(*)
FROM mahasiswa
WHERE jenis_kelamin = 'W';
7. Buatlah query untuk menampilkan banyaknya dosen yang lahir pada tahun 1978.
SELECT COUNT(YEAR(tanggal_lahir))
FROM dosen
WHERE YEAR(tanggal_lahir) = '1978';
8. Buatlah query untuk mencari mahasiswa yang tercatat paling muda.
Catatan: Query ini pasti hanya menghasilkan 1 output sehingga tidak memerlukan GROUP BY
SELECT nama_mhs, MIN(tgl_lahir)
FROM mahasiswa;

Catatan: Jika query pengelompokan menampilkan lebih dari satu kolom/field, maka GROUP BY wajib disertakan. Jika tidak, maka hasil query akan salah (bahkan pada beberapa DBMS akan keluar pesan error). Ketik dan perhatikan contoh berikut.

9. Buat query untuk menampilkan banyaknya KRS yang telah dilakukan tiap-tiap mahasiswa.

```
SELECT nim, COUNT(*)  
FROM krs  
GROUP BY nim;
```

10. Buat query untuk menampilkan banyaknya kota beserta banyaknya mahasiswa yang berada dari kota-kota tersebut.

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

11. Buatlah query untuk menampilkan tahun lahir beserta banyaknya mahasiswa yang lahir pada masing-masing tahun tersebut.

```
SELECT tgl_lahir "Tahun Lahir", COUNT(YEAR(tgl_lahir))  
FROM mahasiswa  
GROUP BY "Tahun Lahir";
```

12. Buatlah query untuk menampilkan kota tempat tinggal dosen beserta banyaknya dosen dari kota tersebut.

13. Buatlah query untuk menampilkan jurusan dan banyaknya dosen dari tiap jurusan tersebut.

14. Buatlah query untuk mencari dosen tertua yang pernah ada. (Query ini hanya menghasilkan 1 output, sehingga tidak memerlukan GROUP BY)