

Oracle Academic Initiative

Oracle9i Introduction to SQL



Oleh:

Tessy Badriyah, SKom.MT

**Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember
Surabaya**

BAB 6 : Aggregate Data dengan Fungsi GROUP

6.1. Sasaran

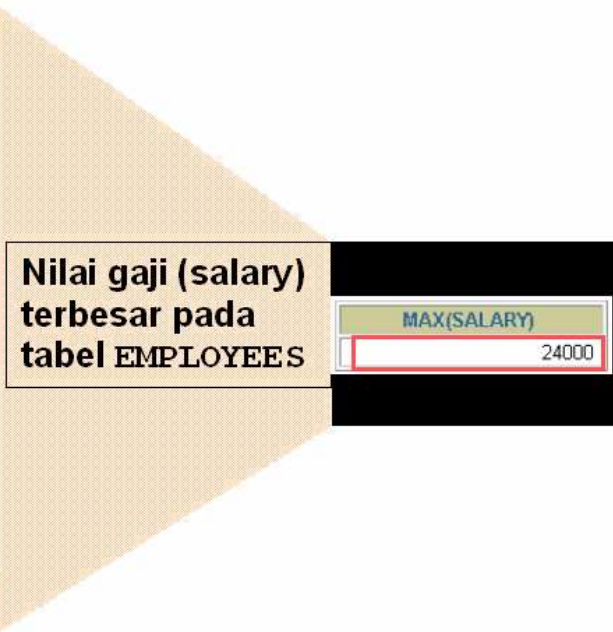
- Mengidentifikasi Fungsi Group yang tersedia
- Menggambarkan Penggunaan dari Fungsi Group
- Mengelompokkan data dengan Klausa GROUP BY
- Meng-include atau meng-exclude suatu baris data dengan klausa HAVING

6.2. Apa itu Fungsi Group

Tidak seperti fungsi baris tunggal, fungsi group beroperasi pada himpunan baris dan memberikan satu hasil per group.

EMPLOYEES

DEPARTMENT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
50	5800
50	3500
50	3100
50	2600
50	2500
80	10500
80	11000
80	8600
	7000
10	4400



...
20 rows selected.

6.3. Tipe-tipe Fungsi Group

Tipe-tipe fungsi group yaitu : AVG, COUNT, MAX, MIN, STDDEV, SUM, VARIANCE.

6.4. Sintak dari Fungsi Group

Sintak (cara penulisan) dari penggunaan Fungsi Group :

```
SELECT [DISTINCT] {*, column [alias] group_function(column) }
FROM table;
[WHERE condition(s) ]
[GROUP BY column]
[ORDER BY column];
```

6.5. Penggunaan Fungsi AVG dan SUM

Fungsi AVG dan SUM digunakan pada data numeric untuk mencari nilai rata-rata dan jumlah total dari sekumpulan data.

```
SELECT AVG(salary) , MAX(salary) ,
       MIN(salary) , SUM(salary)
FROM employees
WHERE job_id LIKE '%REP%' ;
```

AVG(SALARY)	MAX(SALARY)	MIN(SALARY)	SUM(SALARY)
8150	11000	6000	32600

6.6. Penggunaan Fungsi MIN dan MAX

Sedangkan fungsi MIN dan MAX dapat digunakan untuk sembarang tipe data.

```
SELECT MIN(hire_date), MAX(hire_date)
FROM employees;
```

MIN(HIRE_	MAX(HIRE_
17-JUN-87	29-JAN-00

6.7. Penggunaan Fungsi COUNT

Fungsi COUNT digunakan untuk mencari jumlah baris dalam table.

```
SELECT COUNT (*)
FROM employees
WHERE department_id = 50;
```

COUNT(*)
5

Fungsi COUNT hanya mengembalikan hasil dari baris yang tidak mengandung nilai NULL saja, karena fungsi COUNT mengabaikan nilai NULL.

```
SELECT COUNT (*)
FROM employees
WHERE department_id = 50;
```

COUNT(*)
5

6.8. Penggunaan Keyword DISTINCT

COUNT(DISTINCT expr) mengembalikan sejumlah nilai tidak null yang bersifat distinct. Contoh akan ditampilkan jumlah nilai departemen yang distinct dari tabel EMPLOYEES.

```
SELECT COUNT (DISTINCT department_id)
FROM employees;
```

COUNT(DISTINCTDEPARTMENT_ID)
7

6.9. Fungsi Group dan Nilai Null

Fungsi Group mengabaikan nilai NULL.

6.10. Penggunaan fungsi NVL dengan Fungsi Group

Fungsi NVL digunakan bersama dengan fungsi Group untuk mencakup baris data yang mengandung nilai NULL.

```
SELECT AVG (NVL (commission_pct, 0))
FROM employees;
```

AVG(NVL(COMMISSION_PCT,0))
.0425

6.11. Pembuatan Group Data

Hasil dari query dapat dikelompokkan berdasarkan suatu criteria tertentu dengan menggunakan klausa GROUP BY. Misal rata-rata gaji dikelompokkan berdasarkan nomer department.

EMPLOYEES

DEPARTMENT_ID	SALARY
10	4400
20	13000
20	6000
50	5800
50	3500
50	3100
50	2500
50	2600
60	9000
60	6000
60	4200
80	10500
80	8600
80	11000
90	24000
90	17000

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333
90	19333.3333
110	10150
	7000

Rata-rata Gaji (salary) dalam tabel EMPLOYEES untuk Tiap departemen.

...
20 rows selected.

6.12. Pembuatan Group Data : Sintak Klausa GROUP BY

```
SELECT      column, group_function(column)
FROM        table
[WHERE      condition]
[GROUP BY  group_by_expression]
[ORDER BY  column];
```

6.13. Penggunaan Klausa GROUP BY

```
SELECT  department_id, AVG(salary)
FROM    employees
GROUP BY department_id ;
```

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333
90	19333.3333
110	10150
	7000

8 rows selected.

Kolom pada klausa GROUP BY tidak harus berada dalam klausa SELECT, seperti pada contoh berikut

```
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id ;
```

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333
90	19333.3333
110	10150
	7000

8 rows selected.

6.14. Mengelompokkan lebih dari Satu Kolom

Klausula GROUP BY juga dapat digunakan pada lebih dari satu kolom.

DEPARTMENT_ID	JOB_ID	SALARY
90	AD_PRES	24000
90	AD_VP	17000
90	AD_VP	17000
60	IT_PROG	9000
60	IT_PROG	6000
60	IT_PROG	4200
50	ST_MAN	5800
50	ST_CLERK	3500
50	ST_CLERK	3100
50	ST_CLERK	2600
50	ST_CLERK	2500
80	SA_MAN	10500
80	SA_REP	11000
80	SA_REP	8600
...		
20	MK_REP	6000
110	AC_MGR	12000
110	AC_ACCOUNT	8300

20 rows selected.

“Jumlahkan Gaji pada tabel EMPLOYEES Untuk tiap job, dikelompokkan berdasarkan departemen.”

DEPARTMENT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
50	ST_CLERK	11700
50	ST_MAN	5800
60	IT_PROG	19200
80	SA_MAN	10500
80	SA_REP	19600
90	AD_PRES	24000
90	AD_VP	34000
110	AC_ACCOUNT	8300
110	AC_MGR	12000
	SA_REP	7000

13 rows selected.

6.15. Penggunaan Klausula GROUP BY pada lebih dari satu Kolom

Contoh penggunaan klausula GROUP BY pada lebih dari satu kolom :

```
SELECT department_id dept_id, job_id, SUM(salary)
FROM employees
GROUP BY department_id, job_id ;
```

DEPT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
50	ST_CLERK	11700
50	ST_MAN	5800
60	IT_PROG	19200
80	SA_MAN	10500
80	SA_REP	19600
90	AD_PRES	24000
90	AD_VP	34000
110	AC_ACCOUNT	8300
110	AC_MGR	12000
	SA_REP	7000

13 rows selected.

6.16. Illegal Queries dengan Fungsi Group

Kesalahan yang mungkin terjadi pada query dengan Fungsi Group adalah menampilkan kolom pada statement SELECT yang tidak menggunakan fungsi group sedangkan kolom lainnya menggunakan fungsi group, dan kolom yang bukan fungsi group tersebut belum dicantumkan dalam klausa GROUP BY. Contoh kesalahannya seperti ini :

```
SELECT department_id, COUNT(last_name)
FROM employees;
```

```
SELECT department_id, COUNT(last_name)
*
ERROR at line 1:
ORA-00937: not a single-group group function
```

6.16. Pengecualian Hasil dari Fungsi Group : dengan Klausa HAVING

Klausa WHERE tidak boleh digunakan pada query yang sudah dikelompokkan dengan menggunakan Fungsi Group atau klausa GROUP BY.

Contoh kesalahannya seperti ini :

```
SELECT department_id, AVG(salary)
FROM employees
WHERE AVG(salary) > 8000
GROUP BY department_id;
```

```
WHERE AVG(salary) > 8000
*
ERROR at line 3:
ORA-00934: group function is not allowed here
```

Yang benar adalah penggunaan kondisi pada fungsi Group menggunakan HAVING, bukan menggunakan WHERE.

6.17. Penggunaan Klausa HAVING

Klausa HAVING digunakan untuk melakukan pembatasan pada group:

1. Baris yang di-group-kan.
2. Fungsi group yang digunakan
3. Group yang sesuai dengan klausa HAVING saja yang ditampilkan

Sintak dari penggunaan HAVING :

```
SELECT      column, group_function
FROM        table
[WHERE      condition]
[GROUP BY  group_by_expression]
[HAVING     group_condition]
[ORDER BY  column];
```

Contoh penggunaan klausa HAVING :

```
SELECT department_id, MAX(salary)
FROM employees
GROUP BY department_id
HAVING MAX(salary) > 10000 ;
```

DEPARTMENT_ID	MAX(SALARY)
20	13000
80	11000
90	24000
110	12000

Penggunaan HAVING dengan WHERE :

```
SELECT job_id, SUM(salary) PAYROLL
FROM employees
WHERE job_id NOT LIKE '%REP%'
GROUP BY job_id
HAVING SUM(salary) > 13000
ORDER BY SUM(salary) ;
```

JOB_ID	PAYROLL
IT_PROG	19200
AD_PRES	24000
AD_VP	34000

6.18. Fungsi Group Bersarang

Fungsi Group bisa digunakan secara *nested* (bersarang)

Berikut penggunaan Fungsi Group bersarang untuk menampilkan nilai maksimum dari rata-rata gaji berdasarkan departemen.

```
SELECT MAX(AVG(salary))
FROM employees
GROUP BY department_id;
```

MAX(AVG(SALARY))
19333.3333

6.19. Latihan

1. Tampilkan nilai tertinggi, terendah, jumlah dan rata-rata gaji dari seluruh pegawai. Beri judul kolom MAKSIMUM MINIMUM TOTAL RATA2.

MAKSIMUM	MINIMUM	TOTAL	RATA2
5000	800	29025	2073.21429

2. Tampilkan nilai tertinggi, terendah, jumlah dan rata-rata gaji pada tiap-tiap jenis pekerjaan yang ada.. Beri judul kolom MAKSIMUM MINIMUM TOTAL RATA2.

JOB	MAKSIMUM	MINIMUM	TOTAL	RATA2
ANALYST	3000	3000	6000	3000
CLERK	1300	800	4150	1037.5
MANAGER	2975	2450	8275	2758.33333
PRESIDENT	5000	5000	5000	5000
SALESMAN	1600	1250	5600	1400

3. Tampilkan nama pekerjaan dan jumlah pegawai yang bekerja pada tiap-tiap pekerjaan tersebut

JOB	COUNT(*)
ANALYST	2
CLERK	4
MANAGER	3
PRESIDENT	1
SALESMAN	4

4. Buat query yang menampilkan perbedaan antara nilai terendah dan nilai tertinggi dari gaji pegawai, beri judul kolom PERBEDAAN.

PERBEDAAN
4200