

CHAPTER

5

QUERY
LANJUTAN (2)

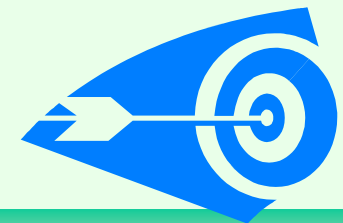
Arif Basofi, S.Kom
Information Technology, PENS-ITS



Objectives

Tujuan:

1. Mengenal ekspresi scalar subquery
2. Memahami korelasi subquery
3. Penggunaan operator EXIST dan NOT EXIST
4. Penggunaan klausa WITH




Lessons

1. Ekspresi Scalar SubQuery
2. Korelasi SubQuery
3. Operator EXIST dan NOT EXIST
4. Penggunaan klausa WITH



Ekspresi Scalar SubQuery

- Ekspresi **Scalar Subquery** adalah sebuah subquery yang hanya mengembalikan satu nilai kolom dari satu baris.
- **Scalar Subquery** dapat digunakan dalam: (pada Oracle 9i)
 - Kondisi dan ekspresi bagian dari **DECODE** dan **CASE**
 - Semua klausa **SELECT** kecuali dalam **GROUP BY**
- Contoh Scalar SubQuery dalam ekspresi **CASE**:

```
SELECT employee_id, last_name,  
       (CASE  
         WHEN department_id = 20  
          (SELECT department_id FROM departments  
           WHERE location id = 1800)  
         THEN 'Canada' ELSE 'USA' END) location  
FROM   employees;
```

Ekspresi Scalar SubQuery

- Contoh Scalar SubQuery dalam klausa **ORDER BY**:

```
SELECT    employee_id, last_name
FROM      employees e
ORDER BY  (SELECT department_name
           FROM departments d
           WHERE e.department_id = d.department_id);
```

Lessons

1. Ekspresi Scalar SubQuery
2. Korelasi SubQuery
3. Operator EXIST dan NOT EXIST
4. Penggunaan klausa WITH



Korelasi SubQuery

- **Korelasi SubQuery** digunakan untuk pemrosesan baris-per-baris (row-by-row).
- Tiap subquery dieksekusi sekali untuk setiap row baris dari outer query.
- Prosesnya sebagai berikut:



Korelasi SubQuery

- Proses korelasi dimulai dengan mengambil baris dari outer query, kemudian inner query dijalankan dengan menggunakan nilai baris kandidat, kemudian nilai dari inner query digunakan untuk melakukan kualifikasi atau mendiskualifikasi baris kandidat.
- Sintaks penulisan korelasi SubQuery:

```
SELECT column1, column2, ...
FROM   table1 outer
WHERE  column1 operator
              (SELECT column1, column2
               FROM   table2
               WHERE  expr1 =
                      outer.expr2);
```

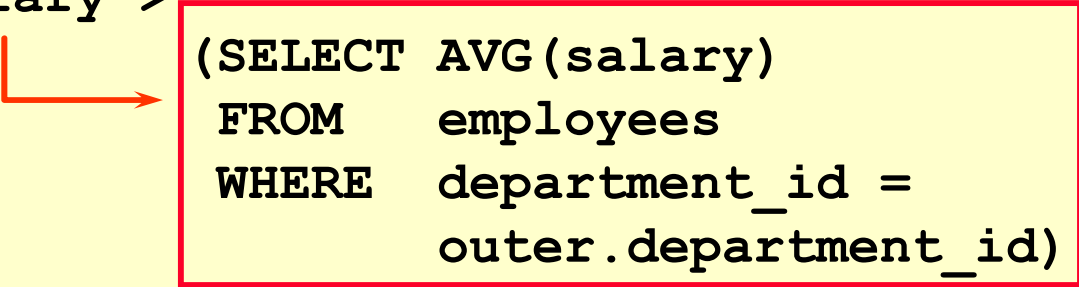
- Sintaks diatas: subquery mereference suatu kolom dari sebuah table dalam parent / outer query.

Korelasi SubQuery

Contoh penggunaan korelasi SubQuery:

1. SubQuery untuk mendapatkan data semua pegawai yang memiliki gaji lebih dari rata-rata gaji pada department tempatnya bekerja.

```
SELECT last_name, salary, department_id
FROM employees outer
WHERE salary >
    (SELECT AVG(salary)
     FROM employees
     WHERE department_id =
       outer.department_id) ;
```



**Setiap saat baris dari
Outer query diproses,
Maka inner query dievaluasi.**

Korelasi SubQuery

- Menampilkan detail pegawai yang pernah berganti job sedikitnya dua kali.

```
SELECT e.employee_id, last_name, e.job_id
FROM   employees e
WHERE  2 <= (SELECT COUNT (*)
             FROM   job_history
             WHERE  employee_id = e.employee_id);
```

EMPLOYEE_ID	LAST_NAME	JOB_ID
101	Kochhar	AD_VP
176	Taylor	SA_REP
200	Whalen	AD_ASST

Korelasi SubQuery

3. Korelasi SubQuery juga dapat digunakan untuk meng-update baris pada satu table berdasarkan pada baris dari table yang lain, korelasi seperti itu dinamakan dengan **Korelasi Update**.

```
UPDATE table1 alias1
SET    column = (SELECT expression
                   FROM    table2 alias2
                   WHERE   alias1.column =
                           alias2.column);
```

Lakukan denormalisasi pada table EMPLOYEES dengan menambahkan satu kolom pada tabel tersebut untuk menyimpan nama departemen.

```
ALTER TABLE employees
ADD (department_name VARCHAR2(14));
```

Korelasi SubQuery

Kemudian isi dari kolom nama departemen yang didapatkan dari tabel DEPARTMENTS dengan menggunakan Korelasi Update :

```
UPDATE employees e
SET    department_name =
        (SELECT department_name
         FROM    departments d
         WHERE   e.department_id = d.department_id);
```

Korelasi SubQuery

4. Korelasi Subquery juga dapat digunakan untuk menghapus baris pada satu table berdasarkan pada baris dari table yang lain, korelasi seperti itu dinamakan dengan **Korelasi Delete**.

```
DELETE FROM table1 alias1
WHERE column operator
      (SELECT expression
       FROM table2 alias2
       WHERE alias1.column = alias2.column);
```

Berikut contoh penggunaan Korelasi DELETE untuk menghapus baris-baris dari tabel EMPLOYEES yang juga terdapat pada tabel EMP_HISTORY.

```
DELETE FROM employees E
WHERE employee_id =
      (SELECT employee_id
       FROM emp_history
       WHERE employee_id = E.employee_id);
```

Lessons

1. Ekspresi Scalar SubQuery
2. Korelasi SubQuery
3. Operator EXIST dan NOT EXIST
4. Penggunaan klausa WITH



Operator **EXIST** dan **NOT EXIST**

- Operator **EXIST** dan **NOT EXIST** digunakan untuk menguji keberadaan (ada/tidak ada) dari baris dalam himpunan hasil dari subquery.
- Jika **ditemukan**, maka :
Pencarian tidak dilanjutkan dalam inner query dan kondisi ditandai **TRUE**.
- Jika **tidak ditemukan**, maka :
Kondisi ditandai **FALSE** dan kondisi pencarian dilanjutkan dalam inner query.

Operator **EXIST** dan **NOT EXIST**

- Berikut contoh penggunaan operator **EXIST** untuk mencari pegawai yang memiliki sedikitnya satu orang bawahan.

```
SELECT employee_id, last_name, job_id, department_id
FROM employees outer
WHERE EXISTS ( SELECT 'X'
                FROM employees
                WHERE manager_id =
                  outer.employee_id);
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90
103	Hunold	IT_PROG	60
124	Mourgos	ST_MAN	50
149	Zlotkey	SA_MAN	80
201	Hartstein	MK_MAN	20
205	Higgins	AC_MGR	110

8 rows selected.

Operator **EXIST** dan **NOT EXIST**

- Berikut contoh penggunaan operator **NOT EXIST** untuk menampilkan semua departemen yang tidak mempunyai pegawai.

```
SELECT department_id, department_name
FROM departments d
WHERE NOT EXISTS (SELECT 'X'
                   FROM employees
                   WHERE department_id
                     = d.department_id);
```

DEPARTMENT_ID	DEPARTMENT_NAME
190	Contracting

Lessons

1. Ekspresi Scalar SubQuery
2. Korelasi SubQuery
3. Operator EXIST dan NOT EXIST
4. Penggunaan klausa WITH



Penggunaan klausa **WITH**

- Dengan menggunakan klausa **WITH**, kita dapat menggunakan blok query yang sama dalam statement **SELECT** pada saat terjadi lebih dari sekali dalam complex query.
- Klausa **WITH** mendapatkan hasil dari blok query dan menyimpannya dalam tablespace temporer kepunyaan user.
- Klausa **WITH** dapat meningkatkan performansi.

Penggunaan klausa WITH

- Berikut Contoh penggunaan klausa WITH yang digunakan untuk menampilkan nama departemen dan total gaji untuk tiap departemen yang memiliki total gaji lebih besar dari gaji rata-rata pada sembarang department.

WITH

```
dept_costs AS (  
    SELECT d.department_name, SUM(e.salary) AS dept_total  
    FROM employees e, departments d  
    WHERE e.department_id = d.department_id  
    GROUP BY d.department_name),  
avg_cost AS (  
    SELECT SUM(dept_total)/COUNT(*) AS dept_avg  
    FROM dept_costs)  
SELECT *  
FROM dept_costs  
WHERE dept_total >  
    (SELECT dept_avg  
    FROM avg_cost)  
ORDER BY department_name;
```

Question ?

End of Session