MODUL VI

SUBQUERY

A. TUJUAN

1. Memahami keterhubungan entitas di dalam basis data
2. Memahami operasi subquery dan jenis-jenisnya di dalam pengambilan data
3. Mampu menyelesaikan kasus-kasus pengambilan data yang kompleks dengan pendekatan subquery

B. DASAR TEORI

Subquery (Subselect) adalah pernyataan SELECT yang merupakan bagian dari pernyataan lain, misal : INSERT. Pernyataan ORDER BY, FOR UPDATE OF, UNION, INTERSECT atau EXCEPT tidak termasuk dalam pernyataan ini.

Subquery menghasilkan sebuah tabel yang merupakan bagian dari tabel atau view yang diidentifikasikan pada klausa FROM. Pembagian ini dapat digambarkan seperti urutan operasi, dimana hasil dari suatu operasi adalah input bagi operasi lain.

Subquery diperlukan pada saat hasil query tidak berhasil dilakukan dengan hanya melalui satu tabel saja, juga pada saat hasil suatu query digunakan pada klausa WHERE query lainnya. Hasil yang diperoleh dari SUBSELECT tidak dapat ditampilkan oleh “main” SELECT.

Urutan operasi pada Subquery adalah :

1. klausa FROM
2. klausa WHERE
3. klausa GROUP BY
4. klausa HAVING
5. klausa SELECT

SUBQUERY – Coding

Fungsi :

Pada klausa kondisi (WHERE atau HAVING), akses lain seperti SELECT dapat melibatkan beberapa tabel. Ada beberapa cara untuk menggabungkan SELECT tambahan pada klausa SELECT atau

HAVING :

- Perbandingan aritmatik (=, >, <)
- ANY (dikombinasikan dengan =, >=, <=)
- SOME (dikombinasikan dengan =, >=, <=)
- IN

Hasil Subquery menentukan key word manakah yang dapat digunakan (ANY, SOME, IN). UNION tidak diperkenankan untuk digunakan dalam SUBSELECT. Sedangkan aritmatik dapat digunakan.

Sintaks formal subquery diperlihatkan sebagai berikut:

```
SELECT A1, A2, ..., An
FROM r1, r2, r3, ..., rm
WHERE P
(SELECT A1, A2, ..., An
FROM r1, r2, r3, ..., rm
WHERE P)
```

Subquery banyak kolom

Pada subquery dengan banyak kolom, tiap baris dari main query dibandingkan dengan nilai dari subquery multiple-row dan multiple-column.

Pembandingan kolom dalam subquery banyak kolom dapat berupa :

- Pembandingan berpasangan (Pairwise Comparison SubQuery)
- Pembandingan tidak berpasangan (NonPairwise Comparison SubQuery)

Ekspresi scalar query

Ekspresi scalar subquery adalah subquery yang mengembalikan hanya satu nilai kolom dari satu baris. Scalar subquery hanya terbatas pada :

- SELECT Statement (klausa FROM dan WHERE saja)
- Daftar VALUE dari statement INSERT

Korelasi subquery

Korelasi SubQuery digunakan untuk pemrosesan baris per baris. Tiap-tiap subquery dijalankan sekali untuk setiap baris dari outer query. Prosesnya sebagai berikut :

Gambar 3-2 : Proses Korelasi Subquery

Pada gambar 3-2, 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.

Multiple Row subquery

Multiple Row Subquery adalah subquery yang menghasilkan lebih dari satu baris. Untuk multiple row subquery ini yang digunakan adalah operator pembandingan : IN, ANY atau ALL.

Penggunaan Operator ANY dalam Multiple Row Subquery

Operator ANY identik dengan operator SOME, yang membandingkan suatu nilai dengan tiap nilai yang ada dalam subquery.

Operator IN

Operator IN memiliki arti : sama dengan member di dalam list. Sebagai contoh, kita bisa menggunakan operator ini untuk mendapatkan data dosen yang mengajar matakuliah

Operator ALL membandingkan suatu nilai dengan semua nilai yang ada dalam subquery.

- Operator >ALL ekuivalen dengan MAKSIMUM.
- Operator <ALL ekuivalen dengan MINIMUM

Penggunaan Subquery dalam klausa FROM

Subquery dapat digunakan dalam klausa FROM, tujuannya adalah untuk membentuk hasil tabel sementara yang berisi data yang sudah diatur sesuai keperluan.

Operator exists dan non exist

Operator EXISTS dan NOT EXIST digunakan untuk menguji keberadaan 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.

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.

Subquery dan Join

Dalam beberapa kasus sederhana, fungsionalitas subquery dan join dapat dipertukarkan. Dengan kata lain, keduanya dapat digunakan untuk menyelesaikan persoalan yang sama

C. HASIL DAN PEMBAHASAN

❖ Latihan Praktikum

Menciptakan database modul6

```
mysql> create database modul6;
Query OK, 1 row affected (0.01 sec)
```

Meggunakan database modul6

```
mysql> use modul6;
Database changed
```

Membuat tabel dosen

```
mysql> CREATE TABLE dosen (
    ->  kode_dos INT (5) NOT NULL,
    ->  nama_dos VARCHAR (30) NOT NULL,
    ->  alamat_dos VARCHAR (30) NOT NULL,
    ->  PRIMARY KEY (kode_dos)
    -> )ENGINE = MyISAM;
Query OK, 0 rows affected (0.01 sec)
```

Membuat tabel jurusan

```
mysql> CREATE TABLE jurusan (
    ->  kode_jur VARCHAR (5) NOT NULL,
    ->  nama_jur VARCHAR (30) NOT NULL,
    ->  kode_dos INT (5) NOT NULL,
    ->  PRIMARY KEY (kode_jur)
    -> )ENGINE = MyISAM;
Query OK, 0 rows affected (0.01 sec)
```

Membuat tabel mahasiswa

```
mysql> CREATE TABLE mahasiswa(
    ->  NIM int (5) NOT NULL,
    ->  nama varchar (30) NOT NULL,
    ->  jenis_kelamin varchar (5) NOT NULL,
    ->  alamat varchar (15) NOT NULL
    -> )ENGINE = MyISAM;
Query OK, 0 rows affected (0.00 sec)
```

Membuat tabel ambil_mk

```
mysql> CREATE TABLE ambil_mk(
    ->  NIM int (5) NOT NULL,
    ->  kode_mk varchar (10) NOT NULL
    -> )ENGINE = MyISAM;
Query OK, 0 rows affected (0.00 sec)
```

Membuat tabel matakuliah

```
mysql> CREATE TABLE matakuliah(
    ->  kode_mk varchar (10) NOT NULL,
    ->  nama_mk varchar (15) NOT NULL,
    ->  sks int (5) NOT NULL,
    ->  semester int (5) NOT NULL,
    ->  kode_dos INT (5) NOT NULL
    -> )ENGINE = MyISAM;
Query OK, 0 rows affected (0.00 sec)
```

mengisi tabel dosen

```
mysql> INSERT INTO dosen VALUES
    -> (10, "Suharto", "Jl. Jombang"),
    -> (11, "Martono", "Jl. Kalpataru"),
    -> (12, "Rahmawati", "Jl. Jakarta"),
    -> (13, "Bambang", "Jl. Bandung"),
    -> (14, "Nurul", "Jl. Raya Tidar");
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

mengisi tabel jurusan

```
mysql> INSERT INTO jurusan VALUES
    -> ("TE", "Teknik Elektro", 10),
    -> ("TM", "Teknik Mesin", 13),
    -> ("TS", "Teknik Sipil", 23);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

mengisi tabel mahasiswa

```
mysql> INSERT INTO mahasiswa VALUES
    -> (101, "Arif", "L", "Jl. Kenangan"),
    -> (102, "Budi", "L", "Jl. Jombang"),
    -> (103, "Wati", "P", "Jl. Surabaya"),
    -> (104, "Ika", "P", "Jl. Jombang"),
    -> (105, "Tono", "L", "Jl. Jakarta"),
    -> (106, "Iwan", "L", "Jl. Bandung"),
    -> (107, "Sari", "P", "Jl. Malang");
Query OK, 7 rows affected (0.00 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

mengisi tabel ambil_mk

```
mysql> INSERT INTO ambil_mk VALUES
    -> (101, "PKT1447"),
    -> (103, "TIK333"),
    -> (104, "PTI333"),
    -> (104, "PTI777"),
    -> (111, "PTI123"),
    -> (123, "PTI999");
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

mengisi tabel matakuliah

```
mysql> INSERT INTO matakuliah VALUES
    -> ("PTI447", "Praktikum Basis Data", 1, 3, 11),
    -> ("TIK342", "Praktikum Basis Data", 1, 3, 11),
    -> ("PTI333", "Basis Data Terdistribusi", 3, 5, 10),
    -> ("TIK123", "Jaringan Komputer", 2, 5, 33),
    -> ("TIK333", "Sistem Operasi", 3, 5, 10),
    -> ("PTI123", "Grafika Multimedia", 3, 5, 12),
    -> ("PTI777", "Sistem Informasi", 2, 3, 99);
Query OK, 7 rows affected, 6 warnings (0.01 sec)
Records: 7  Duplicates: 0  Warnings: 6
```

Scalar Subquery

```
mysql> SELECT *
    -> FROM mahasiswa
    -> WHERE jenis_kelamin=
    -> (SELECT jenis_kelamin
    -> FROM mahasiswa
    -> WHERE nama="Wati");
+-----+------+---------------+--------------+
| NIM | nama | jenis_kelamin | alamat       |
+-----+------+---------------+--------------+
| 103 | Wati | P             | Jl. Surabaya |
| 104 | Ika  | P             | Jl. Jombang  |
| 107 | Sari | P             | Jl. Malang   |
+-----+------+---------------+--------------+
3 rows in set (0.05 sec)
```

Penjelasan:

Mendapatkan data mahasiswa yang Jenis kelaminnya sama dengan mahasiswa dengan nama “Wati”. Pada table diatas menggunakan fungsi SELECT* karena menyeleksi seluruh isi dari table mahasiswa dimana jenis kelamin yang sama dengan wati akan di tampilkan pada table.

Multiple-row query

```
mysql> SELECT d.kode_dos, d.nama_dos
    -> FROM dosen d
    -> WHERE d.kode_dos IN
    -> (SELECT kode_dos
    -> FROM matakuliah);
+----------+-----------+
| kode_dos | nama_dos  |
+----------+-----------+
|       10 | Suharto   |
|       11 | Martono   |
|       12 | Rahmawati |
+----------+-----------+
3 rows in set (0.00 sec)
```

Penjelasan:

Pada table diatas menggunakan fungsi SELECT pada kode dosen dan nama dosen, sehingga yang akan di tampilkan yaitu kode dosen dan nama dosen, dengan menggunakan IN maka sama dengan member di dalam list yaitu mendapatkan data dosen mengajar matakuliah.

```
mysql> SELECT *
    -> FROM matakuliah
    -> WHERE sks > ANY
    -> (SELECT sks
    -> FROM matakuliah
    -> WHERE semester = 3);
+---------+-----------------+-----+----------+----------+
| kode_mk | nama_mk         | sks | semester | kode_dos |
+---------+-----------------+-----+----------+----------+
| PTI333  | Basis Data Terd |   3 |        5 |       10 |
| TIK123  | Jaringan Komput |   2 |        5 |       33 |
| TIK333  | Sistem Operasi  |   3 |        5 |       10 |
| PTI123  | Grafika Multime |   3 |        5 |       12 |
| PTI777  | Sistem Informas |   2 |        3 |       99 |
+---------+-----------------+-----+----------+----------+
5 rows in set (0.05 sec)
```

Penjelasan:

Pada table diatas menggunakan fungsi SELECT* sehingga akan menampilkan seluruh nama table dengan data mata kuliah yang menampilkan sks lebih besar dari sembarang sks matakuliah di semester 3

```
mysql> SELECT *
    -> FROM matakuliah
    -> WHERE sks > ALL
    -> (SELECT sks
    -> FROM matakuliah
    -> WHERE semester = 3);
+---------+-----------------+-----+----------+----------+
| kode_mk | nama_mk         | sks | semester | kode_dos |
+---------+-----------------+-----+----------+----------+
| PTI333  | Basis Data Terd |   3 |        5 |       10 |
| TIK333  | Sistem Operasi  |   3 |        5 |       10 |
| PTI123  | Grafika Multime |   3 |        5 |       12 |
+---------+-----------------+-----+----------+----------+
3 rows in set (0.00 sec)
```

Penjelasan

Pada table diatas menggunakan SELECT* sehingga akan menampilkan seluruh isi table matakuliah dimana sks lebih besar dari semua sks matakuliah di semester 3

Multiple-column subquery

```
mysql> SELECT *
    -> FROM matakuliah
    -> WHERE (semester, sks) IN
    -> (SELECT semester, sks
    -> FROM matakuliah
    -> WHERE kode_mk = 'PTI447');
+---------+-----------------+-----+----------+----------+
| kode_mk | nama_mk         | sks | semester | kode_dos |
+---------+-----------------+-----+----------+----------+
| PTI447  | Praktikum Basis |   1 |        3 |       11 |
| TIK342  | Praktikum Basis |   1 |        3 |       11 |
+---------+-----------------+-----+----------+----------+
2 rows in set (0.03 sec)
```

Penjelasan

Pada table diatas menggunakan SELECT* sehingga menamilkan seluruh isi table pada matakuliah dimana dengan menggunakan IN maka data matakuliah yang semester dan sksnya sesuai dengan semester dan sks mata kuliah kode **PTI447** akan ditampilkan

Operator EXIST dan NOT EXIST

```
mysql> SELECT * FROM matakuliah m
    -> WHERE NOT EXISTS (SELECT* FROM ambil_mk a
    -> WHERE m.kode_mk);
+---------+-----------------+-----+----------+----------+
| kode_mk | nama_mk         | sks | semester | kode_dos |
+---------+-----------------+-----+----------+----------+
| PTI447  | Praktikum Basis |   1 |        3 |       11 |
| TIK342  | Praktikum Basis |   1 |        3 |       11 |
| PTI333  | Basis Data Terd |   3 |        5 |       10 |
| TIK123  | Jaringan Komput |   2 |        5 |       33 |
| TIK333  | Sistem Operasi  |   3 |        5 |       10 |
| PTI123  | Grafika Multime |   3 |        5 |       12 |
| PTI777  | Sistem Informas |   2 |        3 |       99 |
+---------+-----------------+-----+----------+----------+
7 rows in set, 42 warnings (0.00 sec)
```

```
mysql> SELECT *
    -> FROM matakuliah
    -> WHERE sks = (SELECT MIN(sks)
    -> FROM matakuliah);
+---------+-----------------+-----+----------+----------+
| kode_mk | nama_mk         | sks | semester | kode_dos |
+---------+-----------------+-----+----------+----------+
| PTI447  | Praktikum Basis |   1 |        3 |       11 |
| TIK342  | Praktikum Basis |   1 |        3 |       11 |
+---------+-----------------+-----+----------+----------+
2 rows in set (0.00 sec)
```

Penjelasan

Dengan menggunakan EXIST dan NOT EXIST maka dapat dipilih dimana pada tabel matakuliah siswa yang yang mengambil matakuliah yang diambil dan tidak diambil

Subquery dan fungsi agregat

```
mysql> SELECT *
    -> FROM matakuliah
    -> WHERE sks = (SELECT MIN(sks)
    -> FROM matakuliah);
+---------+-----------------+-----+----------+----------+
| kode_mk | nama_mk         | sks | semester | kode_dos |
+---------+-----------------+-----+----------+----------+
| PTI447  | Praktikum Basis |   1 |        3 |       11 |
| TIK342  | Praktikum Basis |   1 |        3 |       11 |
+---------+-----------------+-----+----------+----------+
2 rows in set (0.00 sec)
```

Penjelasan

Pada table diatas, dengan menggunakan SELECT MIN maka akan menampilkan data mahasiswa yang meliliki sks sama dengan sks terkecil dari table matakuliah.

Subquery dan join

```
mysql> SELECT d.kode_dos, d.nama_dos
    -> FROM dosen d
    -> WHERE d.kode_dos NOT IN
    -> (SELECT kode_dos
    -> FROM matakuliah);
+----------+----------+
| kode_dos | nama_dos |
+----------+----------+
|       13 | Bambang  |
|       14 | Nurul    |
+----------+----------+
2 rows in set (0.00 sec)

mysql> SELECT d.kode_dos, d.nama_dos
    -> FROM dosen d
    -> LEFT OUTER JOIN matakuliah m
    -> ON d.kode_dos = m.kode_dos
    -> WHERE m.kode_dos IS NULL;
+----------+----------+
| kode_dos | nama_dos |
+----------+----------+
|       13 | Bambang  |
|       14 | Nurul    |
+----------+----------+
2 rows in set (0.00 sec)
```

Penjelasan

Pada table diatas mendapatkan kode dosen dan nama dosen yang tidak mengajar matakuliah yang diambil dari table dosen. Dengan menggunkan NOT IN.

D. Tugas Praktikum

1. Dapatkan data mahasiswa yang alamatnya sama dengan mahasiswa dengna nim 102, tidak termasuk mahasiswa tersebut.

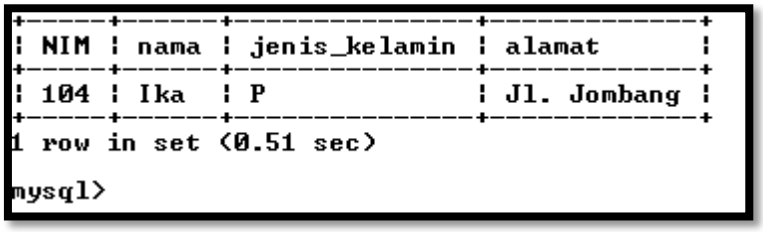
   a. SQL

```
select * from mahasiswa where alamat =

(select alamat from mahasiswa where nim=102)

and nim not like 102;
```

   b. Output



2. Dapatkan matakuliah yang tidak diajar oleh dosen terdaftar

   a. SQL

```
select nama_mk from matakuliah where not exists

(select kode_dos from dosen where kode_dos = matakuliah.kode_dos)
```

   b. Output

3. Dapatkan adata dosen yang mengajar matakuliah dengan sks lebih kecil dari sembarang sks

   a. SQL

```
select distinct d.* from dosen d inner join matakuliah k on
d.kode_dos=k.kode_dos where sks < any (select distinct sks from
matakuliah);
```

b. Output

```
mysql> SOURCE D:\prak6.sql
+----------+----------+--------------+
| kode_dos | nama_dos | alamat_dos   |
+----------+----------+--------------+
|       11 | Martono  | Jl.Kalpataru |
+----------+----------+--------------+
1 row in set (0.00 sec)
```

4. Dapatkan data dosen yang mengajar matakuliah yang diajar dosen yang sekaligus menjadi ketua jurusan teknik elektro, tidak termasuk ketua jurusan teknik elektro

a. SQL

```
select distinct d.*from dosen d inner join matakuliah m on d.kode_dos
= m.kode_dos

 where sks < all (select sks from matakuliah where kode_dos in

 (select kode_dos from jurusan where nama_jur='Teknik Elektro'));
```

b. OUTPUT

```
mysql> SOURCE D:\prak6.sql
+----------+----------+--------------+
| kode_dos | nama_dos | alamat_dos   |
+----------+----------+--------------+
|       11 | Martono  | Jl.Kalpataru |
+----------+----------+--------------+
1 row in set (0.11 sec)
```

5. Dapatkan nim, nama, dan alamat mahasiswa yang tempat tinggalnya sama dengan dosen yang mengajar matakuliah dengan sks di bawah rata-rata

a. SQL

```
select nim, nama, alamat from mahasiswa where alamat in

(select distinct alamat_dos from dosen where kode_dos in

(select distinct kode_dos from matakuliah where sks < any

(select avg(sks) from matakuliah)));
```

b. Output

```
mysql> SOURCE D:\prak6.sql
Empty set (0.00 sec)
```

E. Kesimpulan

Dari praktikum yang dilakukan maka dapat disimpulkan bahwa adalah query SELECT yang ada didalam perintah SQL lain—misalnya SELECT, INSERT, atau DELETE.

Terdapat beberapa jenis subquery yaitu:

- Scalar Subquery: Subquery baris tunggal (*scalar*) hanya mengembalikan hasil satu baris data. Pada percobaan yang dilakukan yaitu Mendapatkan data mahasiswa yang Jenis kelaminnya sama dengan mahasiswa dengan nama "Wati"
- Multiple-Row Subquery: Subquery baris ganda (*multiple-row)* mengembalikan lebih dari satu baris data terdapat operator komparasi **IN, ANY / SOME**. Pada latihan yang dilakukan yaitu diataranya data dosen yang mengajar matakuliah dengan menggunakan IN, mendapatkan data matakuliah yang memiliki sks lebih besar dari sembarang sks matakuliah di semester 3 dengan menggunakan ANY/SOME, dan mendapatkan data matakuliah yang memiliki sks lebih besar dari semua sks matakuliah si semester 3 dengan menggunakan ALL
- Multiple-Column Subquery: Subquery kolom ganda (*multiple-column*) mengembalikan lebih dari satu baris dan satu kolom data.
- Operator EXIST dan NOT EXISTS digunakan untuk memeriksa apakah subquery mengembalikan hansil atau tidak.
- Operasi-operasi dalam Subquery

    a. Pada scalar subquery terdapat operator baris tunggal =, >, >=, <, <=, atau < >.
    b. Pada multiple-row subquery terdapat operator komparasi IN, ANY / SOME, atau ALL.

F. Daftar Rujukan

- Modul Praktikum,2011.*SubQuery* .Pendidikan Teknik Informatika, Teknik Elektro, Universitas Negeri Malang.
- http://beginner-sql-tutorial.com/sql-subquery.htm
- http://www.hastinapura.com/subquery.html
- http://expnotes.com/index.php?option=com_content&view=category&id=50&Itemid=79