

## **MODUL 6**

### **SUB QUERY**

#### **A. TUJUAN**

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

#### **B. PETUNJUK**

- Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.
- Pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.
- Kerjakan tugas-tugas praktikum dengan baik, sabar dan jujur.
- Tanyakan kepada asisten/dosen apabila ada hal-hal yang kurang jelas

#### **C. DASAR TEORI**

##### *1. Subquery*

*Subquery* (disebut juga *subselect* atau *nestedselect / query* atau *inner-select*) adalah *query* SELECT yang ada di dalam perintah SQL lain misalnya SELECT, INSERT, UPDATE, atau DELETE. Keberadaan *subquery* secara nyata mampu menyederhanakan persoalan-persoalan rumit berkaitan *querydata*. Sebagai contoh, misal terdapat pernyataan sebagai berikut:

“Dapatkan data mahasiswa yang alamatnya sama dengan mahasiswa dengan nim 104”

Secara normal, diperlukan dua tahapan untuk menyelesaikan kasus di atas. Pertama adalah mendapatkan alamat dari mahasiswa yang memiliki nim 104. Langkah selanjutnya, baru kita bisa mengetahui data mahasiswa yang alamatnya sama dengan mahasiswa dengan nim 104. Adapun dengan memanfaatkan *subquery*, maka penyelesaian kasus di atas hanya memerlukan sebuah *query* (akan dijelaskan nanti).

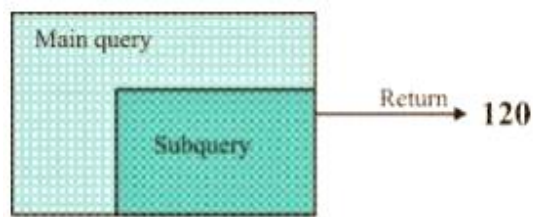
Pada hakekatnya, *subquery* sangat berguna ketika sebuah *query* didasarkan pada nilai-nilai yang tak diketahui. Sintaks formal *subquery* diperlihatkan sebagai berikut:

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

*Subquery* dapat diklasifikasikan ke dalam tiga jenis : *scalar*, *multiple-row*, dan *multiplecolumn*.

a. *ScalarSubquery*

*Subquery* baris tunggal (*scalar*) hanya mengembalikan hasil satu baris data. Bentuk *subquery* ini diperlihatkan seperti Gambar 1.

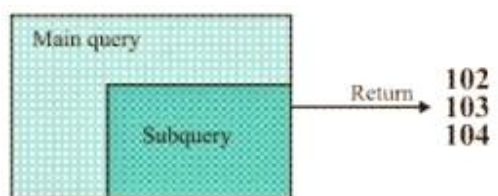


Gambar 1 *ScalarSubquery*

*Subquery* baris tunggal dapat menggunakan operator baris tunggal =, >, >=, <, <=, atau <>.

b. *Multiple-RowSubquery*

*Subquery* baris ganda (*multiple-row*) mengembalikan lebih dari satu baris data. Bentuk *subquery* ini diperlihatkan seperti Gambar 2.

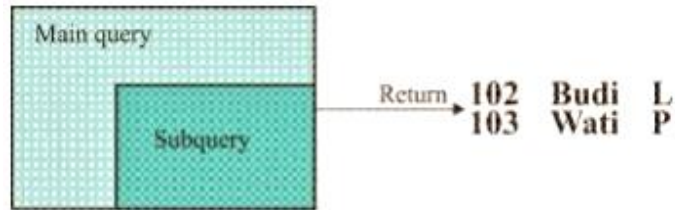


Gambar 2 *Multiple-rowsubquery*

*Subquery* baris ganda dapat menggunakan operator komparasi IN, ANY, SOME, atau ALL.

c. *Multiple-ColumnSubquery*

Subquery kolom ganda (*multiple-column*) mengembalikan lebih dari satu baris dan satu kolom data. Bentuk *subquery* ini diperlihatkan seperti Gambar 3.



Gambar 3 *Multiple-column subquery*

## D. LATIHAN

### 1. Himpunan Entitas

Dalam latihan ini digunakan kembali tiga buah tabel meliputi mahasiswa, ambil\_mk, dan matakuliah yang telah dibuat sebelumnya, dan ditambah dengan dua buah tabel baru, yaitu dosen dan jurusan. Untuk itu, terlebih dahulu buat beberapa tabel dengan struktur sebagai berikut.

#### Tabel Mahasiswa

nim	nama	jk	alamat
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

#### Tabel Matakuliah

kd_mk	nama_mk	sks	semester	kode_dos
PTI123	Grafika Multimedia	3	5	12
PTI333	Basis Data Terdistribusi	3	5	10
PTI447	Praktikum Basis Data	1	3	11
PTI777	Sistem Informasi	2	3	99
TIK123	Jaringan Komputer	2	5	33
TIK333	Sistem Operasi	3	5	10
TIK342	Praktikum Basis Data	1	3	11

#### Tabel Jurusan

Data dosen pada tabel ini berisi kode dosen yang menjadi ketua jurusan.

kode_jur	nama_jur	kode_dos
TE	Teknik Elektro	10
TM	Teknik Mesin	13
TS	Teknik Sipil	23

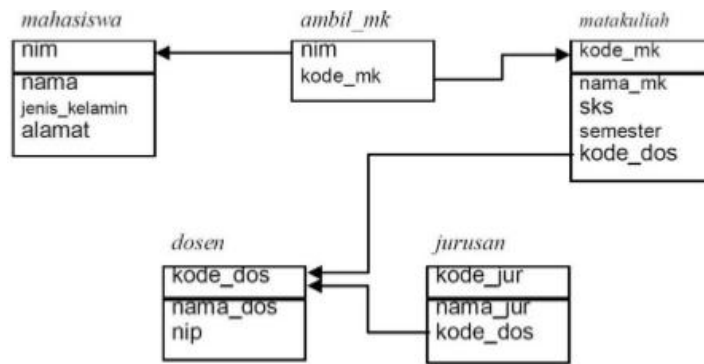
**Tabel Dosen**

kode_dos	nama_dos	alamat_dos
10	Suharto	Jl. Jombang
11	Martono	Jl. Kalpataru
12	Rahmawati	Jl. Jakarta
13	Bambang	Jl. Bandung
14	Nurul	Jl. Raya Tidar

**Tabel Ambil Matakuliah**

nim	kd_mk
101	PTI447
103	TIK333
104	PTI333
104	PTI777
105	PTI123
107	PTI777

Himpunan entitas di atas dapat direpresentasikan kedalam diagram skema (*schema diagram*) seperti berikut.



## 2. ScalarSubquery

Contoh *subquery* baris tunggal adalah mendapatkan data mahasiswa yang jenis kelaminnya sama dengan mahasiswa dengan nama “Wati”.

```

SELECT *
FROM mhs
WHERE jk =
    (SELECT jk
     FROM mhs
     WHERE nama = "Wati")
    
```

Sebagai hasilnya, didapatkan jenis kelamin mahasiswa dengan nama “Wati”, yakni “P” yang selanjutnya digunakan oleh *main query* sehingga menghasilkan sebagai berikut.

nim	nama	jk	alamat
103	Wati	P	Jl. Surabaya
104	Ika	P	Jl. Jombang
107	Sari	P	Jl. Malang

### 3. Multiple-RowSubquery

Pada *subquery* ini, kita menggunakan operator komparasi IN, ANY / SOME, atau ALL.

#### a. 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.

```
SELECT d.kode_dos, d.nama_dos
FROM dosen d
WHERE d.kode_dos IN
      (SELECT kode_dos
       FROM matkul)
```

kode_dos	nama_dos
10	Suharto
11	Martono
12	Rahmawati

#### b. Operator ANY/SOME

Operator ANY / SOME memiliki arti : membandingkan suatu nilai dengan setiap nilai yang dikembalikan oleh *subquery*. Misalkan kita ingin mendapatkan data matakuliah yang memiliki sks lebih besar dari sembarang sks matakuliah di semester 3.

```
SELECT *
FROM matkul
WHERE sks > ANY
      (SELECT sks
       FROM matkul
       WHERE semester = 3)
```

kd_mk	nama_mk	sks	semester	kode_dos
PTI123	Grafika Multimedia	3	5	12
PTI333	Basis DataTerdistribusi	3	5	10
PTI777	Sistem Informasi	2	3	99
TIK123	Jaringan Komputer	2	5	33
TIK333	Sistem Operasi	3	5	10



Operator = **ANY** ekuivalen dengan **IN**.

Operator < **ANY** ekuivalen dengan **MAX** (kurang dari maks).

Operator > **ANY** ekuivalen dengan **MIN** (lebih dari min).

### c. Operator ALL

Operator ALL memiliki arti: membandingkan suatu nilai dengan semua nilai yang dikembalikan oleh *subquery*. Misal, kita ingin mendapatkan data matakuliah yang memiliki sks lebih besar dari semua sks matakuliah di semester 3.

```
SELECT *
FROM matkul
WHERE sks > ALL
  (SELECT sks
   FROM matkul
   WHERE semester = 3)
```

kd_mk	nama_mk	sks	semester	kode_dos
PTI123	Grafika Multimedia	3	5	12
PTI333	Basis Data Terdistribusi	3	5	10
TIK333	Sistem Operasi	3	5	10



Operator < **ALL** ekuivalen dengan **MIN** (kurang dari min).

Operator > **ALL** ekuivalen dengan **MAX** (lebih dari maks)

### 4. Multiple-Column Subquery

*Subquery* kolom ganda (atau tabel) juga menggunakan operator komparasi IN, ANY/SOME, atau ALL. Pada *query* ini, nilai dari *subquery* dalam bentuk kolom ganda dikomparasi main *query*. Sebagai contoh, misalkan kita ingin menampilkan data matakuliah yang semester dan sksnya sesuai dengan semester dan sks matakuliah dengan kode “PTI447”.

kd_mk	nama_mk	sks	semester	kode_dos
PTI447	Praktikum Basis Data	1	3	11
TIK342	Praktikum Basis Data	1	3	11

## 5. Operator EXISTS dan NOT EXISTS

Operator EXISTS dan NOT EXISTS digunakan pada *correlated subquery* untuk memeriksa apakah *subquery* mengembalikan hasil atau tidak. Apabila *subquery* mengembalikan hasil, EXIST akan mengembalikan nilai true. Begitu pula sebaliknya, jika tidak mengembalikan hasil.

### a. EXISTS

Pernyataan berikut akan mendapatkan data matakuliah yang diambil oleh mahasiswa.

```
SELECT *
FROM matkul m
WHERE EXISTS
  (SELECT *
   FROM ambil_mk a
   WHERE m.kd_mk = a.kd_mk)
```

kd_mk	nama_mk	sks	semester	kode_dos
PTI123	Grafika Multimedia	3	5	12
PTI333	Basis DataTerdistribusi	3	5	10
PTI447	Praktikum Basis Data	1	3	11
PTI777	Sistem Informasi	2	3	99
TIK333	Sistem Operasi	3	5	10

### b. NOT EXISTS

Pernyataan berikut akan mendapatkan data matakuliah yang tidak diambil oleh mahasiswa.

```
SELECT *
FROM matkul m
WHERE NOT EXISTS
  (SELECT *
   FROM ambil_mk a
   WHERE m.kd_mk = a.kd_mk)
```

kd_mk	nama_mk	sks	semester	kode_dos
TIK123	Jaringan Komputer	2	5	33
TIK342	Praktikum Basis Data	1	3	11

**E. TUGAS PRAKTIKUM**

1. Dapatkan kode dan nama matakuliah dosen yang menjadi Ketua Jurusan Teknik Elektro.
2. Dapatkan data mahasiswa yang tidak mengambil matakuliah.
3. Dapatkan data dosen yang mengajar matakuliah diatas semester 3.
4. Dapatkan data matakuliah dosen yang bukan merupakan Ketua Jurusan Teknik Elektro.
5. Dapatkan data dosen pengajar matakuliah yang tidak diambil oleh mahasiswa.

**F. TUGAS RUMAH**

1. Dapatkan data dosen yang mengajar matakuliah dengan sks lebih besar dari sembarang sks.
2. Dapatkan data mahasiswa yang tinggal satu wilayah dengan dosen yang bukan merupakan Ketua Jurusan Teknik Elektro.
3. Dapatkan data mahasiswa yang diajar oleh Ketua Jurusan Teknik Elektro.