

# Modul VII

## Sub Query

### A. Kompetensi Dasar :

1. Mahasiswa dapat mengubah query join menjadi query dengan subquery.
2. Mahasiswa dapat membuat query dengan subquery .

### B. Teori Singkat :

1. Subquery adalah perintah SELECT yang berada di dalam perintah SELECT yang lain.

2. MySQL mendukung konsep subquery mulai versi 4.1. Contoh subquery adalah

```
SELECT * FROM t1 WHERE column1 = (SELECT column1 FROM t2)
```

Pada contoh tersebut, `SELECT * FROM t1 ...` adalah query utama/*outer query* dan `(SELECT column1 FROM t2)` adalah *subquery*. Subquery bersifat nested di dalam query utama. Subquery dapat dibuat secara bertingkat sampai 32 tingkat. Subquery harus dituliskan didalam tanda kurung.

3. Beberapa keuntungan menggunakan subquery adalah :
  - a. Memungkinkan membuat query yang terstruktur, sehingga memudahkan pemisahan masing-masing bagian dari perintah/query yang dibuat.
  - b. Sebagai alternatif menggantikan query yang memerlukan JOIN dan UNION yang kompleks.
  - c. Query menjadi lebih mudah dibaca dan dikoreksi, memenuhi kaidah SQL ``Structured Query Language."''
4. Sebuah subquery memberikan beberapa kemungkinan hasil, diantaranya
  - a. Sebuah nilai tunggal (scalar) □ scalar subquery
  - b. Sebuah baris □ row subquery
  - c. Sebuah kolom ataupun □ coloumns subquery
  - d. sebuah tabel. □ table subquery

Masing-masing tipe subquery (berdasarkan nilai yang dihasilkan) tersebut, dapat digunakan pada konteks tertentu yang sesuai.

5. Beberapa batasan dalam membuat subquery :

- a. Outer query dapat berupa salah satu dari : SELECT, INSERT, UPDATE, DELETE, SET, or DO.
- b. Subquery dapat mengandung opsi-opsi yang mendukung SELECT misalnya DISTINCT, GROUP BY, ORDER BY, LIMIT, joins, index hints, UNION, komentar, fungsi, dll.

6. Sintaks subquery secara umum adalah :

```
SELECT kolom1 FROM Tabel WHERE kolom2 [Operator  
Pembandingan] (SELECT kolom1 FROM Tabel WHERE [Condition])
```

[Operator Pembandingan] dapat berupa =, >, <, >=, <= Atauapun LIKE, ANY, SOME, ALL, IN, NOT IN dll

- a. 

```
SELECT M>Nama, B.Judul, B.Penulis, B.Jumlah FROM  
Mahasiswa M, Pinjam P, Buku B WHERE (M.Nim=P.Nim and  
P.kode_buku = B.kode_buku) AND jumlah=(SELECT  
Max(jumlah) FROM `buku`)
```
- b. 

```
SELECT M>Nama, B.Judul, B.Penulis, B.Jumlah FROM  
Mahasiswa M JOIN Pinjam P USING(Nim) JOIN Buku B  
using(kode_buku) WHERE jumlah= ( SELECT Max(jumlah)  
FROM `buku` )
```
- c. 

```
SELECT Judul, Penulis, Jumlah FROM `buku` WHERE  
jumlah= ( SELECT Max(jumlah) FROM `buku` ) UNION  
SELECT Judul, Penulis, Jumlah FROM `buku` WHERE  
jumlah= ( SELECT Min(jumlah) FROM `buku` )
```
- d. *(Bandingkan d dengan e)*  

```
SELECT P.kode_prodi, P.nama_prodi from Prodi P JOIN  
Fakultas F using(kode_fak) where F.Lokasi='Kampus  
III'
```
- e. 

```
SELECT kode_prodi, nama_prodi FROM prodi as P WHERE  
kode_fak IN (SELECT kode_fak from Fakultas as F where  
Lokasi='Kampus III')
```

f. *(Bandingkan f dengan g)*

```
SELECT F.* FROM Fakultas F LEFT JOIN Prodi P USING  
(kode_fak) WHERE P.kode_prodi is NULL
```

g. 

```
SELECT fakultas.* FROM Fakultas WHERE kode_fak <>  
ALL(SELECT kode_fak from prodi)
```

h. *(Bandingkan h dengan i)*

```
SELECT DISTINCT F.* FROM Fakultas F LEFT JOIN Prodi P  
USING (kode_fak) WHERE P.kode_prodi is NOT NULL
```

i. 

```
SELECT * FROM Fakultas WHERE kode_fak = ANY(SELECT  
kode_fak from prodi)
```

### C. Tugas :

1. Tambahkan data pada tabel buku berupa kolom jumlah

	🔑 kode_buku	Judul	penulis	Jumlah ▾	penerbit
1	B01	Sastra Nuklir	Imam Samudra	25	Jamaah Isalmiyah
2	B02	Matematika	Sugiyarto	240	Yudistira
3	B03	Organisasi Komputer	Imam Riadi	75	Andi Offset
4	B04	Struktur data	Suprihatin	100	Andi Offset
5	B05	Mikrobiologi	Hadi Sasaongko	80	Elex Media
6	B06	Akutansi	Sugiarto	120	Andi Star

2. Cobalah contoh-contoh query yang diberikan.
3. Dengan memanfaatkan subquery, tuliskan perintah untuk menampilkan
  - a. Data mahasiswa yang tidak meminjam buku !
  - b. Data buku yang tidak dipinjam mahasiswa !
4. Berikan kesimpulan terhadap-masing-masing query.