

# **LAPORAN PRAKTIKUM BASIS DATA**

## **SQL SUB QUERY**



**DISUSUN OLEH :**

Oktario Mufti Yudha

2320506044

**PROGRAM STUDI S1 TEKNOLOGI INFORMASI**

**JURUSAN TEKNIK ELEKTRO FAKULTAS TEKNIK**

**UNIVERSITAS TIDAR**

**2024**

**LAPORAN**  
**PRAKTIKUM BASIS DATA**



<b>Diisi Mahasiswa Praktikan</b>									
Nama Praktikan	Oktario Mufti Yudha								
NPM	2320506044								
Rombel	04								
Judul Praktikum	SQL SUB QUERY								
Tanggal Praktikum	25 April 2024								
<b>Diisi Asisten Praktikum</b>									
Tanggal Pengumpulan	<table border="1" style="width: 100%;"><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr></table>								
Catatan									

PENGESAHAN		NILAI
Diperiksa oleh :	Disahkan oleh :	
Asisten Praktikum	Dosen Pengampu	
	Imam Adi Nata, S.Kom., M.Kom.	

## I. Tujuan Praktikum

1. Mahasiswa mampu menjelaskan SQL bertingkat dalam basis data
2. Mahasiswa mampu menerapkan SQL bertingkat dalam basis data

## II. Dasar Teori

Subquery, atau yang juga dikenal sebagai nested query, adalah sebuah query (perintah SQL) yang disisipkan ke dalam query utama sebagai bagian dari klausa WHERE, FROM, atau SELECT. Subquery memungkinkan pengguna untuk mengeksekusi query yang lebih kompleks dengan menggabungkan hasil dari satu query ke dalam query lainnya.

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

Subquery digunakan untuk mencapai tujuan tertentu dalam pengambilan data atau manipulasi data yang tidak dapat dicapai dengan menggunakan query tunggal. Dengan subquery, pengguna dapat menggabungkan data dari beberapa tabel, melakukan perbandingan antara hasil query, atau membuat pengambilan data yang lebih terperinci.

Subquery dapat disisipkan ke dalam klausa WHERE, FROM, atau SELECT dari query utama. Dalam klausa WHERE, subquery digunakan untuk memfilter baris berdasarkan kriteria tertentu. Dalam klausa FROM, subquery digunakan untuk menentukan sumber data untuk query utama. Sedangkan dalam klausa SELECT, subquery dapat menghasilkan nilai tunggal yang akan dimasukkan ke dalam hasil query utama.

### III. Metode Praktikum

#### A. Alat dan bahan

Alat :

1. Laptop
2. Keyboard
3. Mouse

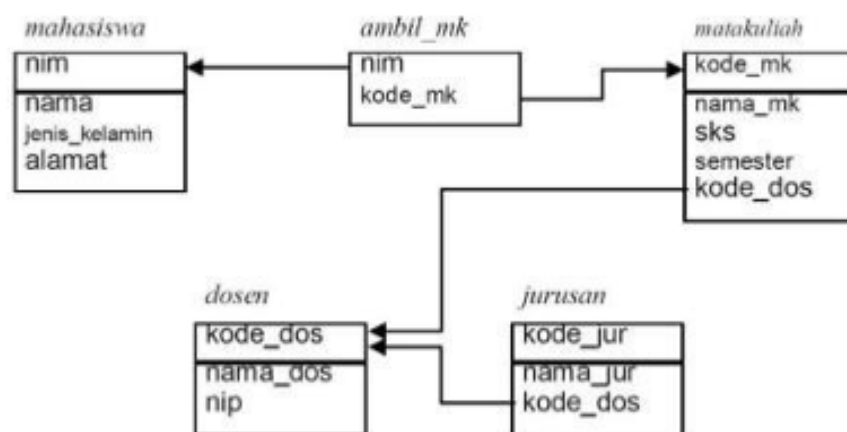
Bahan :

1. Operating System Linux (Ubuntu)
2. File Materi Praktikum
3. Terminal
4. Aplikasi Kantor Libre Office

#### B. Langkah kerja

Dalam latihan ini digunakan tiga buah tabel meliputi mahasiswa, ambil\_mk, matakuliah, dosen dan jurusan. Untuk itu, terlebih dahulu buat beberapa tabel seperti pada module.

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



#### III.2.1. SCALAR

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

### III.2.2. 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 DataTerdistribusi	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)

### III.2.3. Multiple-ColumnSubquery

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

#### III.2.4. 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. EXIST

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 EXIST

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

#### IV. Soal

1. Dapatkan kode dan nama matakuliah dosen yang menjadi Ketua Jurusan Teknik Elektro.

```
SELECT mk.kd_mk, mk.nama_mk FROM mata_kuliah mk JOIN dosen d ON
mk.kode_dos = d.kode_dos JOIN jurusan j ON d.kode_dos = j.kode_dos
WHERE j.nama_jur = 'Teknik Elektro';
```

Gambar 4.1

- Memilih kolom `kd\_mk` dan `nama\_mk` dari tabel `mata\_kuliah`.
- Menggabungkan `mata\_kuliah` dengan `dosen` berdasarkan `kode\_dos`.
- Menggabungkan `dosen` dengan `jurusan` berdasarkan `kode\_dos`.
- Menyaring data untuk hanya menampilkan mata kuliah yang diajarkan oleh dosen Ketua Jurusan Teknik Elektro (`j.nama\_jur = 'Teknik Elektro'`).

2. Dapatkan data mahasiswa yang tidak mengambil matakuliah.

```
SELECT m.nim, m.nama, m.jk, m.alamat FROM mahasiswa m WHERE NOT
EXISTS ( SELECT 1 FROM ambil_mk am WHERE m.nim = am.nim );
```

Gambar 4.2

- Memilih kolom `nim`, `nama`, `jk` (jenis kelamin), dan `alamat` dari tabel `mahasiswa`.
- Kondisi untuk memeriksa bahwa tidak ada entri di tabel `ambil\_mk` yang memiliki `nim` yang sama dengan `nim` di tabel `mahasiswa`.
- `SELECT 1 FROM ambil\_mk am WHERE m.nim = am.nim` memeriksa apakah ada entri di `ambil\_mk` untuk mahasiswa tertentu.
- Memastikan mahasiswa tersebut tidak memiliki entri di tabel `ambil\_mk`, artinya mahasiswa tersebut tidak mengambil mata kuliah apapun.



3. Dapatkan data dosen yang mengajar matakuliah diatas semester 3.

```
SELECT DISTINCT d.kode_dos, d.nama_dos, d.alamat_dos FROM dosen d
JOIN mata_kuliah mk ON d.kode_dos = mk.kode_dos WHERE mk.semester
> 3;
```

Gambar 4.3

- Memilih kolom `kode\_dos`, `nama\_dos`, dan `alamat\_dos` dari tabel `dosen`.
- Menggunakan `JOIN` untuk menggabungkan tabel `dosen` dengan tabel `mata\_kuliah` berdasarkan kolom `kode\_dos`.
- Kondisi `WHERE` digunakan untuk memfilter baris yang hanya memiliki nilai semester di atas 3 dari tabel `mata\_kuliah`.

4. Dapatkan data matakuliah dosen yang bukan merupakan Ketua Jurusan Teknik Elektro.

```
SELECT mk.kd_mk, mk.nama_mk, mk.sks, mk.semester, mk.kode_dos FROM
mata_kuliah mk JOIN dosen d ON mk.kode_dos = d.kode_dos WHERE
d.kode_dos NOT IN ( SELECT j.kode_dos FROM jurusan j WHERE
j.nama_jur = 'Teknik Elektro' );
```

Gambar 4.4

- Memilih kolom `kd\_mk`, `nama\_mk`, `sks`, `semester`, dan `kode\_dos` dari tabel `mata\_kuliah`.
- Menggunakan `JOIN` untuk menggabungkan tabel `mata\_kuliah` dengan tabel `dosen` berdasarkan kolom `kode\_dos`.
- Menggunakan subquery untuk memfilter baris dengan menggunakan kondisi `NOT IN`, di mana kode dosen yang bukan termasuk dalam daftar dosen yang terkait dengan jurusan 'Teknik Elektro' dari tabel `jurusan`.

5. Dapatkan data dosen pengajar matakuliah yang tidak diambil oleh mahasiswa.

```
SELECT DISTINCT d.kode_dos, d.nama_dos, d.alamat_dos FROM dosen d
JOIN mata_kuliah mk ON d.kode_dos = mk.kode_dos WHERE mk.kd_mk NOT
IN ( SELECT am.kd_mk FROM ambil_mk am );
```

Gambar 4.5

- Memilih kolom `kode\_dos`, `nama\_dos`, dan `alamat\_dos` dari tabel `dosen`.
  - Menggunakan `JOIN` untuk menggabungkan tabel `dosen` dengan tabel `mata\_kuliah` berdasarkan kolom `kode\_dos`.
  - Menggunakan subquery untuk memfilter baris dengan menggunakan kondisi `NOT IN`, di mana kode mata kuliah yang tidak termasuk dalam daftar kode mata kuliah yang diambil oleh mahasiswa dari tabel `ambil\_mk`.
6. Dapatkan data dosen yang mengajar matakuliah dengan sks lebih besar dari sembarang sks.

```
SELECT DISTINCT d.kode_dos, d.nama_dos, d.alamat_dos FROM dosen d
JOIN mata_kuliah mk ON d.kode_dos = mk.kode_dos WHERE mk.sks > ANY
( SELECT sks FROM mata_kuliah );
```

Gambar 4.6

- Memilih kolom `kode\_dos`, `nama\_dos`, dan `alamat\_dos` dari tabel `dosen`.
  - Menggunakan `JOIN` untuk menggabungkan tabel `dosen` dengan tabel `mata\_kuliah` berdasarkan kolom `kode\_dos`.
  - Menggunakan subquery untuk memfilter baris dengan menggunakan kondisi `> ANY`, di mana jumlah SKS mata kuliah yang diajar oleh dosen lebih besar dari salah satu nilai SKS dari semua mata kuliah.
7. Dapatkan data mahasiswa yang tinggal satu wilayah dengan dosen yang bukan merupakan Ketua Jurusan Teknik Elektro.

```
SELECT m.nim, m.nama, m.jk, m.alamat FROM mahasiswa m WHERE
m.alamat IN ( SELECT d.alamat_dos FROM dosen d WHERE d.kode_dos
NOT IN ( SELECT j.kode_dos FROM jurusan j WHERE j.nama_jur =
'Teknik Elektro' ) );
```

Gambar 4.7

- Memilih kolom `nim`, `nama`, `jk` (jenis kelamin), dan `alamat` dari tabel `mahasiswa`.
- Menggunakan kondisi `WHERE` untuk memfilter baris berdasarkan alamat mahasiswa yang ada dalam hasil subquery.

- Subquery digunakan untuk mendapatkan alamat dosen dari tabel `dosen`, yang mana kode dosen tidak termasuk dalam daftar dosen yang terkait dengan jurusan 'Teknik Elektro' dari tabel `jurusan`.

8. Dapatkan data mahasiswa yang diajar oleh Ketua Jurusan Teknik Elektro.

```
SELECT m.nim, m.nama, m.jk, m.alamat FROM mahasiswa m JOIN
ambil_mk am ON m.nim = am.nim JOIN mata_kuliah mk ON am.kd_mk =
mk.kd_mk JOIN dosen d ON mk.kode_dos = d.kode_dos JOIN jurusan j
ON d.kode_dos = j.kode_dos WHERE j.nama_jur = 'Teknik Elektro';
```

Gambar 4.8

- Memilih kolom `nim`, `nama`, `jk` (jenis kelamin), dan `alamat` dari tabel `mahasiswa`.
- Menggunakan `JOIN` untuk menggabungkan tabel `mahasiswa` dengan tabel `ambil\_mk` berdasarkan kolom `nim`, kemudian menggabungkan hasilnya dengan tabel `mata\_kuliah` berdasarkan kolom `kd\_mk`, dan menggabungkan lagi dengan tabel `dosen` berdasarkan kolom `kode\_dos`.
- Selanjutnya, melakukan `JOIN` dengan tabel `jurusan` untuk memastikan bahwa dosen yang mengajar mata kuliah tersebut adalah dari jurusan 'Teknik Elektro'.
- Kondisi `WHERE` digunakan untuk memfilter baris di mana nama jurusan yang terhubung dengan dosen adalah 'Teknik Elektro'.

## V. Kesimpulan

Memahami subquery dengan baik meningkatkan kemampuan mengoptimalkan pengambilan data dalam database relasional. Subquery adalah alat penting yang memungkinkan Anda menulis kueri lebih kompleks dan mendapatkan wawasan lebih mendalam tentang data. Menguasai subquery sangat berguna untuk analisis dan manipulasi data yang kompleks, meningkatkan fleksibilitas dan efisiensi dalam pengembangan aplikasi database.

## **VI. Referensi**

Modul