

# **LAPORAN PRAKTIKUM BASIS DATA**

## **SQL SUB QUERY**



**DISUSUN OLEH :**

Restu Wibisono

2340506061

**PROGRAM STUDI S1 TEKNOLOGI INFORMASI**

**JURUSAN TEKNIK ELEKTRO FAKULTAS TEKNIK**

**UNIVERSITAS TIDAR**

**2024**

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



<b>Diisi Mahasiswa Praktikan</b>								
Nama Praktikan	Restu Wibisono							
NPM	2340506061							
Rombel	03							
Judul Praktikum	SQL SUB QUERY							
Tanggal Praktikum	25 April 2024							
<b>Diisi Asisten Praktikum</b>								
Tanggal Pengumpulan								
Catatan								

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

**PROGRAM STUDI S1 TEKNOLOGI INFORMASI**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS TIDAR**  
**Genap 2023/2024**

## 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)
```

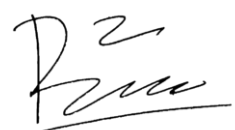
Subquery digunakan untuk mencapai tujuan tertentu dalam pengambilan data atau manipulasi data yang tidak dapat dicapai dengan menggunakan query tunggal. Dengan subquery, pengguna dapat menggabungkan data dari beberapa tabel, melakukan perbandingan antara hasil query, atau membuat pengambilan data yang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

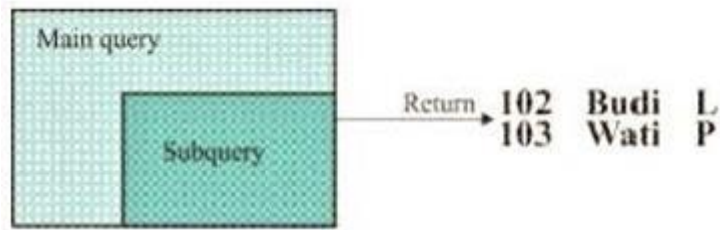
Subquery dapat disisipkan ke dalam klausa WHERE, FROM, atau SELECT dari query utama. Dalam klausa WHERE, subquery digunakan untuk memfilter baris berdasarkan kriteria tertentu. Dalam klausa FROM, subquery digunakan untuk menentukan sumber data untuk query utama. Sedangkan dalam klausa SELECT, subquery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

Terdapat beberapa tipe subquery yaitu :

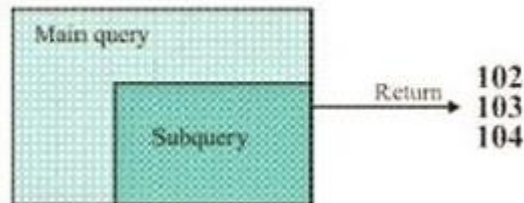
1. Multiple column Subquery: Subquery kolom ganda (multiple-column) mengembalikan lebih dari satu baris dan satu kolom data.

Tanda Tangan



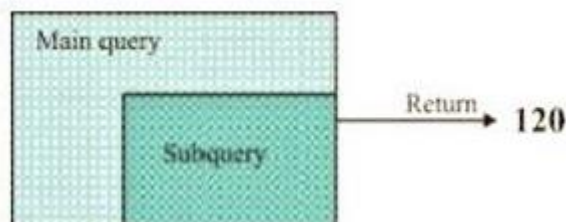


2. Multiple-row Subquery: Subquery ini menghasilkan beberapa baris data.



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

3. Scalar Subquery: Subquery ini menghasilkan nilai tunggal, seperti hasil perhitungan atau pengambilan data tunggal.



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

Salah satu contoh umum dari subquery adalah penggunaannya dalam klausa WHERE untuk memfilter hasil query berdasarkan kriteria yang berasal dari hasil query lainnya. Misalnya, subquery dapat digunakan untuk mencari pelanggan yang melakukan pembelian diatas rata-rata total pembelian. Contoh lainnya adalah penggunaan subquery dalam klausa SELECT untuk menghasilkan nilai agregat, seperti jumlah total penjualan per kategori produk.

Keuntungan utama dari penggunaan subquery adalah kemampuannya untuk menangani tugas-tugas yang kompleks dengan logika yang lebih rumit dalam manipulasi data. Namun, penggunaan subquery yang berlebihan dapat menyebabkan penurunan kinerja dan kompleksitas yang sulit dipelihara dalam kode SQL.

Dengan pemahaman yang baik tentang subquery, pengguna dapat mengoptimalkan kinerja dan fleksibilitas dalam pengambilan data dan manipulasi data

Tanda Tangan

dalam basis data relasional. Subquery adalah alat yang berguna dalam pengembangan aplikasi database yang memungkinkan pengguna untuk mengeksekusi query yang lebih kompleks dan mendapatkan wawasan yang lebih dalam tentang data mereka.

### III. Metode Praktikum

#### A. Alat dan bahan

Alat :

1. PC (Komputer)
2. Keyboard
3. Mouse

Bahan :

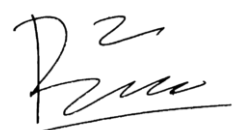
1. Operating System Windows 10
2. File Materi Praktikum
3. Aplikasi Paket Web server XAMPP
4. Aplikasi Kantor (Microsoft Office/Libre Office/WPS Office/etc)

#### 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 dengan struktur sebagai berikut.

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

Tanda Tangan



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
TIK123	Jaringan Komputer	2	5	33
TIK333	Sistem Operasi	3	5	10
TIK342	Praktikum Basis Data	1	3	11

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

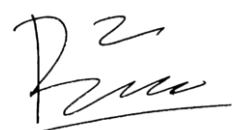
*Gambar 6. Tabel jurusan*

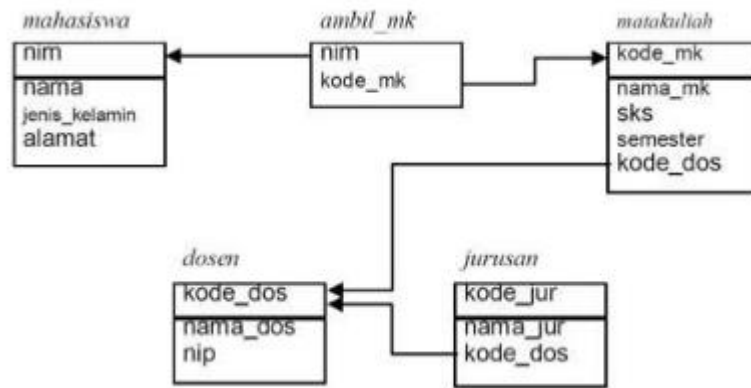
kode_dos	nama_dos	alamat_dos
10	Suharto	Jl. Jombang
11	Martono	Jl. Kalpataru
12	Rahmawati	Jl. Jakarta
13	Bambang	Jl. Bandung
14	Nurul	Jl. Raya Tidar

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.

Tanda Tangan





### 3.2.1. 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

### 3.2.2. 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.

Tanda Tangan

```
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.

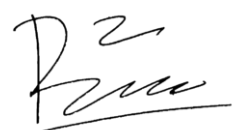
Operator < ANY ekuivalen dengan MAX (kurang dari maks).

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

Tanda Tangan





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)

### 3.2.3. 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	3	11
TIK342	Praktikum Basis Data 1	3	3	11

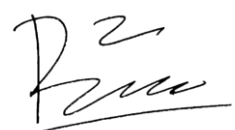
### 3.2.4. Operator EXISTS dan NOT EXISTS

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.

Tanda Tangan



```

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)

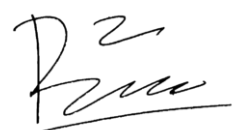
```

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.

Tanda Tangan



✓ Menampilkan baris 0 - 5 (total 6, Pencarian dilakukan dalam 0,0003 detik.)

```
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';
```

☐ Profil [ [Edit dikotak](#) ] [ [Ubah](#) ] [ [Jelaskan SQL](#) ] [ [Buat kode PHP](#) ] [ [Segarkan](#) ]

☐ Tampilkan semua | Jumlah baris: 25 ▾

Extra options

kd_mk	nama_mk
PTI333	Basis Data Terdistribusi
TIK333	Sistem Operasi
PTI333	Basis Data Terdistribusi
TIK333	Sistem Operasi
PTI333	Basis Data Terdistribusi
TIK333	Sistem Operasi

(Gambar 4.1.1)

- 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.

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0003 detik.)

```
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 );
```

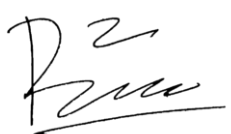
☐ Profil [ [Edit dikotak](#) ] [ [Ubah](#) ] [ [Jelaskan SQL](#) ] [ [Buat kode PHP](#) ] [ [Segarkan](#) ]

nim	nama	jk	alamat
-----	------	----	--------

(Gambar 4.2.1)

- Memilih kolom `nim`, `nama`, `jk` (jenis kelamin), dan `alamat` dari tabel `mahasiswa`.

Tanda Tangan



- 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.

✓ Menampilkan baris 0 - 2 (total 3, Pencarian dilakukan dalam 0,0002 detik.)

```
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;
```

☐ Profil [ [Edit kotak](#) ] [ [Ubah](#) ] [ [Jelaskan SQL](#) ] [ [Buat kode PHP](#) ] [ [Segarkan](#) ]

☐ Tampilkan semua | Jumlah baris: 25 ▾

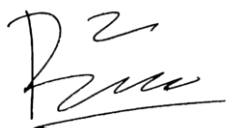
Extra options

kode_dos	nama_dos	alamat_dos
12	Rahmawati	0
10		0
10	Suharto	0

(Gambar 4.3.1)

- 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.

Tanda Tangan



✓ Menampilkan baris 0 - 2 (total 3, Pencarian dilakukan dalam 0,0004 detik.)

```
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' );
```

☐ Profil [ [Edit kotak](#) ] [ [Ubah](#) ] [ [Jelaskan SQL](#) ] [ [Buat kode PHP](#) ] [ [Segarkan](#) ]

☐ Tampilkan semua | Jumlah baris: 25 ▾

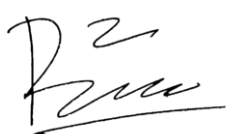
Extra options

kd_mk	nama_mk	sks	semester	kode_dos
PTI447	Praktikum Basis Data	1	3	11
TIK342	Praktikum Basis Data	1	3	11
PTI123	Grafika Multimedia	3	5	12

(Gambar 4.4.1)

- 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.

Tanda Tangan



✓ Menampilkan baris 0 - 0 (total 1, Pencarian dilakukan dalam 0,0009 detik.)

```
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 );
```

☐ Profil [ [Edit dikotak](#) ] [ [Ubah](#) ] [ [Jelaskan SQL](#) ] [ [Buat kode PHP](#) ] [ [Segarkan](#) ]

☐ Tampilkan semua | Jumlah baris: 25 ▾

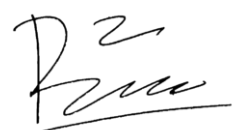
Extra options

kode_dos	nama_dos	alamat_dos
11	Martono	0

(Gambar 4.5.1)

- 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.

Tanda Tangan



✓ Menampilkan baris 0 - 2 (total 3, Pencarian dilakukan dalam 0,0004 detik.)

```
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 );
```

☐ Profil [ [Edit dikotak](#) ] [ [Ubah](#) ] [ [Jelaskan SQL](#) ] [ [Buat kode PHP](#) ] [ [Segarkan](#) ]

☐ Tampilkan semua | Jumlah baris: 25 ▾

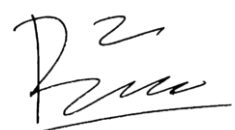
Extra options

kode_dos	nama_dos	alamat_dos
12	Rahmawati	0
10		0
10	Suharto	0

(Gambar 4.6.1)

- 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.

Tanda Tangan



✓ Menampilkan baris 0 - 6 (total 7, Pencarian dilakukan dalam 0,0004 detik.)

```
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' ) );
```

☐ Profil [ [Edit dikotak](#) ] [ [Ubah](#) ] [ [Jelaskan SQL](#) ] [ [Buat kode PHP](#) ] [ [Segarkan](#) ]

☐ Tampilkan semua | Jumlah baris: 25 ▾

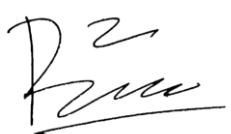
Extra options

nim	nama	jk	alamat
101	Arif	L	Jl. Kenanga
102	Budi	L	Jl. Jombang
103	Wati	P	Jl. Surabaya
104	Ika	P	Jl. Jombang
105	Tono	L	Jl. Jakarta
106	Iwan	L	Jl. Malang
107	Sari	P	Jl. Malang

(Gambar 4.7.1)

- 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.

Tanda Tangan





✓ Menampilkan baris 0 - 5 (total 6, Pencarian dilakukan dalam 0,0003 detik.)

```
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';
```

☐ Profil [ [Edit kotak](#) ] [ [Ubah](#) ] [ [Jelaskan SQL](#) ] [ [Buat kode PHP](#) ] [ [Segarkan](#) ]

☐ Tampilkan semua | Jumlah baris: 25 ▾

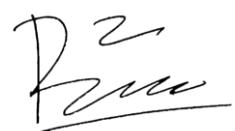
Extra options

nim	nama	jk	alamat
102	Budi	L	Jl. Jombang
106	Iwan	L	Jl. Malang
102	Budi	L	Jl. Jombang
106	Iwan	L	Jl. Malang
102	Budi	L	Jl. Jombang
106	Iwan	L	Jl. Malang

(Gambar 4.8.1)

- 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'.

Tanda Tangan



## V. Kesimpulan

Pemahaman yang baik tentang subquery akan meningkatkan kemampuan Anda untuk mengoptimalkan kinerja dan fleksibilitas pengambilan data dalam database relasional. Subquery adalah alat penting dalam pengembangan aplikasi database, memungkinkan Anda melakukan kueri yang lebih kompleks dan mendapatkan pemahaman yang lebih baik tentang data yang ada. Menguasai subquery bisa sangat berguna saat melakukan berbagai tugas analisis dan manipulasi data yang kompleks di dunia nyata.

## VI. Referensi

Van Rossum, G. (2003). An introduction to Python (p. 115). F. L. Drake (Ed.).

Bristol: Network Theory Ltd..

<http://atk.fam.free.fr/fichiers/stage/Python/JF/site/pytut.pdf>

Kuhlman, D. (2009). A python book: Beginning python, advanced python, and python exercises (pp. 1-227). Lutz: Dave Kuhlman.

[https://www.davekuhlman.org/python\\_book\\_01.pdf](https://www.davekuhlman.org/python_book_01.pdf)

Python, W. (2021). Python. Python Releases for Windows, 24.

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=1f2ee3831eebfc97bfafd514ca2abb7e2c5c86bb>

Tanda Tangan

