

MODUL 7 – SQL SUB QUERY

1.1. CAPAIAN PEMBELAJARAN

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

1.2. ALAT DAN BAHAN

1. Seperangkat komputer lengkap/Laptop dengan koneksi internet
2. Sistem Operasi Windows/Mac/Linux
3. Aplikasi Paket Web server XAMPP
4. Aplikasi Kantor (Microsoft Office/Libre Office/WPS Office/etc)

1.3. 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.

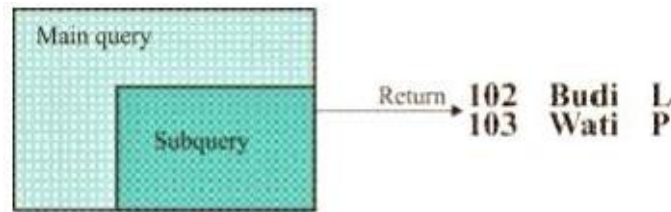
```
SELECT A1, A2, ..., An
FROM r1, r2, r3, ..., rm
WHERE P
      (SELECT A1, A2, ..., An
       FROM r1, r2, r3, ..., rm
       WHERE P)
```

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.

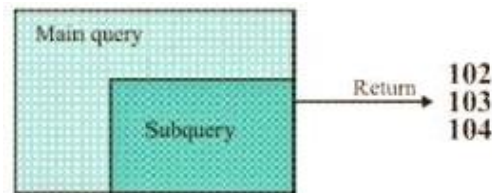
Terdapat beberapa tipe subquery yaitu :

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



Gambar 1. Multiple column Subquery

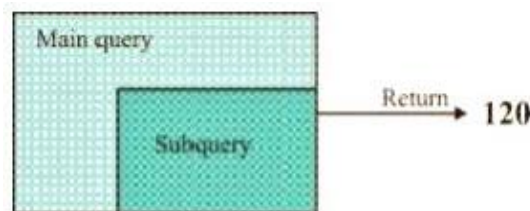
2. Multiple-row Subquery: Subquery ini menghasilkan beberapa baris data.



Gambar 2. multiple-row subquery

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.



Gambar 3. scalar sub query

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 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.

1.5. PRAKTIKUM PENERAPAN SQL BERTINGKAT/SUB QUERY

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

Gambar 4. Tabel mahasiswa

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

Gambar 5. tabel matakuliah

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

Gambar 6. Tabel jurusan

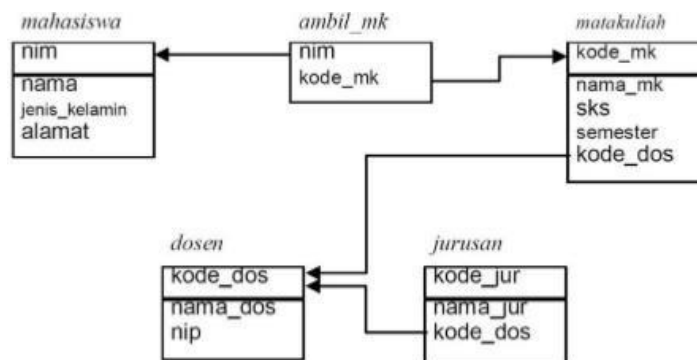
kode_dos	nama_dos	alamat_dos
10	Suharto	Jl. Jombang
11	Martono	Jl. Kalpataru
12	Rahmawati	Jl. Jakarta
13	Bambang	Jl. Bandung
14	Nurul	Jl. Raya Tidar

Gambar 7. Tabel dosen

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

Gambar 8. Tabel ambil_mk

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



Gambar 9. diagram relasi

1.5.1. 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

1.5.2. 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)

1.5.3. 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

1.5.4. 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

1.6. TUGAS MODUL 7

1.6.1. Soal

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

1.6.2. Petunjuk Pengerjaan

a) Tugas:

- Tugas membuat perintah SQL sesuai pada soal
- Tuliskan perintah dan screenshot hasil perintah.

b) Laporan:

- Buatlah laporan akhir berdasarkan diagram yang Anda buat.
- Laporan dibuat sesuai format dan **diketik**.

- Masukkan langkah-langkah pengerjaan tugas ke dalam laporan dalam bentuk screenshot dan penjelasan.
- **Cantumkan tanda tangan** Anda di setiap halaman dokumen laporan.
- Jika di dalam laporan ada gambar atau *screenshot* yang ingin ditampilkan, bisa ditempelkan pada halaman(menyesuaikan)
- Laporan disimpan dalam bentuk pdf
- Penamaan *file* pdf : " **LaporanModul7_DBMS_NPM.pdf**"

c) Pengumpulan:

- *File* yang dikumpulkan yaitu:
 - Laporan : **TugasModul7_DBMS_NPM.pdf**
- **Batas Pengumpulan:** Sebelum Pertemuan Praktik Ke 11.