

TUGAS MODUL 5
LAPORAN PRAKTIKUM BASIS DATA
“SUBQUERY”



Disusun oleh:

Nama : Antonius Munthe

NIM : 121140032

Kelas : Basis Data RA

PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN
2023

BAB I

TEORI DASAR

SubQuery

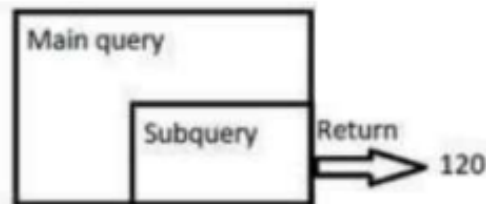
Subquery (disebut juga subselect atau nested select) adalah query SELECT yang ada di dalam perintah SQL lain misalnya SELECT, INSERT, UPDATE, atau DELETE. Sebagai contoh, misal terdapat pernyataan sebagai berikut :

“DAPATKAN DATA MAHASISWA YANG IPNYA LEBIH DARI MAHASISWA DENGAN NIM 14117149”

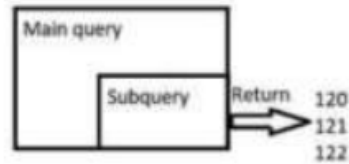
Secara normal, diperlukan dua tahapan untuk menyelesaikan kasus diatas. Pertama adalah mendapatkan IP dari mahasiswa yang memiliki nim 14117149. Langkah selanjutnya, baru kita bisa mengetahui data mahasiswa yang IP lebih dari mahasiswa dengan nim 14117149. Salah satu contoh sintaks subquery dapat dilihat pada Gambar:

```
MariaDB [kampus]> SELECT Nama, IP  
-> FROM Mahasiswa  
-> WHERE IP > SELECT IP  
-> FROM Mahasiswa  
-> WHERE NIM="14117149";
```

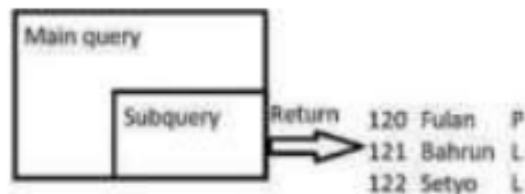
Subquery diklasifikasikan ke dalam tiga jenis : scalar, multiple-row, dan multiple-column.
A. Scalar Subquery Subquery baris tunggal (scalar) hanya mengembalikan hasil sat baris data. Bentuk subquery ini dipelihatkan seperti Gambar:



- B. Multiple row subquery Subquery baris multiple row mengembalikan lebih dari satu baris data. Bentuk subquery ini diperlihatkan seperti Gambar:



- C. Multiple-Column Subquery Subquery multiple column mengembalikan lebih dari satu baris dan satu kolom data. Bentuk subquery ini dapat dilihat seperti Gambar:



Beberapa aturan pada subquery :

- Subqueries harus tertutup dalam tanda kurung.
- Sebuah subquery hanya dapat memiliki satu kolom pada klausa SELECT, kecuali beberapa kolom yang di query utama untuk subquery untuk membandingkan kolom yang dipilih.
- ORDER BY tidak dapat digunakan dalam subquery, meskipun permintaan utama dapat menggunakan ORDER BY. GROUP BY dapat digunakan untuk melakukan fungsi yang sama seperti ORDER BY dalam subquery.
- Subqueries yang kembali lebih dari satu baris hanya dapat digunakan dengan beberapa value operator, seperti operator IN.
- Daftar SELECT tidak bisa menyertakan referensi ke nilai-nilai yang mengevaluasi ke BLOB, ARRAY, CLOB, atau NCLOB.
- Sebuah subquery tidak dapat segera tertutup dalam fungsi set.
- Operator BETWEEN tidak dapat digunakan dengan subquery; Namun, BETWEEN dapat digunakan dalam subquery.

BAB II

ANALISIS DAN PEMBAHASAN

1. Tampilkan nama produk dan jumlah stock yang di supply oleh perusahaan

Suka Maju

Dalam subquery ini, kita menggabungkan data dari tabel suppliers dan melakukan pengelompokan berdasarkan supplier_id untuk mencari ID supplier yang sesuai dengan nama perusahaan "Suka Maju". Kemudian di luar subquery, kita melakukan join dengan tabel produk untuk mendapatkan data produk yang disuplai oleh perusahaan tersebut. Dengan menggunakan sintaks subquery seperti ini, kita dapat mengambil data dari beberapa tabel sekaligus dan melakukan analisis data yang lebih kompleks.

Perintah: “SELECT column_name FROM table1 WHERE column_name IN (SELECT column_name [,column_name] FROM table2 [WHERE kondisi]);”.

```
MariaDB [northwind]> select produk_nama,jumlah_stok
-> from produk
-> where suppliers_id in (select suppliers_id from suppliers where company_nama="Suka Maju");
```

produk_nama	jumlah_stok
The Kotak 300 ml	40 pcs
Rojo Lele 5 kg	60 pcs

```
2 rows in set (0.173 sec)
```

2. Tampilkan seluruh data produk yang nama kontak supplier nya Rahmat

Dalam perintah SQL di atas, kita menggunakan INNER JOIN untuk menggabungkan data dari tabel products dan suppliers berdasarkan kolom supplier_id. Kemudian, kita menambahkan kondisi WHERE untuk memfilter data supplier berdasarkan nama perusahaan 'Rahmat'. Dengan menggunakan sintaks subquery seperti ini, kita dapat mengambil data dari beberapa tabel sekaligus dan melakukan analisis data yang lebih kompleks.

Perintah: “SELECT column_name FROM table1 WHERE column_name IN (SELECT column_name FROM table2 [WHERE kondisi]);”.

```
MariaDB [northwind]> Select * from produk
-> where suppliers_id in (select suppliers_id from suppliers where nama_kontak="Rahmat");
```

produk_id	produk_nama	jumlah_stok	suppliers_id
P109	The Kotak 300 ml	40 pcs	S002
P441	Rojo Lele 5 kg	60 pcs	S002

```
2 rows in set (0.001 sec)
```

3. Tampilkan Transaksi yang dilakukan diatas tanggal 15 september dilayani oleh

Siska dan di supply oleh Surya Kun

Dalam perintah SQL di atas, kita menggunakan INNER JOIN untuk menggabungkan data dari tabel products dan suppliers berdasarkan kolom supplier_id. Kemudian, kita menambahkan kondisi WHERE untuk memfilter data supplier berdasarkan nama perusahaan 'Rahmat'. Dengan menggunakan sintaks subquery seperti ini, kita dapat mengambil data dari beberapa tabel sekaligus dan melakukan analisis data yang lebih kompleks.

perintah “SELECT column_name FROM table1 WHERE column_name and column_name = (SELECT column_name FROM table2 [WHERE kondisi]) and column_name IN (SELECT column_name FROM table3 [WHERE kondisi] = (SELECT column_name FROM table4 [WHERE kondisi]);”.

```
MariaDB [(none)]> use northwind;
Database changed
MariaDB [northwind]> select * from transaksi where tgl_transaksi > "2017-09-15"
-> and id_pegawai = (select id_pegawai from pegawai where pegawai_nama = "Siska")
-> and produk_id in (select produk_id from produk where suppliers_id = (select suppliers_id
-> from suppliers
-> where company_nama = "Surya Kun"));
+-----+-----+-----+-----+-----+-----+
| Id_transaksi | Id_pembeli | Id_pegawai | Produk_id | Tgl_transaksi | Jumlah_beli |
+-----+-----+-----+-----+-----+-----+
|          6 | C_810      | Pg_002     | P123      | 2017-09-20    | 10 pcs      |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.599 sec)
```

4. Tampilkan seluruh nama Perusahaan yang jumlah pembeliannya dalam tabel transaksi dibawah 3pcs.

Dalam perintah SQL di atas, kita menggunakan subquery untuk mengambil data dari tabel transactions dan melakukan pengelompokan berdasarkan produk_id. Kemudian, kita menggabungkan data dari tabel products dan melakukan pengelompokan berdasarkan supplier_id. Terakhir, kita melakukan join dengan tabel suppliers untuk mendapatkan nama perusahaan. Dengan menggunakan sintaks subquery seperti ini, kita dapat melakukan analisis data yang lebih kompleks dan mendapatkan hasil yang lebih spesifik.

Perintah: “SELECT column_name FROM table1 WHERE column_name IN (SELECT column_name FROM table2 [WHERE kondisi] IN (SELECT column_name FROM table3 [WHERE kondisi]));”.

```
MariaDB [northwind]> select company_nama from suppliers where suppliers_id in (select suppliers_id
-> from produk
-> where produk_id in (select produk_id
-> from transaksi
-> where jumlah_beli < 3));
+-----+
| company_nama |
+-----+
| Ali          |
+-----+
1 row in set, 6 warnings (0.029 sec)
```

5. Tampilkan semua data customer yang dilayani oleh Santi

Untuk menampilkan seluruh data pelanggan yang dilayani oleh Santi, digunakan sintaks subquery dengan perintah "SELECT column_name FROM table1 WHERE column_name IN (SELECT column_name FROM table2 [WHERE kondisi] IN (SELECT column_name FROM table3 [WHERE kondisi]));". Di dalam subquery, data dari tabel pegawai digabungkan dan dikelompokkan berdasarkan pegawai_id. Selanjutnya, data dari tabel transaksi digabungkan dan dikelompokkan berdasarkan id_pembeli. Kemudian, di luar subquery, dilakukan join dengan tabel pelanggan untuk mendapatkan data pelanggan.

Perintah: "SELECT column_name FROM table1 WHERE column_name IN (SELECT column_name FROM table2 [WHERE kondisi] IN (SELECT column_name FROM table3 [WHERE kondisi]));".

```
MariaDB [northwind]> select * from pembeli where id_pembeli in (select id_pembeli from transaksi where id_pegawai in (select id_pegawai from pegawai where pegawai_nama = 'Santi'));
```

id_pembeli	pembeli_nama	pembeli_kontak
C_800	Egi	0812521221
C_810	Ardi	0862145121

```
2 rows in set (0.001 sec)
```

6. Tampilkan kontak pembeli yang membeli 3pcs barang dan dilayani oleh Siska

Untuk menampilkan kontak pembeli yang membeli 3 pcs barang dan dilayani oleh Siska kita dapat menggunakan syntaks subquery dengan perintah "SELECT column_name FROM table1 WHERE column_name IN (SELECT column_name FROM table2 [WHERE kondisi] IN (SELECT column_name FROM table3 [WHERE kondisi]));". Di dalam subquery, kita menggabungkan data dari tabel transaksi dan melakukan pengelompokan berdasarkan id_pembeli. Kemudian kita menggabungkan data dari tabel pegawai dan melakukan pengelompokan berdasarkan id_pegawai. Selanjutnya, di luar subquery, kita melakukan join dengan tabel pembeli untuk mendapatkan data kontak pembeli yang sesuai dengan kondisi yang diminta.

Perintah: "SELECT column_name FROM table1 WHERE column_name IN (SELECT column_name FROM table2 [WHERE kondisi] IN (SELECT column_name FROM table3 [WHERE kondisi]));".

```
MariaDB [northwind]> select pembeli_kontak from pembeli where id_pembeli in (select id_pembeli from transaksi where jumlah_beli = 3 and id_pegawai in (select id_pegawai from pegawai where pegawai_nama = 'Siska'));
```

pembeli_kontak
081231511

```
1 row in set, 6 warnings (0.133 sec)
```

7. Tampilkan kontak pembeli yang membeli 3pcs barang, tidak dilayani oleh Siska, dan barangnya di supply oleh Surya Kun

Untuk menampilkan kontak pembeli yang membeli 3 pcs barang, tidak dilayani oleh Siska, dan barangnya disuplai oleh Surya Kun, kita menggunakan syntaks subquery. Dengan perintah "SELECT column_name FROM table1 WHERE column_name IN (SELECT column_name FROM table2 [WHERE kondisi] != (SELECT column_name FROM table3 [WHERE kondisi] AND (SELECT column_name FROM table4 [WHERE kondisi]) AND (SELECT column_name FROM table5 [WHERE kondisi])));". Di dalam subquery pertama, kita menggabungkan data dari tabel transaksi dan melakukan pengelompokan berdasarkan produk_id. Kemudian, di subquery kedua, kita menggabungkan data dari tabel pegawai dan melakukan pengelompokan berdasarkan id_pegawai. Kemudian, di subquery ketiga, kita menggabungkan data dari tabel suppliers dan melakukan pengelompokan berdasarkan suppliers_id. Kemudian di luar subquery, kita melakukan join dengan tabel produk, pegawai, supplier, dan transaksi untuk mendapatkan data pembeli.

```
MariaDB [northwind]> select pembeli_kontak from pembeli where id_pembeli in ( select id_pembeli
-> from transaksi
-> where jumlah_beli = 3
-> and id_pegawai != (select id_pegawai
-> from pegawai
-> where pegawai_nama = "Siska")
-> and produk_id in (select produk_id
-> from produk
-> where suppliers_id = (select suppliers_id
-> from suppliers
-> where company_nama = "Surya Kun")));
+-----+
| pembeli_kontak |
+-----+
| 085212021111   |
+-----+
1 row in set, 6 warnings (0.087 sec)
```

8. Tampilkan kontak pembeli yang membeli diatas 3 pcs barang dan transaksi dilakukan tanggal 15 September

Untuk menampilkan informasi kontak pembeli yang melakukan pembelian lebih dari 3 pcs barang dan transaksi dilakukan pada tanggal 15 September, dapat digunakan syntaks subquery dengan perintah "SELECT column_name FROM table1 WHERE column_name IN (SELECT column_name FROM table2 [WHERE kondisi]);". Pada subquery, data dari tabel transaksi digabungkan dan dikelompokkan berdasarkan id_pembeli. Kemudian pada query utama dilakukan join dengan tabel pembeli untuk mendapatkan informasi kontak pembeli tersebut.

```
MariaDB [northwind]> select pembeli_kontak from pembeli where id_pembeli in
(select id_pembeli from transaksi where jumlah_beli > 3 and tgl_transaksi
= '2017-09-15');
+-----+
| pembeli_kontak |
+-----+
| 081231511      |
+-----+
1 row in set, 2 warnings (0.106 sec)
```

9. Tampilkan pada tanggal berapa saja barang dari perusahaan Ceria Kasih laku terjual

Perintah: “SELECT column_name FROM table1 WHERE column_name IN (SELECT column_name FROM table2 [WHERE kondisi] IN (SELECT column_name FROM table3 [WHERE kondisi]));”.

```
MariaDB [northwind]> select tgl_transaksi from transaksi where produk_id in (select`Produk_id`
-> from produk
-> where suppliers_id in (select suppliers_Id
-> from suppliers
-> where company_nama = "Ceria Kasih"));
Empty set (0.001 sec)
```

10. Tampilkan nama pelanggan yang pernah dilayani oleh Siska dan Nuri

Perintah: “SELECT column_name FROM table1 WHERE column_name IN (SELECT column_name FROM table2 [WHERE kondisi] IN (SELECT column_name FROM table3 [kondisi]));”.

```
MariaDB [northwind]> select pembeli_nama from pembeli where id_pembeli in (
select id_pembeli from transaksi where id_pegawai in (select id_pegawai fro
m pegawai where pegawai_nama in ('Nuri', 'Siska')));
+-----+
| pembeli_nama |
+-----+
| Ardi         |
| Rudi         |
+-----+
2 rows in set (0.001 sec)
```

11. Tampilkan nama pelanggan yang membeli barang dengan jumlah stock diatas 100 dan di supply oleh Ali

Perintah: “SELECT column_name FROM table1 WHERE column_name and column_name = (SELECT column_name FROM table2 [WHERE kondisi]) and column_name IN (SELECT column_name FROM table3 [WHERE kondisi] = (SELECT column_name FROM table4 [WHERE kondisi]));”.


```

MariaDB [(none)]> use northwind;
Database changed
MariaDB [northwind]> select pembeli_nama from pembeli where id_pembeli in (select id_pembeli
-> from transaksi
-> where produk_id in (select produk_id
-> from produk
-> where jumlah_stok > 100
-> and suppliers_Id = (select suppliers_Id
-> from suppliers
-> where company_nama = "ALI"));
+-----+
| pembeli_nama |
+-----+
| Egi          |
| Ardi         |
| Rudi         |
+-----+
3 rows in set, 6 warnings (0.381 sec)

```

12. Tampilkan data seluruh transaksi yang dilakukan oleh Andi dan dilayani oleh Jamal

Perintah: “SELECT column_name FROM table1 WHERE column_name IN (SELECT column_name FROM table2 [WHERE kondisi] IN (SELECT column_name FROM table3 [WHERE kondisi]));”.

```

MariaDB [northwind]> select * from transaksi where id_pembeli = (select id_pembeli from pembeli where pembeli_nama = 'Andi') and id_pegawai = (select id_pegawai from pegawai where pegawai_nama = 'Jamal');
+-----+-----+-----+-----+-----+-----+
| Id_transaksi | Id_pembeli | Id_pegawai | Produk_id | Tgl_transaksi | Jumlah_beli |
+-----+-----+-----+-----+-----+-----+
| 4            | C_991      | Pg_004     | P333      | 2017-09-18    | 3 pcs       |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.095 sec)

```

13. Tampilkan Nama produk dan jumlah stock yang di supply oleh Company Maju Terus

Perintah: “SELECT column_name FROM table1 WHERE column_name IN (SELECT column_name FROM table2 [WHERE kondisi]);”.

```

MariaDB [northwind]> select produk_nama, jumlah_stok from produk where suppliers_id = (select suppliers_id from suppliers where company_nama = 'Maju Terus');
+-----+-----+
| produk_nama | jumlah_stok |
+-----+-----+
| Millo 150 ml | 50 pcs      |
| Grand 320 ml | 400 pcs     |
+-----+-----+
2 rows in set (0.136 sec)

```

14. Tampilkan Nama Perusahaan dan kontakannya yang transaksi barang-barangnya dilayani oleh Yaya

Perintah: “SELECT column_name FROM table1 WHERE column_name and column_name = (SELECT column_name FROM table2 [WHERE kondisi]) and column_name IN (SELECT

column_name FROM table3 [WHERE kondisi] = (SELECT column_name FROM table4 [WHERE kondisi]);”.

```
MariaDB [northwind]> select company_nama, nama_kontak from suppliers where suppliers_id in (select suppliers_id
-> from produk
-> where produk_id in (select produk_id
-> from transaksi
-> where id_pegawai in (select id_pegawai
-> from pegawai
-> where pegawai_nama = "Yaya"));
Empty set (0.097 sec)
```

15. Tampilkan nama pegawai yang berhasil menjual lebih dari 5 pcs barang dalam satu transaksi

Perintah: “SELECT column_name FROM table1 WHERE column_name IN (SELECT column_name FROM table2 [WHERE kondisi]);”.

```
MariaDB [northwind]> select pegawai_nama from pegawai where id_pegawai in (select id_pegawai
-> from transaksi
-> where jumlah_beli > 5);
+-----+
| pegawai_nama |
+-----+
| Siska        |
| Nuri         |
+-----+
2 rows in set, 6 warnings (0.001 sec)
```

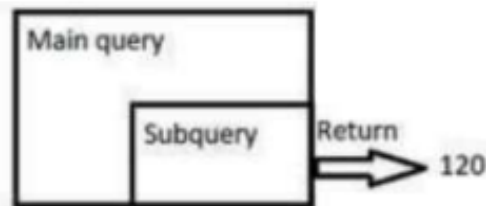
BAB III

KESIMPULAN

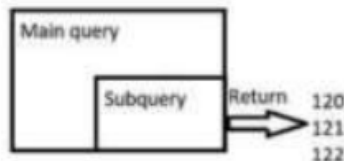
>> Subquery atau subquery nesting adalah suatu teknik dalam bahasa SQL yang memungkinkan pembuatan query sebagai subquery di dalam query lain. Fungsi utama subquery adalah untuk menghasilkan satu set data yang diperlukan untuk digunakan dalam query utama.

>> Subquery diklasifikasikan ke dalam tiga jenis : scalar, multiple-row, dan multiple-column.

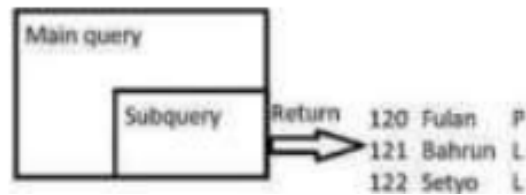
A. Scalar Subquery Subquery baris tunggal (scalar) hanya mengembalikan hasil sat baris data. Bentuk subquery ini dipelihatkan seperti Gambar:



B. Multiple row subquery Subquery baris multiple row mengembalikan lebih dari satu baris data. Bentuk subquery ini diperlihatkan seperti Gambar:



C. Multiple-Column Subquery Subquery multiple column mengembalikan lebih dari satu baris dan satu kolom data. Bentuk subquery ini dapat dilihat seperti Gambar:



>> Syntaks dasar subquery adalah sebagai berikut :

```
SELECT column_name [,column_name] FROM table1 [,table2] WHERE column_name  
OPERATOR (SELECT column_name [,column_name] FROM table1 [,table2] [WHERE]);
```

>> Jika digunakan secara tidak tepat atau pada skala yang besar, penggunaan subquery dapat menyebabkan performa database menurun, karena subquery harus dieksekusi secara terpisah dari query utama. Oleh karena itu, penting untuk memperhatikan efisiensi query saat menggunakan subquery dan mengoptimalkan query jika diperlukan.

DAFTAR PUSTAKA

- Modul Praktikum Basis Data Pertemuan 5
- <https://www.w3resource.com/sql/subqueries/understanding-sql-subqueries.php>
- <https://www.programiz.com/sql/subquery>
- <https://learn.microsoft.com/en-us/sql/relational-databases/performance/subqueries?view=sql-server-ver16>