

Ajeng Astri Utami, S.Pd
SMKN 4 Kota Bogor

MODUL BASIS DATA

SQL Bertingkat

SQL BERTINGKAT

Daftar Isi

A. Pendahuluan	2
B. Pembelajaran	4
D. Kunci Jawaban	2
E. Daftar Pustaka	4
F. Penutup	5
Company Information	5

SQL BERTINGKAT

A. Pendahuluan

Subquery (disebut juga *subselect* atau *nested select/query* atau *inner-select*) adalah query `SELECT` yang ada di dalam perintah SQL lain misalnya `SELECT`, `INSERT`, `UPDATE`, atau `DELETE`.

Keberadaan subquery secara nyata mampu menyederhanakan persoalan-persoalan rumit berkaitan query data. Sebagai contoh, misal terdapat pernyataan sebagai berikut:

“Dapatkan data siswa yang alamatnya sama dengan siswa dengan nis 123”

Secara normal, diperlukan dua tahapan untuk menyelesaikan kasus di atas. Pertama adalah mendapatkan alamat dari siswa yang memiliki nis 123. Langkah selanjutnya, baru kita bisa mengetahui data siswa yang alamatnya sama dengan siswa dengan nis 123.

Adapun dengan memanfaatkan subquery, maka penyelesaian kasus di atas hanya memerlukan sebuah query (akan dijelaskan nanti). Pada hakekatnya, subquery sangat berguna ketika sebuah query didasarkan pada nilai-nilai yang tak diketahui.

1. STANDART KOMPTENESI

3.11 Merancang SQL bertingkat dalam memanipulasi basis data

4.11 Membuat perintah SQL bertingkat dalam memanipulasi basis data

2. DESKRIPSI

Modul ini merupakan modul pembelajaran mata pelajaran Basis Data untuk siswa SMK kelas XII semester 5. Modul pembelajaran ini dibuat dalam rangka untuk mempermudah proses pembelajaran. Modul ini berisi materi SQL bertingkat.

3. WAKTU

Alokasi pembelajaran untuk modul SQL bertingkat ini dapat digunakan dalam waktu pembelajaran selama satu kali tatap muka.

4. PETUNJUK PENGGUNAAN MODUL

Sebelum Pembelajaran

1. Sebelum masuk pada materi, disajikan pendahuluan sebagai pengantar menuju materi utama.
2. Disajikan kompetensi dasar dan alokasi waktu sebagai pedoman bagi pengguna modul untuk mencapai tujuan pembelajaran.

SQL BERTINGKAT

Selama Pembelajaran

1. Mempelajari dan memahami materi pada modul.
2. Mempelajari dan mencatat materi yang dianggap penting.
3. Mengerjakan tugas yang terdapat pada bagian evaluasi.
4. Mengerjakan tes untuk mengukur kemampuan dalam memahami modul.

Setelah Pembelajaran

1. Mengevaluasi jawaban dengan kunci jawaban.
2. Mengetahui hasil evaluasi (sudah memenuhi kriteria ketuntasan atau belum).
3. Memutuskan untuk meneruskan belajar pada materi selanjutnya atau tetap pada materi yang sama.

SQL BERTINGKAT

B. Pembelajaran

Materi SQL bertingkat dapat kita sebut dengan istilah SubQuery.

Sintaks formal subquery diperlihatkan sebagai berikut:

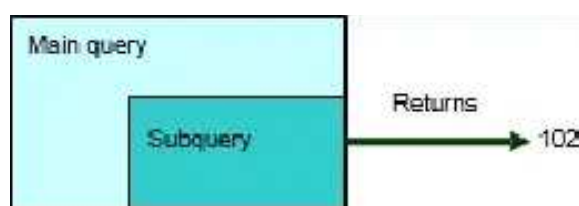
```
SELECT  $A1, A2, \dots, An$   
FROM  $r1, r2, r3, \dots, rm$   
WHERE  $P$   
  
(SELECT  $A1, A2, \dots, An$   
FROM  $r1, r2, r3, \dots, rm$   
WHERE  $P$ )
```

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

a. Scalar Subquery

Subquery baris tunggal (*scalar*) hanya mengembalikan hasil satu baris data.

Bentuk subquery ini diperlihatkan seperti Gambar 1.

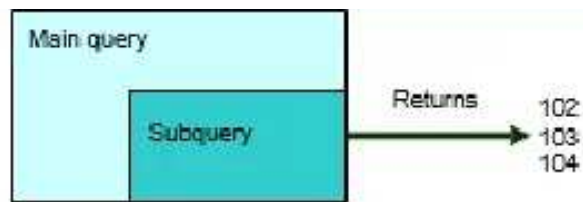


Gambar 1 Scalar subquery

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

b. Multiple-Row Subquery

Subquery baris ganda (*multiple-row*) mengembalikan lebih dari satu baris data. Bentuk subquery ini diperlihatkan seperti Gambar 2.



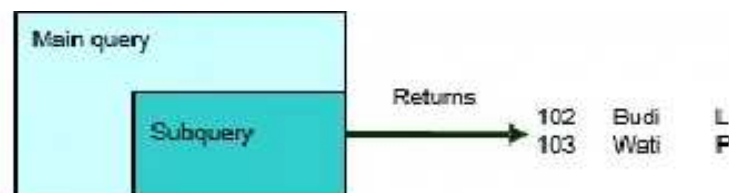
Gambar 2 Multiple-row subquery

SQL BERTINGKAT

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

c. Multiple-Column Subquery

Subquery kolom ganda (*multiple-column*) mengembalikan lebih dari satu baris dan satu kolom data. Bentuk subquery ini diperlihatkan seperti Gambar 3.



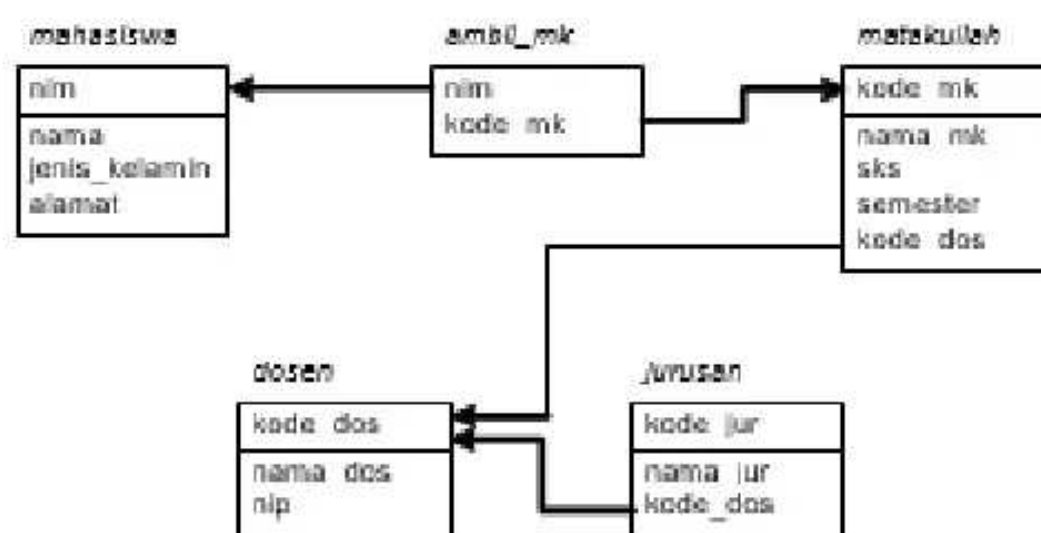
Gambar 3 Multiple-column subquery



Subquery dapat diimplementasikan di dalam klausa WHERE, HAVING, atau FROM.

SQL BERTINGKAT

Himpunan entitas di atas dapat direpresentasikan ke dalam diagram skema (*schema diagram*) seperti Gambar 4.



Gambar 4 Diagram skema

2. Scalar Subquery

Contoh subquery baris tunggal adalah mendapatkan data mahasiswa yang alamatnya sama dengan mahasiswa dengan nim 104.

```
mysql> SELECT *
      => FROM mahasiswa
      => WHERE alamat =
```

Jl. Jombang

```

-> WHERE alamat
-> (SELECT alamat
-> FROM mahasiswa
-> WHERE nim = 104);

```

nim	nama	jenis_kelamin	alamat
102	Budi	L	Jl. Jombang
104	Ika	P	Jl. Jombang

2 rows in set (0.00 sec)

Bisa digambarkan, langkah pertama dari operasi di atas adalah mendapatkan alamat mahasiswa dengan nim 104, kemudian hasilnya—yakni “Jl.Jombang”—akan digunakan oleh main query.

3. Multiple-Row Subquery

Pada subquery ini, kita menggunakan operator komparasi IN, ANY/SOME, atau ALL.

Operator IN

Operator IN memiliki arti: sama dengan member di dalam list. Sebagai contoh, kita bisa menggunakan operator ini untuk mendapatkan data mahasiswa yang mengambil matakuliah.

SQL BERTINGKAT

```

mysql> SELECT m.nim, m.nama
-> FROM mahasiswa m
-> WHERE m.nim IN
-> (SELECT nim
-> FROM ambil_mk);

```

nim	nama
101	Arif
103	Wati
104	Ika

3 rows in set (0.00 sec)

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 kecil dari sembarang sks matakuliah di semester 5.

```

mysql> SELECT *
-> FROM matakuliah

```

3, 3, 2, 3


```

-> WHERE sks < ANY
->
->
->

```

```

(SELECT sks
FROM matakuliah
WHERE semester = 5);

```

kode_mk	nama_mk	sks	semester	kode_dos
PTI447	Praktikum Basis Data	1	3	11
PTI777	Sistem Informasi	2	3	99
TIK123	Jaringan Komputer	2	5	33
TIK342	Praktikum Basis Data	1	3	11

4 rows in set (0.00 sec)

Operator = ANY ekuivalen dengan IN.

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

Operator > ANY ekuivalen dengan MIN (lebih dari min).

Operator ALL

Operator ALL memiliki arti membandingkan suatu nilai dengan semua nilai yang dikembalikan oleh subquery.

Misalkan kita ingin mendapatkan data matakuliah yang memiliki sks lebih kecil dari semua sks matakuliah di semester 5.

```

mysql> SELECT *
-> FROM matakuliah
-> WHERE sks < ALL
->
->
->

```

```

(SELECT sks
FROM matakuliah
WHERE semester = 5);

```

kode_mk	nama_mk	sks	semester	kode_dos
PTI447	Praktikum Basis Data	1	3	11
TIK342	Praktikum Basis Data	1	3	11

2 rows in set (0.16 sec)

3, 3, 2, 3

7 | P

SQL BERTINGKAT

Operator < ALL ekuivalen dengan MIN (kurang dari min).

Operator > ALL ekuivalen dengan MAX (lebih dari maks).

4. Multiple-Column Subquery

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 dengan main query.

Sebagai contoh, misalkan kita ingin menampilkan data matakuliah yang semester dan sksnya sesuai dengan semester dan sks matakuliah dengan

kode "PTI333".

```
mysql> SELECT *  
-> FROM matakuliah  
-> WHERE (semester, sks) IN  
-> (SELECT semester, sks  
-> FROM matakuliah  
-> WHERE kode_mk= 'PTI333'); 5 3
```

kode_mk	nama_mk	sks	semester	kode_dos
PTI123	Grafika Multimedia	3	5	12
PTI333	Basis Data Terdistribusi	3	5	10
TIK333	Sistem Operasi	3	5	10

3 rows in set (0.00 sec)

5.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, EXISTS akan mengembalikan nilai true; begitu pula sebaliknya, jika tidak mengembalikan hasil.

Sebagai contoh, pernyataan berikut akan mendapatkan data mahasiswa yang mengambil matakuliah.

```
mysql> SELECT * FROM mahasiswa m  
-> WHERE EXISTS (SELECT * FROM ambil_mk a  
-> WHERE m.nim = a.nim);
```

nim	nama	jenis_kelamin	alamat
101	Arif	L	Jl. Kenangan
103	Wati	P	Jl. Surabaya
104	Ika	P	Jl. Jombang

3 rows in set (0.08 sec)

SQL BERTINGKAT

Pernyataan berikut akan mendapatkan data mahasiswa yang tidak mengambil matakuliah.

```
mysql> SELECT * FROM mahasiswa m  
-> WHERE NOT EXISTS (SELECT * FROM ambil_mk a  
-> WHERE m.nim = a.nim);
```

nim	nama	jenis_kelamin	alamat
102	Budi	L	Jl. Jombang
105	Tono	L	Jl. Jakarta
106	Iwan	L	Jl. Bandung

107	Sari	P	21. Malang
-----	------	---	------------

4 rows in set (0.00 sec)

6. Subquery dan Fungsi Agregat

Operasi-operasi pada subquery juga dapat melibatkan fungsi-fungsi agregat.

Sebagai contoh, misalkan ingin mendapatkan data matakuliah yang memiliki sks sama dengan sks terbesar.

```
mysql> SELECT *
-> FROM matakuliah
-> WHERE sks =
-> (SELECT MAX(sks)
-> FROM matakuliah);
```

kode_mk	nama_mk	sks	semester	kode_dos
PTI123	Grafika Multimedia	3	5	12
PTI333	Basis Data Terdistribusi	3	5	10
TIK333	Sistem Operasi	3	5	10

3 rows in set (0.06 sec)

7. Subquery dan Join

Dalam beberapa kasus sederhana, fungsionalitas subquery dan join dapat dipertukarkan. Dengan kata lain, keduanya dapat digunakan untuk menyelesaikan persoalan yang sama.

Sebagai contoh, misalkan kita ingin mendapatkan nim dan nama mahasiswa yang tidak mengambil matakuliah.

Pendekatan subquery:

```
mysql> SELECT m.nim, m.nama
-> FROM mahasiswa m
-> WHERE m.nim NOT IN
-> (SELECT nim
-> FROM ambil_mk);
```

nim	nama
102	Budi
105	Tono
106	Iwan
107	Sari

4 rows in set (0.00 sec)

```
mysql> SELECT m.nim, m.nama  
-> FROM mahasiswa m  
-> LEFT OUTER JOIN ambil_mk a  
-> ON m.nim = a.nim  
-> WHERE a.nim IS NULL;
```

nim	nama
102	Budi
105	Tono
106	Iwan
107	Sari

4 rows in set (0.00 sec)



Jika diperhatikan, pernyataan pada subquery mengalir secara alami—lebih mudah dicerna—dibanding join. Bagaimanapun, salah satu kendala dalam join adalah menentukan mana yang akan menjadi tabel kiri dan kanan ☹.

C. Evaluasi

Dalam mengerjakan tugas praktikum ini, sebaiknya pernyataan SQL disimpan di file untuk kemudian dieksekusi.

Selesaikan tugas-tugas di bawah ini dengan menggunakan pendekatan subquery.

1. Dapatkan data mahasiswa yang jenis kelaminnya sama dengan "Arif" tidak termasuk "Arif". (Point 10)
2. Dapatkan matakuliah yang tidak diambil oleh mahasiswa terdaftar (mahasiswa di tabel mahasiswa). (Point 20)
3. Dapatkan data mahasiswa yang mengambil matakuliah dengan sks lebih kecil dari sembarang sks. (Point 25)
4. Dapatkan data dosen yang mengajar matakuliah di semester yang sama dengan dosen yang sekaligus menjadi ketua jurusan Teknik Elektro tidak termasuk ketua jurusan Teknik Elektro. (Point 30)
5. Dapatkan nim, nama, dan alamat mahasiswa yang tempat tinggalnya sama dengan dosen yang mengajar matakuliah dengan sks di atas rata-rata. (Point 35)

D. Kunci Jawaban

No Soal	Jawaban	Skor
1.	<pre>select * from mahasiswa where nama <> (select nama from mahasiswa where nama='arif') and jenis_kelamin='L';</pre>	20
2.	<pre>select * from matakuliah where kode_mk not in (select kode_mk from ambil_mk);</pre>	20
3.	<pre>select m.nim, m.nama, m.jenis_kelamin, m.alamat from mahasiswa m, ambil_mk a, matakuliah mk where m.nim = a.nim and a.kode_mk = mk.kode_mk and mk.sks < any (select sks from matakuliah); atau select m.nim, m.nama, m.jenis_kelamin, m.alamat from mahasiswa m, ambil_mk a, matakuliah mk where m.nim = a.nim and a.kode_mk = mk.kode_mk and mk.sks < all (select sks from matakuliah where semester=5);</pre>	20
4.	<pre>select d.kode_dos, d.nama_dos, d.alamat_dos from dosen d, matakuliah mk where d.kode_dos = mk.kode_dos and mk.semester in (select mk.semester from jurusan j, matakuliah mk where mk.kode_dos = j.kode_dos and j.nama_jur='Teknik Elektro')</pre>	20

E. Daftar Pustaka

Buku

Madcoms. 2018. *PHP & MySQL*. Madiun: Madcoms.

Aunur Rofiq. 2008. *Rekayasa Perangkat Lunak Jilid 3*. Jakarta: Departemen Pendidikan Nasional.

Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Malang (UM). 2015. *Modul Praktikum Basis Data..* Malang: Universitas Negeri Malang.

Internet

Achmad, Fahmi. 2013. ERD (*Entity Relationship Diagram*). Online. (<http://fahmiachmad96.blogspot.com/2013/09/erd-entity-relationship-diagram.html>), diakses 27 Agustus 2018.

Dany Septian. Ferdinan. 2013. *Basis data materi #3*. Online. (<http://fseptian.mhs.uksw.edu/2013/02/basis-data-materi-3.html>), diakses 26 Agustus 2018.

Darmawan. 2018. *Normalisasi database beserta pengertian dan contohnya*. Online. darmawan.blogspot.com, diakses 28 Agustus 2018.

Irine. Herlinna. 2014. *Entity Relationship Diagram (ERD) dan Contoh Kasus*. Online. (<http://herlinnairine.wordpress.com/2014/02/06/entity-relationship-diagram-erd-dan-contoh-kasus/>), diakses 26 Agustus 2018.

Rudiawan. 2018. *Normalisasi database beserta pengertian dan contohnya*. Online. <https://rudiawan16.wordpress.com/normalisasi-database-beserta-pengertian-dan-contohnya>. diakses 28 Agustus 2018.

Winahyu. Rheza. 2013. *Komponen Sistem Basis Data dan Abstraksi Data*. Online. (<http://the-simple-blogs.blogspot.com/2013/02/komponen-sistem-basis-data-dan.html>), diakses 27 Agustus 2018.