

BAB 7. JOIN

7.1 TUJUAN PRAKTIKUM

Tujuan Umum

Mahasiswa mampu memahami dan menerapkan Kueri SQL – Join

Tujuan Khusus

Mahasiswa mampu:

- Menerapkan Kueri SQL - Inner Join
- Menerapkan Kueri SQL - Left Join
- Menerapkan Kueri SQL - Right Join
- Menerapkan Kueri SQL - View
- Menerapkan Kueri SQL - SubQuery

7.2 TEORI SINGKAT

a. Join

Join digunakan untuk menghubungkan dua tabel atau merelasikan beberapa tabel yang berbeda untuk mengambil beberapa kolom yang dibutuhkan berdasarkan hubungan antar kolom tertentu pada sebuah tabel atau bisa disebut dengan *foreign key*.

Sebelum mencoba menggunakan JOIN, terlebih dahulu kita buat tabel dengan nama `mata_kuliah` dengan menjalankan kueri SQL seperti di bawah ini:

```
create table mata_kuliah(  
    matkul_kode varchar(10) not null primary key,  
    matkul_nama varchar(30) not null unique  
);  
  
insert into mata_kuliah values  
    ("MK001", "Algoritma"),  
    ("MK002", "Basis Data"),  
    ("MK003", "Jaringan Komputer"),  
    ("MK004", "Pemrograman Web"),  
    ("MK005", "Pemrograman Mobile");
```

Dengan menggunakan perintah SQL pengambilan data, isi tabel `mata_kuliah` dapat dilihat pada Gambar 7.1 berikut:

```
mysql> select * from mata_kuliah;
```

matkul_kode	matkul_nama
MK001	Algoritma
MK002	Basis Data
MK003	Jaringan Komputer
MK005	Pemrograman Mobile
MK004	Pemrograman Web

Gambar 7.1 Hasil tampilan data pada tabel mata kuliah

Ada beberapa teknik join pada SQL, diantaranya :

b. Inner Join

Inner join merupakan jenis join yang paling umum yang dapat digunakan pada semua database. Jenis ini dapat digunakan bila ingin merelasikan dua set data yang ada pada tabel, letak relasinya setelah pada perintah ON pada join.

Format SQL yang umum digunakan adalah:

```
SELECT nama_kolom
FROM nama_tabel1
INNER JOIN nama_tabel2 ON
nama_tabel1.nama_kolom=nama_tabel2.nama_kolom
```

Bisa juga tidak memakai kata inner pada join di atas, sehingga kuerinya menjadi:

```
SELECT nama_kolom
FROM nama_tabel1
JOIN nama_tabel2 ON
nama_tabel1.nama_kolom=nama_tabel2.nama_kolom
```

Contoh :

```
select * from tabel_nilai n join mata_kuliah m on n.matkul_kode=m.matkul_kode;
```

```
mysql> select * from tabel_nilai n join mata_kuliah m on n.matkul_kode=m.matkul_kode;
```

nilai_id	nim	matkul_kode	nilai	matkul_kode	matkul_nama
1	1001001	MK001	70	MK001	Algoritma
2	1001001	MK002	80	MK002	Basis Data
3	1001001	MK003	65	MK003	Jaringan Komputer
4	1001001	MK004	80	MK004	Pemrograman Web
5	1001002	MK001	60	MK001	Algoritma
6	1001002	MK002	80	MK002	Basis Data
7	1001002	MK003	80	MK003	Jaringan Komputer
8	1001002	MK004	80	MK004	Pemrograman Web
9	1001003	MK001	90	MK001	Algoritma
10	1001003	MK002	60	MK002	Basis Data
11	1001003	MK003	80	MK003	Jaringan Komputer
12	1001003	MK004	80	MK004	Pemrograman Web
13	1001004	MK001	85	MK001	Algoritma
14	1001004	MK002	80	MK002	Basis Data
15	1001004	MK003	75	MK003	Jaringan Komputer
16	1001004	MK004	80	MK004	Pemrograman Web
17	1001005	MK001	70	MK001	Algoritma
18	1001005	MK002	80	MK002	Basis Data
19	1001005	MK003	75	MK003	Jaringan Komputer
20	1001005	MK004	80	MK004	Pemrograman Web

Gambar 7.2 Hasil tampilan semua kolom dengan Inner join

c. Left Join

Left join akan mengeluarkan semua kolom pada tabel di kiri(pertama) meskipun pada tabel kanan(kedua) tidak memiliki kesamaan

SQL yang umum digunakan adalah:

```
SELECT nama_kolom
FROM nama_tabel1
LEFT JOIN nama_tabel2
ON nama_tabel1.nama_kolom=nama_tabel2.nama_kolom
```

Contoh:

```
select * from mata_kuliah m left join tabel_nilai n on n.matkul_kode=
m.matkul_kode;
```

```
mysql> select * from mata_kuliah m left join tabel_nilai n on n.matkul_kode=m.matkul_kode;
```

matkul_kode	matkul_nama	nilai_id	nim	matkul_kode	nilai
MK001	Algoritma	1	1001001	MK001	70
MK002	Basis Data	2	1001001	MK002	80
MK003	Jaringan Komputer	3	1001001	MK003	65
MK004	Pemrograman Web	4	1001001	MK004	80
MK001	Algoritma	5	1001002	MK001	60
MK002	Basis Data	6	1001002	MK002	80
MK003	Jaringan Komputer	7	1001002	MK003	80
MK004	Pemrograman Web	8	1001002	MK004	80
MK001	Algoritma	9	1001003	MK001	90
MK002	Basis Data	10	1001003	MK002	60
MK003	Jaringan Komputer	11	1001003	MK003	80
MK004	Pemrograman Web	12	1001003	MK004	80
MK001	Algoritma	13	1001004	MK001	85
MK002	Basis Data	14	1001004	MK002	80
MK003	Jaringan Komputer	15	1001004	MK003	75
MK004	Pemrograman Web	16	1001004	MK004	80
MK001	Algoritma	17	1001005	MK001	70
MK002	Basis Data	18	1001005	MK002	80
MK003	Jaringan Komputer	19	1001005	MK003	75
MK004	Pemrograman Web	20	1001005	MK004	80
MK005	Pemrograman Mobile	NULL	NULL	NULL	NULL

Gambar 7.3 Hasil tampilan semua kolom dengan Left join

d. Right Join

Right join adalah kebalikan dari left join, dimana right join akan mengeluarkan semua kolom pada tabel kanan (kedua) meskipun pada tabel kiri (pertama) tidak memiliki pasangan.

Format SQL yang umum digunakan adalah:

```
SELECT nama_kolom
FROM nama_tabel1
RIGHT JOIN nama_tabel2
ON nama_tabel1.nama_kolom=nama_tabel2.nama_kolom
```

Contoh :

```
select * from tabel_nilai n right join mata_kuliah m on n.matkul_kode
=m.matkul_kode;
```

```
mysql> select * from tabel_nilai n right join mata_kuliah m on n.matkul_kode=m.matkul_kode;
```

nilai_id	nim	matkul_kode	nilai	matkul_kode	matkul_nama
1	1001001	MK001	70	MK001	Algoritma
2	1001001	MK002	80	MK002	Basis Data
3	1001001	MK003	65	MK003	Jaringan Komputer
4	1001001	MK004	80	MK004	Pemrograman Web
5	1001002	MK001	60	MK001	Algoritma
6	1001002	MK002	80	MK002	Basis Data
7	1001002	MK003	80	MK003	Jaringan Komputer
8	1001002	MK004	80	MK004	Pemrograman Web
9	1001003	MK001	90	MK001	Algoritma
10	1001003	MK002	60	MK002	Basis Data
11	1001003	MK003	80	MK003	Jaringan Komputer
12	1001003	MK004	80	MK004	Pemrograman Web
13	1001004	MK001	85	MK001	Algoritma
14	1001004	MK002	80	MK002	Basis Data
15	1001004	MK003	75	MK003	Jaringan Komputer
16	1001004	MK004	80	MK004	Pemrograman Web
17	1001005	MK001	70	MK001	Algoritma
18	1001005	MK002	80	MK002	Basis Data
19	1001005	MK003	75	MK003	Jaringan Komputer
20	1001005	MK004	80	MK004	Pemrograman Web
NULL	NULL	NULL	NULL	MK005	Pemrograman Mobile

Gambar 7.4 Hasil tampilan semua kolom dengan Right join

Selain itu, kita dapat melakukan JOIN lebih dari 2 tabel dengan kode sama seperti kode di atas. Berikut adalah contohnya:

```
select o.nim, o.nama, m.matkul_nama, n.nilai from
mahasiswa o join tabel_nilai n join mata_kuliah m
on o.nim=n.nim and n.matkul_kode=m.matkul_kode limit 5;
```

```
mysql> select o.nim, o.nama, m.matkul_nama, n.nilai from
-> mahasiswa o join tabel_nilai n join mata_kuliah m
-> on o.nim=n.nim and n.matkul_kode=m.matkul_kode limit 5;
```

nim	nama	matkul_nama	nilai
1001004	ZASKIA	Algoritma	85
1001004	ZASKIA	Basis Data	80
1001004	ZASKIA	Jaringan Komputer	75
1001004	ZASKIA	Pemrograman Web	80
1001005	ZIDNI	Algoritma	70

Gambar 7.5 Hasil tampilan beberapa kolom dengan join lebih dari 2 tabel

e. View

View merupakan sebuah tabel semu / tabel lojik, dimana datanya berasal dari satu atau lebih tabel lain yang disebut sebagai tabel sumber. View biasa dibuat untuk memudahkan user menampilkan data.

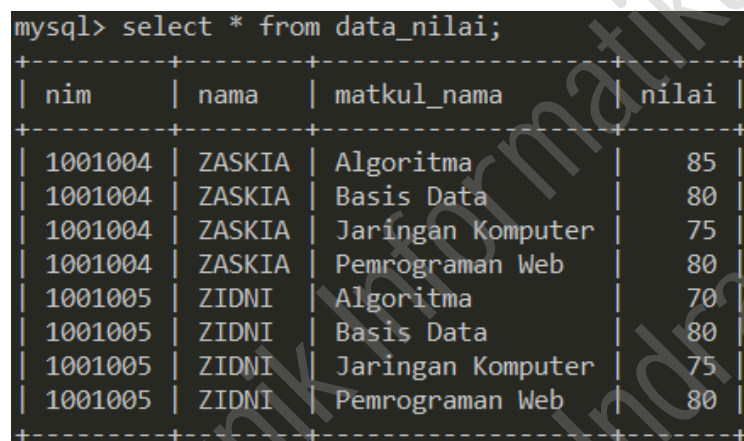
Format SQL yang umum digunakan adalah:

```
create view nama_view as <query expression>
```

Contoh :

```
create view data_nilai as
select o.nim, o.nama, m.matkul_nama, n.nilai from mahasiswa o join
tabel_nilai n join mata_kuliah m on o.nim=n.nim and n.matkul_kode
=m.matkul_kode;
```

```
select * from data_nilai;
```



```
mysql> select * from data_nilai;
```

nim	nama	matkul_nama	nilai
1001004	ZASKIA	Algoritma	85
1001004	ZASKIA	Basis Data	80
1001004	ZASKIA	Jaringan Komputer	75
1001004	ZASKIA	Pemrograman Web	80
1001005	ZIDNI	Algoritma	70
1001005	ZIDNI	Basis Data	80
1001005	ZIDNI	Jaringan Komputer	75
1001005	ZIDNI	Pemrograman Web	80

Gambar 7.6 Hasil tampilan dengan View data_nilai

f. SubQuery

Subquery atau Inner query atau Nested query adalah kueri yang berada di dalam kueri SQL lain dan baik terletak pada bagian FROM maupun klausa WHERE. Sebuah subquery digunakan untuk mengembalikan data yang akan digunakan dalam query utama sebagai syarat untuk lebih membatasi data yang akan diambil.

Format SQL yang umum digunakan adalah:

```
SELECT [nama-nama kolom]
FROM (SELECT [nama-nama kolom]
FROM [nama-nama table]
[WHERE])
```

```
WHERE [nama kolom] [operator]
(SELECT [nama-nama kolom]
FROM [nama-nama table]
[WHERE])
```

Contoh :

Berikut adalah contoh subquery dalam kondisi (where) dengan *single value*

```
select nim, nama from mahasiswa where nim=(select nim from
tabel_nilai order by nilai desc limit 1);
```

```
mysql> select nim, nama from mahasiswa where
-> nim=(select nim from tabel_nilai order by nilai desc limit 1);
+-----+-----+
| nim    | nama  |
+-----+-----+
| 1001003 | ZAFRAN |
+-----+-----+
```

Gambar 7.7 Hasil tampilan subquery dengan satu kondisi

Berikut adalah contoh subquery dalam kondisi (where) dengan *multi value*

```
select nim, nama from mahasiswa where nim in
(select nim from tabel_nilai where matkul_kode="MK001" and nilai>=80);
```

```
mysql> select nim, nama from mahasiswa where nim in
-> (select nim from tabel_nilai where matkul_kode="MK001" and nilai>=80);
+-----+-----+
| nim    | nama  |
+-----+-----+
| 1001003 | ZAFRAN |
| 1001004 | ZASKIA |
+-----+-----+
```

Gambar 7.8 Hasil tampilan subquery dengan lebih dari satu kondisi

Berikut adalah contoh subquery dalam kondisi (having) dengan *single value*

```
select nim, avg(nilai) as rerata from tabel_nilai group by nim
having rerata>(select avg(nilai) from tabel_nilai where nim=
"1001003");
```

```
mysql> select nim, avg(nilai) as rerata from tabel_nilai
-> group by nim having rerata>
-> (select avg(nilai) from tabel_nilai where nim="1001003");
```

nim	rerata
1001004	80.0000

Gambar 7.9 Hasil tampilan subquery dengan having satu kondisi

Berikut adalah contoh subquery dalam kondisi (having) dengan *multi value*

```
select nim, min(nilai) as nilai_terkecil from tabel_nilai
group by nim having nilai_terkecil in
(select nilai from tabel_nilai where matkul_kode="MK001"
order by nilai asc);
```

```
mysql> select nim, min(nilai) as nilai_terkecil from tabel_nilai
-> group by nim having nilai_terkecil in
-> (select nilai from tabel_nilai where matkul_kode="MK001"
-> order by nilai asc);
```

nim	nilai_terkecil
1001002	60
1001003	60
1001005	70

Gambar 7.10 Hasil tampilan subquery dengan having lebih dari satu kondisi

Berikut adalah contoh subquery dalam from:

```
select x.nama, x.matkul_nama from
(select o.nim, o.nama, m.matkul_nama, n.nilai
from mahasiswa o join tabel_nilai n join mata_kuliah m
on o.nim=n.nim and n.matkul_kode=m.matkul_kode) as x;
```



```
mysql> select x.nama, x.matkul_nama from
-> (select o.nim, o.nama, m.matkul_nama, n.nilai
-> from mahasiswa o join tabel_nilai n join mata_kuliah m
-> on o.nim=n.nim and n.matkul_kode=m.matkul_kode) as x;
```

nama	matkul_nama
DZAMAAR	Algoritma
DZAMAAR	Basis Data
DZAMAAR	Jaringan Komputer
DZAMAAR	Pemrograman Web
ZACHIRAH	Algoritma
ZACHIRAH	Basis Data
ZACHIRAH	Jaringan Komputer
ZACHIRAH	Pemrograman Web
ZAFRAN	Algoritma
ZAFRAN	Basis Data
ZAFRAN	Jaringan Komputer
ZAFRAN	Pemrograman Web
ZASKIA	Algoritma
ZASKIA	Basis Data
ZASKIA	Jaringan Komputer
ZASKIA	Pemrograman Web
ZIDNI	Algoritma
ZIDNI	Basis Data
ZIDNI	Jaringan Komputer
ZIDNI	Pemrograman Web

Gambar 7.11 Hasil tampilan subquery dalam from

7.3 LATIHAN

1. Berdasarkan tabel barang yang sudah dibuat pada latihan sebelumnya, tabel transaksi dan tabel detail_transaksi, sesuaikanlah hingga datanya seperti berikut ini:

```
mysql> select * from barang;
```

barang_id	barang_nama	barang_harga	barang_kode	barang_jenis
1	Beras	8000	B01	primer
2	Gula	8000	B02	primer
3	Garam	10000	B11	primer
4	Mie	5000	B04	primer
5	Sabun	10000	B05	sekunder
6	Deterjen	6000	B06	sekunder
7	Minyak Tanah	10000	B07	sekunder
8	Minyak Goreng	10000	B08	primer
9	Roti A	10000	B09	sekunder
10	Sirup	10000	B10	sekunder
11	perhiasan	2000000	B90	tersier
12	sepeda gunung	2600000	B91	tersier

```
12 rows in set (0.00 sec)
```

```
mysql> select * from transaksi;
```

transaksi_id	waktu	total_bayar
4	2020-10-27 00:17:45	NULL
5	2020-10-27 00:17:46	NULL

```
2 rows in set (0.00 sec)
```

```
mysql> select * from detail_transaksi;
```

detail_transaksi_id	transaksi_id	barang_id	harga	jumlah	total
9	4	1	8000	3	24000
10	4	2	8000	3	24000
11	5	2	8000	2	16000
12	5	4	5000	3	15000
13	5	3	10000	5	50000
14	5	10	10000	5	50000

2. Buatlah kueri SQL hingga tiga tabel diatas dapat ditampilkan seperti berikut:

nama barang	harga	jumlah	total
Beras	8000	3	24000
Gula	8000	3	24000
Gula	8000	2	16000
Mie	5000	3	15000
Garam	10000	5	50000
Sirup	10000	5	50000

3. Buatlah kueri SQL dari latihan no.2 menjadi sebuah view!

7.4 TUGAS

1. Berdasarkan tabel yang ada pada bagian latihan dihalaman sebelumnya, buatlah kueri SQL dengan menerapkan subquery untuk mengubah data pada tabel transaksi hingga seperti berikut:

```
mysql> select * from transaksi;
```

transaksi_id	waktu	total_bayar
4	2020-10-27 00:17:45	48000
5	2020-10-27 00:17:46	131000

2. Buatlah sebuah view untuk menampilkan data seperti berikut:

transaksi_id	nama barang	harga	jumlah	total	total_bayar
4	Beras	8000	3	24000	48000
4	Gula	8000	3	24000	48000