

PRAKTIKUM SISTEM BASIS DATA

Modul 6: JOIN Antar Tabel

Tujuan dan Capaian Pembelajaran Praktikum:

1. Mahasiswa dapat menjelaskan konsep serta jenis-jenis JOIN.
2. Mahasiswa dapat menampilkan data gabungan dari berbagai tabel menggunakan JOIN.

INSTRUKSI UMUM: Bacalah dahulu modul ini sebelum praktikum, sehingga ketika praktikum bisa langsung mencoba latihan dan menjelaskan variasi kasus yang mungkin muncul.

Alat dan Bahan:

1. Modul Praktikum
2. Komputer
3. Software MySQL
4. Viewer LCD

Materi dan Pembagian Waktu:

Materi dan pembagian waktu pelaksanaan praktikum meliputi:

No	Materi	Waktu
1	Pretest	10 menit
2	DML SELECT dengan JOIN	50 menit
3	Latihan Mandiri	45 menit
4	Posttest	45 menit
Total		150 menit

Materi Praktikum

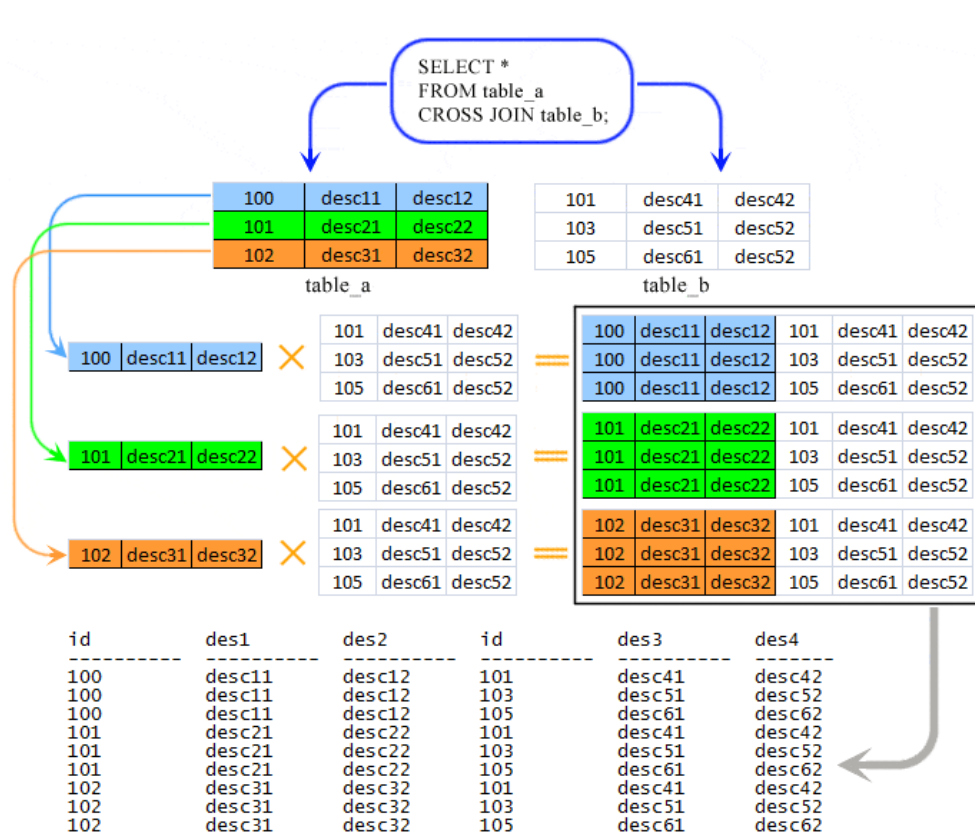
Seperti sudah dipelajari pada materi sebelumnya, SELECT memiliki kemampuan untuk mengambil data dari satu tabel atau lebih. Hasil SELECT dari tabel terpisah tersebut bisa berdasarkan relasi tabel induk dengan anak ataupun tidak ada relasi sama sekali. Join tidak bisa berdiri sendiri. Penggunaannya diletakkan sebagai klausa di dalam statemen SELECT, UPDATE maupun DELETE walau umumnya cukup jarang menggunakan join pada UPDATE dan DELETE.

Jika kita sudah memahami konsep normalisasi basis data yang berfungsi memecah tabel tidak normal menjadi tabel-tabel induk dan anak yang memenuhi kaidah Normal Form, maka Join di sini bertugas membalikkan fungsi normalisasi, atau dengan kata lain melakukan denormalisasi untuk mendapatkan satu tabel yang tidak normal namun memenuhi kebutuhan informasi yang hendak didapatkan.

MySQL versi MariaDB memiliki empat jenis join, yaitu CROSS JOIN, INNER JOIN, NATURAL JOIN dan OUTER JOIN.

1. CROSS JOIN

Tipe join ini akan memasangkan semua row dari tabel pertama dan tabel kedua tanpa mempertimbangkan kondisi relasi nilai kunci yang terkait. Join ini juga disebut **cartesian join**. Tanpa adanya pembatasan kondisi relasi, maka join ini relatif tidak memiliki manfaat. Contoh cross join terlihat pada Gambar 1.

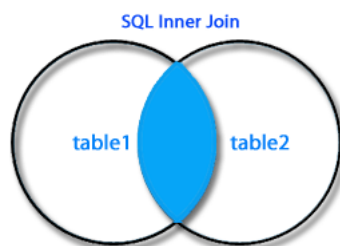


Gambar 1. Cross Join

(<https://www.w3resource.com/mysql/advance-query-in-mysql/mysql-cross-join.php>)

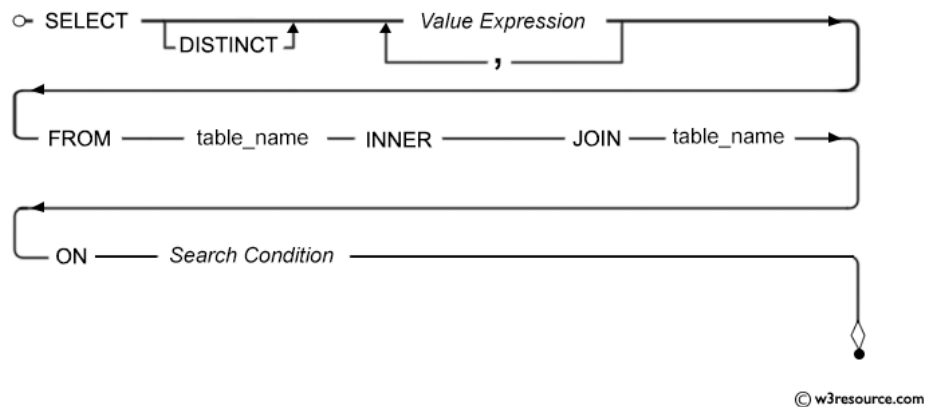
2. INNER JOIN

Tipe join ini akan mengambil semua row dari tabel pertama dan tabel kedua jika dan hanya jika terdapat kondisi nilai kunci yang terkait. Dengan kata lain, inner join menghasilkan data irisan antar kedua tabel yang dijoinkan (Gambar 2).



Gambar 2. Irisan dalam Inner Join

Jika tidak ada keterkaitan/irisan maka row tersebut tidak akan muncul. Sintaks Inner Join dalam statemen SELECT dipaparkan pada Gambar 3.



Gambar 3. Sintaks Inner Join

Asumsikan contoh tabel berikut yang berelasi one-to-many melalui FK negaraID:

negara	
negaraID	namaNegara
N1	Indonesia
N2	Singapura
N3	Jepang
N4	Korea Selatan
N5	Australia

kota		
kotaID	namaKota	negaraID
K1	Jakarta	N1
K2	Yogyakarta	N1
K3	Tokyo	N3
K4	Osaka	N3
K5	Busan	N4
K6	Sydney	N5
K7	New York	(null)

Sintaks DDLnya adalah sebagai berikut:

```
CREATE TABLE negara (
    negaraID VARCHAR(3) PRIMARY KEY,
    namaNegara VARCHAR(50) NOT NULL
);
CREATE TABLE kota (
    kotaID VARCHAR(3) PRIMARY KEY,
    namaKota VARCHAR(50) NOT NULL,
    negaraID VARCHAR(3),
    CONSTRAINT fkNegara FOREIGN KEY (negaraID) REFERENCES negara(negaraID)
-- Di sini CONSTRAINT FK tidak terlalu berpengaruh
);
```

Untuk menampilkan negara dengan kota yang terkait, kita bisa menuliskan sintaks berikut:

```
SELECT * FROM negara
    INNER JOIN kota ON negara.negaraID = kota.negaraID;
```

	negara.negaraID	namaNegara	kotaID	namaKota	kota.negaraID
1	N1	Indonesia	K1	Jakarta	N1
2	N1	Indonesia	K2	Yogyakarta	N1
3	N3	Jepang	K3	Tokyo	N3
4	N3	Jepang	K4	Osaka	N3
5	N4	Korea Selatan	K5	Busan	N4
6	N5	Australia	K6	Sydney	N5

Untuk menampilkan kolom nama negara dan nama kota saja, kita menuliskan:

```
SELECT negara.namaNegara, kota.namaKota FROM negara
      INNER JOIN kota ON negara.negaraID = kota.negaraID;
```

	namaNegara	namaKota
1	Indonesia	Jakarta
2	Indonesia	Yogyakarta
3	Jepang	Tokyo
4	Jepang	Osaka
5	Korea Selatan	Busan
6	Australia	Sydney

Selain penulisan seperti di atas, Inner Join dapat dituliskan secara implisit dengan menyamakan kunci penghubung di klausa WHERE.

```
SELECT negara.namaNegara, kota.namaKota FROM negara, kota
      WHERE negara.negaraID = kota.negaraID;
```

Kita juga bisa mengaliaskan nama tabel sehingga sintaks menjadi lebih pendek.

```
SELECT n.namaNegara, k.namaKota FROM negara n, kota k
      WHERE n.negaraID = k.negaraID;
```

Apabila tidak menuliskan kondisi pertautan kunci dalam join, maka yang ditampilkan adalah *cross product* dari kedua tabel.

```
SELECT * FROM negara INNER JOIN kota;
```

3. NATURAL JOIN

Natural Join mirip dengan Inner Join. Hanya saja syarat utamanya, nama FK harus sama persis dengan PK di tabel induknya. Dengan kata lain Inner Join mengizinkan nama FK berbeda dengan nama PK-nya. Penulisan sintaks natural join lebih pendek karena tidak harus melengkapi kondisi joinnya.

Contoh:

```
SELECT * FROM negara NATURAL JOIN kota;
```

	negaraID	namaNegara	kotaID	namaKota
1	N1	Indonesia	K1	Jakarta
2	N1	Indonesia	K2	Yogyakarta
3	N3	Jepang	K3	Tokyo
4	N3	Jepang	K4	Osaka
5	N4	Korea Selatan	K5	Busan
6	N5	Australia	K6	Sydney

4. OUTER JOIN

Tipe join ini akan menggabungkan tabel-tabel berdasarkan kunci penghubung, dengan berfokus pada tabel utama dijoinkan dengan tabel kedua dari sisi tertentu. Perbedaannya lagi dengan Inner Join atau Natural Join adalah Outer Join memunculkan data row tabel yang tidak memiliki pasangan di tabel yang dijoinkan. Juga perlu diketahui bahwa MariaDB hanya mengizinkan RIGHT OUTER JOIN / RIGHT JOIN dan LEFT OUTER JOIN / LEFT JOIN saja.

1. RIGHT OUTER JOIN

Sintaksnya querynya adalah:

```
SELECT [kolom] FROM [tabelA] RIGHT JOIN [tabelB]
ON tabelA.pkA = tabelB.fkA;
```

Contoh untuk tabel negara dan kota:

```
SELECT namaNegara, namaKota FROM negara RIGHT JOIN kota
ON negara.negaraID = kota.negaraID;
```

Hasil query:

	namaNegara	namaKota
1	Indonesia	Jakarta
2	Indonesia	Yogyakarta
3	Jepang	Tokyo
4	Jepang	Osaka
5	Korea Selatan	Busan
6	Australia	Sydney
7	<null>	New York

Deskripsikan hasil query di atas sesuai kata-kata Anda sendiri: _____

Perhatikan bahwa kolom yang dipilih melalui klausa SELECT tidak perlu diawali dengan nama tabel **sepanjang** nama atribut/kolom tersebut unik, alias tidak kembar di tabel lain yang terjoin.

Cobalah membalik susunan join menjadi **kota RIGHT JOIN negara** dan amati hasilnya!

2. LEFT OUTER JOIN

Sintaksnya querynya adalah:

```
SELECT [kolom] FROM [tabelA] LEFT JOIN [tabelB]
```

```
ON tabelA.pkA = tabelB.fkA;
```

Contoh untuk tabel negara dan kota:

```
SELECT namaNegara, namaKota FROM negara LEFT JOIN kota  
ON negara.negaraID = kota.negaraID;
```

Hasil query:

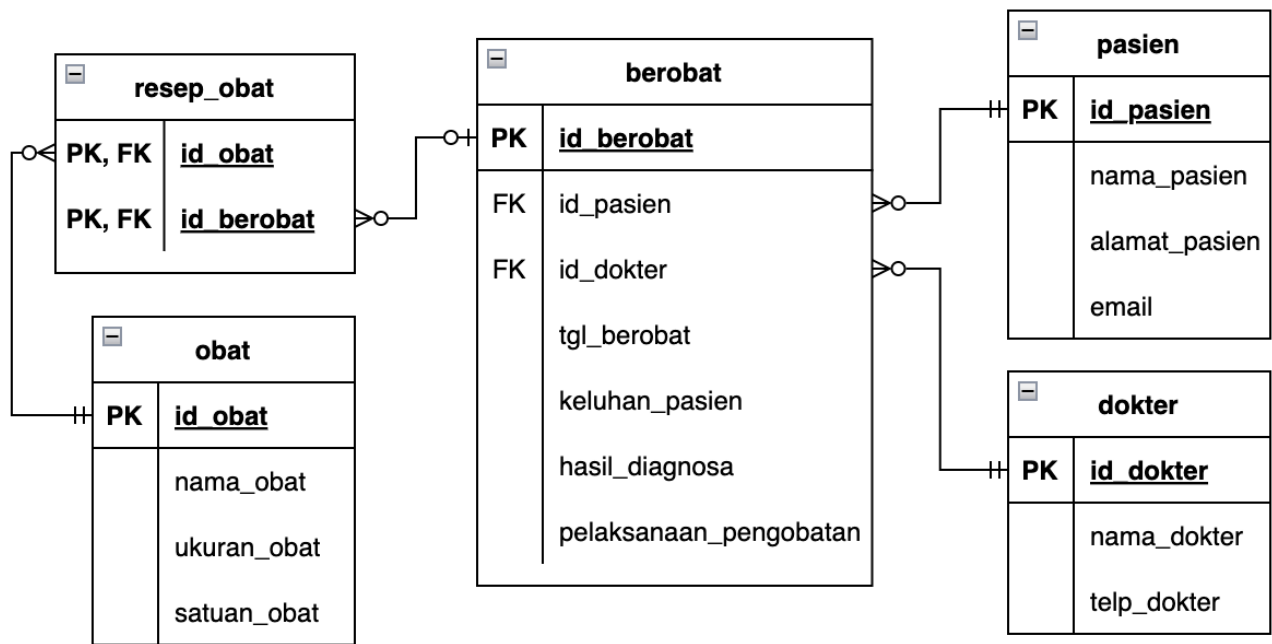
	namaNegara	namaKota
1	Indonesia	Jakarta
2	Indonesia	Yogyakarta
3	Singapura	<null>
4	Jepang	Tokyo
5	Jepang	Osaka
6	Korea Selatan	Busan
7	Australia	Sydney

Deskripsikan hasil query di atas sesuai kata-kata Anda sendiri: _____

Cobalah membalik susunan join menjadi **kota LEFT JOIN negara** dan amati hasilnya!

5. Latihan Mandiri

Berikut ini adalah kasus database klinik.



Berikut adalah Stuktur Tabel DB Klinik:

a. Tabel Obat

Field	Type	Null	Key	Default	Extra
id_obat	varchar(3)	NO	PRI	NULL	
nama_obat	varchar(50)	YES		NULL	
ukuran_obat	int(11)	YES		NULL	
satuan_obat	enum('mg', 'ml', 'g')	YES		NULL	

b. Tabel Pasien

Field	Type	Null	Key	Default	Extra
id_pasien	varchar(3)	NO	PRI	NULL	
nama_pasien	varchar(50)	YES		NULL	
alamat_pasien	varchar(255)	YES		NULL	
email	varchar(50)	YES		NULL	

c. Tabel Dokter

Field	Type	Null	Key	Default	Extra
id_dokter	varchar(3)	NO	PRI	NULL	
nama_dokter	varchar(50)	YES		NULL	
telp_dokter	varchar(15)	YES		NULL	

d. Tabel Berobat

Field	Type	Null	Key	Default	Extra
id_berobat	int(11)	NO	PRI	NULL	auto_increment
id_pasien	varchar(3)	YES	MUL	NULL	
id_dokter	varchar(3)	YES	MUL	NULL	
tgl_berobat	date	YES		NULL	
keluhan_pasien	varchar(255)	YES		NULL	
hasil_diagnosa	varchar(255)	YES		NULL	
pelaksanaan_pengobatan	enum('Rawat Jalan', 'Rawat Inap', 'Rujuk', 'Lainnya')	YES		NULL	

e. Tabel Resep Obat

Field	Type	Null	Key	Default	Extra
id_obat	varchar(3)	NO	PRI	NULL	
id_berobat	int(11)	NO	PRI	NULL	

Isikan data sebagai berikut pada database Klinik Sehat:

a. Tabel obat

id_obat	nama_obat	ukuran_obat	satun_obat
001	Simvastatin	10	mg
002	Sanmol	500	mg
003	Cendo Lyteers Eye Drops	15	ml
004	Counterpain Cool Gel	15	g
005	Neo Kaolana Sirup	120	ml

b. Tabel pasien

id_pasien	nama_pasien	alamat_pasien	email
P01	roger	Jl. Mawar no 10	roger@email.com
P02	rika	Jl. Anggur No 40	rika@email.com
P03	rupawan	Gang Matahari I No 90	rupawan@email.com

c. Tabel dokter

id_dokter	nama_dokter	telp_dokter
D01	Nila Anggia	08127777771
D02	Agata Filiana	08127777772
D03	Rosa Delima	08127777773
D04	Lukas Chrisantyo	08127777774

d. Tabel berobat

id_berobat	id_pasien	id_dokter	tgl_berobat	keluhan_pasien	hasil_diagnosa	pelaksanaan_pengobatan
1	P01	D03	NOW()	pusing, demam	flu	Rawat Jalan
2	P02	D04	NOW()	sering buang air	diare	Rawat Inap
3	P03	D04	NOW()	kecelakaan	patah tulang	Rujuk

e. Tabel resep_obat

id_berobat	id_obat
1	O02
1	O01
2	O05

- Lakukan DDL terhadap tabel-tabel di atas.
 - Tabel Obat,
 - Tabel Pasien,
 - Tabel Dokter
 - Tabel Berobat
 - Tabel Resep_Obat
- Lakukan DML INSERT secepat mungkin.
 - Tabel Obat,
 - Tabel Pasien,
 - Tabel Dokter
 - Tabel Berobat
 - Tabel Resep_Obat
- Tampilkan semua data obat yang pernah diresepkan oleh dokter
- Tampilkan semua obat yang belum pernah diresepkan oleh dokter
- Tampilkan obat yang memiliki satuan 'ml'
- Tampilkan data pengobatan yang memiliki resep obat

id_berobat	id_pasien	id_dokter	tgl_berobat	keluhan_pasien	hasil_diagnosa	pelaksanaan_pengobatan	id_obat	id_berobat
1	P01	D03	2023-10-01	pusing, demam	flu	Rawat Jalan	O01	1
1	P01	D03	2023-10-01	pusing, demam	flu	Rawat Jalan	O02	1
2	P02	D04	2023-10-01	sering buang air	diare	Rawat Inap	O05	2

- Tampilkan semua data pengobatan baik yang memiliki resep obat maupun tidak

id_berobat	id_pasien	id_dokter	tgl_berobat	keluhan_pasien	hasil_diagnosa	pelaksanaan_pengobatan	id_obat	id_berobat
1	P01	D03	2023-10-01	pusing, demam	flu	Rawat Jalan	O01	1
1	P01	D03	2023-10-01	pusing, demam	flu	Rawat Jalan	O02	1
2	P02	D04	2023-10-01	sering buang air	diare	Rawat Inap	O05	2
3	P03	D04	2023-10-01	kecelakaan	patah tulang	Rujuk	NULL	NULL

- Tampilkan semua data pengobatan yang tidak meresepkan obat

id_berobat	id_pasien	id_dokter	tgl_berobat	keluhan_pasien	hasil_diagnosa	pelaksanaan_pengobatan
3	P03	D04	2023-10-01	kecelakaan	patah tulang	Rujuk

9. Tampilkan data pengobatan beserta nama pasien dan nama dokter

id_berobat	nama_dokter	nama_pasien
1	Rosa Delima	roger
2	Lukas Chrisantyo	rika
3	Lukas Chrisantyo	rupawan

10. Tampilkan dokter yang belum pernah mengobati pasien

nama_dokter
Nila Anggia
Agata Filiana

11. Tampilkan obat-obatan yang pernah diresepkan oleh dokter Lukas

id_obat	nama_obat	ukuran_obat	satuan_obat
O05	Neo Kaolana Sirup	120	ml

12. Tampilkan dalam 1 kolom, informasi dokter mengobati sakit seorang pasien

pengobatan
Rosa Delima mengobati keluhan pusing, demam pasien roger
Lukas Chrisantyo mengobati keluhan sering buang air pasien rika
Lukas Chrisantyo mengobati keluhan kecelakaan pasien rupawan

13. Tampilkan nama pasien yang menjalani pengobatan rawat inap

nama_pasien	pelaksanaan_pengobatan
rika	Rawat Inap

14. Tampilkan obat yang perlu disiapkan perawat untuk pasien bernama roger.

id_obat	nama_obat	ukuran_obat	satuan_obat
O01	Simvastatin	10	mg
O02	Sanmol	500	mg

15. Tampilkan nama dokter dan nama pasiennya, tampilkan juga yang belum memiliki pasien. Jika belum punya pasien tampilkan dengan '-'

nama_dokter	nama_pasien
Rosa Delima	roger
Lukas Chrisantyo	rika
Lukas Chrisantyo	rupawan
Nila Anggia	-
Agata Filiana	-

REFERENSI:

W3Resource Inner Join - <https://www.w3resource.com/sql/joins/perform-an-inner-join.php>

W3 Resource Cross Join - <https://www.w3resource.com/sql/joins/cross-join.php>

MariaDB Joins on TechonTheNet - <https://www.techonthenet.com/mariadb/joins.php>

Lynn Beighley, "Head First SQL", 2007, O'Reilly. ISBN: 978-0-596-52684-9