



BASIS DATA

SELECT JOIN

TIM AJAR BASIS DATA JTI-POLINEMA

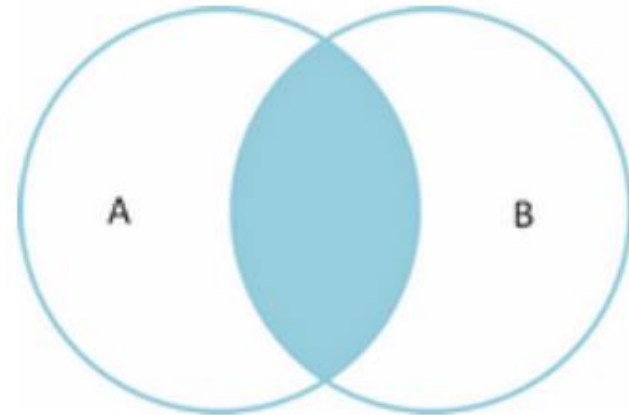
JOIN



- **JOIN** → Mengambil data yang **bersesuaian** dari 2 tabel atau lebih.
 - Dalam 1 statement
 - Ditampilkan sebagai 1 *result set*.
- Menurut syntaxnya ada 2:
 - IMPLICIT JOIN → Tidak ada kata JOIN di statementnya
 - **EXPLICIT JOIN** → Ada kata JOIN. Standar yang baru.
- Join dapat diklasifikasikan ke dalam dua jenis: **INNER JOIN** dan **OUTER JOIN**.
 - Serta 1 lagi, namun jarang digunakan → CROSS JOIN

INNER JOIN

- **INNER JOIN** pada dasarnya adalah menemukan persimpangan (irisan, intersection) antara dua buah tabel atau lebih.
- Cara 1 Explicit Join:
SELECT A1, A2, ..., An
FROM r1
INNER JOIN r2
ON r1.join_key = r2.join_key
- Cara 2: Implicit Join:
SELECT A1, A2, ..., An
FROM r1, r2
WHERE r1.key = r2.key





INNER JOIN

```
[mysql> SELECT * FROM nilai;
```

nim	kode_mk	nilai_huruf
123	DB	A
123	MM	B+
123	PBO	B+
124	DB	B
124	MM	B+
124	PBO	A
125	DB	B+
125	MM	A
125	PBO	A
126	MM	A

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

```
[mysql> SELECT * FROM mahasiswa;
```

nim	nama_mhs
123	Ani
124	Budi
125	Christine
126	Dita
127	Ella

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

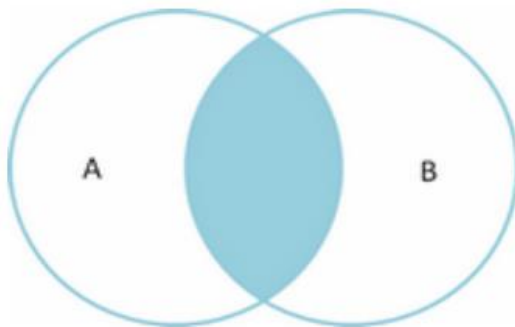
```
mysql> SELECT mhs.nim, mhs.nama_mhs, n.kode_mk, n.nilai_huruf  
-> FROM mahasiswa mhs INNER JOIN nilai n  
-> ON mhs.nim = n.nim;
```

nim	nama_mhs	kode_mk	nilai_huruf
123	Ani	DB	A
123	Ani	MM	B+
123	Ani	PBO	B+
124	Budi	DB	B
124	Budi	MM	B+
124	Budi	PBO	A
125	Christine	DB	B+
125	Christine	MM	A
125	Christine	PBO	A
126	Dita	MM	A

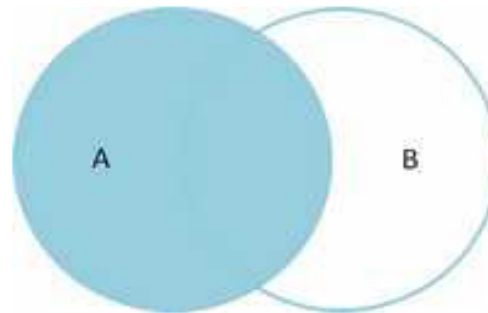
```
10 rows in set (0.00 sec)
```

OUTER JOIN

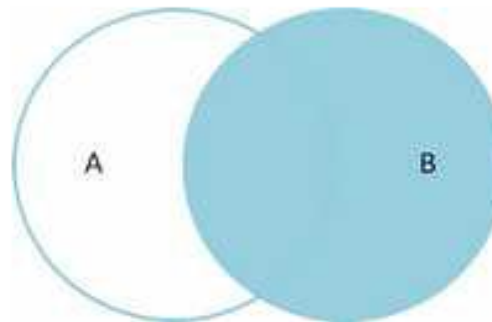
- **OUTER JOIN** dibagi menjadi 2:
 - 1. **LEFT** OUTER JOIN
 - 2. **RIGHT** OUTER JOIN
- Pada intinya adalah mengembalikan:
 - Baris-baris yang bersesuaian + semua baris yang **tidak** bersesuaian di tabel kanan/kiri



INNER JOIN



LEFT OUTER JOIN



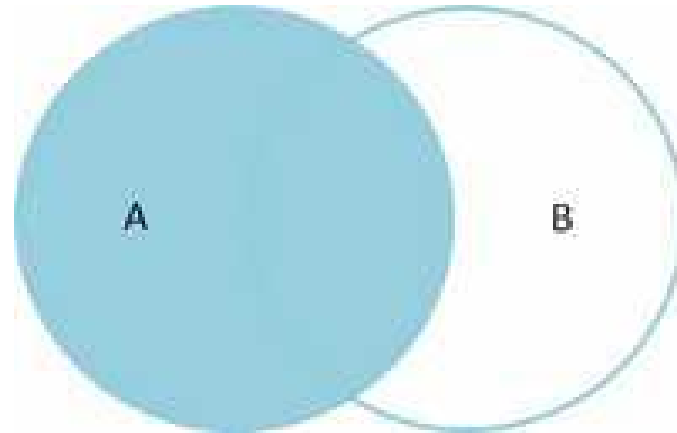
RIGHT OUTER JOIN

LEFT OUTER JOIN

LEFT OUTER JOIN (atau LEFT JOIN) mengembalikan semua nilai dari tabel kiri ditambah dengan nilai dari tabel kanan yang sesuai (atau NULL jika tidak ada nilai yang sesuai).

Syntax

```
SELECT A1, A2, ..., An  
FROM r1  
LEFT OUTER JOIN r2  
ON r1.join_key = r2.join_key
```





LEFT OUTER JOIN

- Informasi yang ditampilkan → **Semua** data dari **tabel** di sebelah **kiri** kata JOIN beserta **pasangannya** dari tabel sebelah **kanan**.
 - Jika ada yang tidak memiliki pasangan, maka akan dipasangkan dengan nilai **NULL**

```
[mysql> SELECT * FROM matakuliah;
```

kode_mk	nama_mk	kode_dosen
DB	Database	ATW
MM	Multimedia	EAP
PB0	Pemrograman Berorientasi Objek	PRM
SPK	Sistem Pendukung Keputusan	ATW

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

```
[mysql> SELECT * FROM nilai;
```

nim	kode_mk	nilai_huruf
123	DB	A
123	MM	B+
123	PB0	B+
124	DB	B
124	MM	B+
124	PB0	A
125	DB	B+
125	MM	A
125	PB0	A
126	MM	A

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



LEFT OUTER JOIN

```
mysql> SELECT mk.kode_mk, mk.nama_mk, n.nim, n.nilai_huruf FROM matakuliah mk  
LEFT OUTER JOIN nilai n ON mk.kode_mk = n.kode_mk;
```

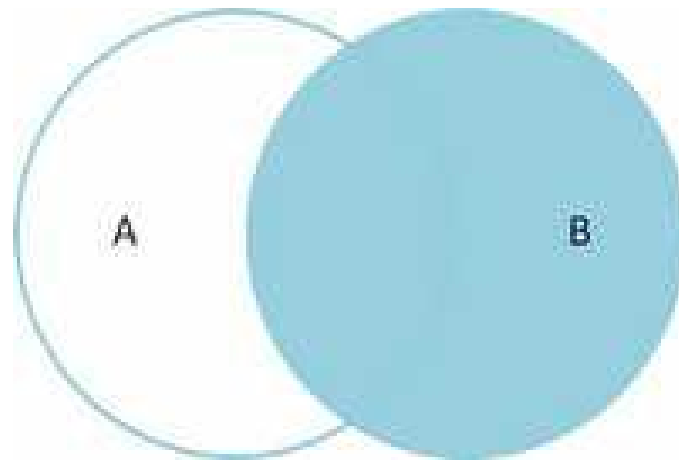
kode_mk	nama_mk	nim	nilai_huruf
DB	Database	123	A
DB	Database	124	B
DB	Database	125	B+
MM	Multimedia	123	B+
MM	Multimedia	124	B+
MM	Multimedia	125	A
MM	Multimedia	126	A
PBO	Pemrograman Berorientasi Objek	123	B+
PBO	Pemrograman Berorientasi Objek	124	A
PBO	Pemrograman Berorientasi Objek	125	A
SPK	Sistem Pendukung Keputusan	NULL	NULL

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


RIGHT OUTER JOIN

Right outer join (atau right join) pada dasarnya sama seperti left join, namun dalam bentuk terbalik—kanan dan kiri.

- Syntax
SELECT A1, A2, ..., An
FROM r1
RIGHT OUTER JOIN r2
ON r1.join_key = r2.join_key





RIGHT OUTER JOIN

```
mysql> SELECT mk.kode_mk, mk.nama_mk, n.nim, n.nilai_huruf FROM matakuliah mk  
RIGHT OUTER JOIN nilai n ON mk.kode_mk = n.kode_mk;
```

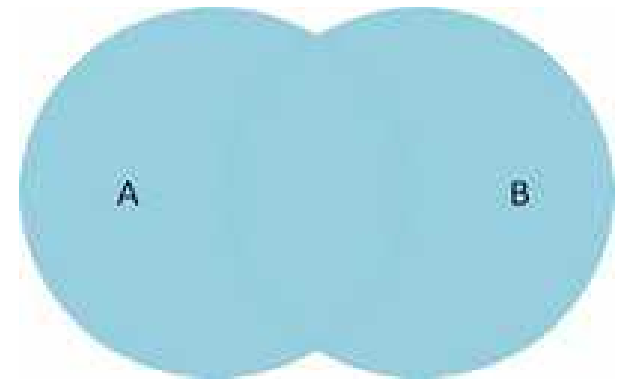
kode_mk	nama_mk	nim	nilai_huruf
DB	Database	123	A
DB	Database	124	B
DB	Database	125	B+
MM	Multimedia	123	B+
MM	Multimedia	124	B+
MM	Multimedia	125	A
MM	Multimedia	126	A
PBO	Pemrograman Berorientasi Objek	123	B+
PBO	Pemrograman Berorientasi Objek	124	A
PBO	Pemrograman Berorientasi Objek	125	A

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

OUTER JOIN

- **Full outer join (atau full join)** pada hakekatnya merupakan kombinasi dari left dan right join.

- Syntax
SELECT A1, A2, ..., An
FROM r1
FULL OUTER JOIN r2
ON r1.join_key = r2.join_key



- MySQL tidak support syntax FULL OUTER JOIN!
- Cara:
{LEFT OUTER JOIN}
UNION ALL
{RIGHT OUTER JOIN}



FULL JOIN

- Informasi yang ditampilkan adalah dari kedua tabel baik kiri maupun kanan statement full join dengan pasangannya.
 - Bagi data yang tidak memiliki pasangan akan diberi pasangan dengan NULL
- Contoh
SELECT M.NIM, M.nama,
 K.IdKelas, K.Nilai
FROM Mahasiswa M Full join KRS K
On M.NIM = K.NIM
WHERE Nama like '%a%'
- Ingat! Di MySQL (dan MariaDB?) saat ini belum mendukung sintaksis FULL JOIN.
 - Gunakan UNION ALL untuk mengakalinya.



SYNTAX

SELECT

Tabel1.Kolom1, Tabel1.Kolom2, ..., Tabel1.Kolomn,
Tabel2.Kolom1, Tabel2.Kolom2, ..., Tabel2.Kolomn,

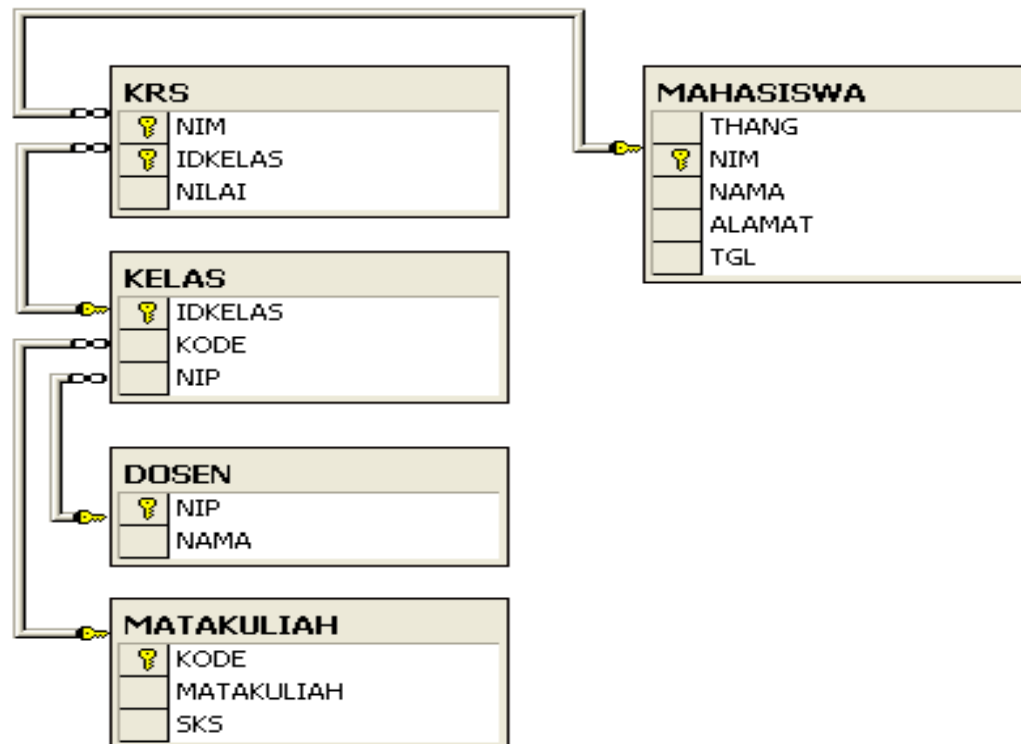
...

Tabeln.Kolom1, Tabeln.Kolom2, ..., Tabeln.Kolomn

FROM Tabel1, Tabel2,..., Tabeln

[WHERE kondisi];

RELASI ANTAR TABEL





PENGGUNAAN ALIAS

SELECT alias1.kolom, alias2.kolom
FROM tabel1 alias1, tabel2 alias2
WHERE kondisi

Contoh:

SELECT M.NIM, M.nama,
K.IdKelas, K.Nilai
FROM Mahasiswa M, KRS K

RANGKUMAN & CONTOH SYNTAX



1. JOIN atau INNER JOIN

Menggabungkan dua tabel dimana diantara dua tabel datanya bersesuaian.

2. LEFT JOIN atau LEFT OUTER JOIN

Menggabungkan dua tabel dimana diantara dua tabel datanya bersesuaian dan juga semua record pada tabel sebelah kiri.

3. RIGHT JOIN atau RIGHT OUTER JOIN

Menggabungkan dua tabel dimana diantara dua tabel datanya bersesuaian dan juga semua record pada tabel sebelah kanan.



RANGKUMAN & CONTOH SYNTAX

```
SELECT Nilai.NIM, NAMA_MHS, KD_MK, MID  
FROM Nilai INNER JOIN Mahasiswa  
ON Nilai.NIM = Mahasiswa.NIM
```

Hasil :

NIM	NAMA_MHS	KD_MK	MID
10296832	Nurhayati	KK021	60
10296126	Astuti	KD132	70
31296500	Budi	KK021	55
41296525	Prananigru	KU122	90
21196353	m	KU122	75
50095487	Quraish Pipit	KD132	80



RANGKUMAN & CONTOH SYNTAX

```
SELECT Mahasiswa.NIM, NAMA_MHS, KD_MK, MID
FROM Mahasiswa LEFT OUTER JOIN Nilai
ON Nilai.NIM = Mahasiswa.NIM
```

Hasil :

NIM	NAMA_MHS	KD_MK	MID
10296832	Nurhayati	KK021	60
10296126	Astuti	KD132	70
31296500	Budi	KK021	55
41296525	Prananigru	KU122	90
21196353	m	KU122	75
50095487	Quraish	KD132	80
10296001	Pipit	-	-
21198002	Fintri Julizar	-	-



RANGKUMAN & CONTOH SYNTAX

```
SELECT Mahasiswa.NIM, NAMA_MHS, KD_MK, MID
FROM Nilai RIGHT OUTER JOIN Mahasiswa
ON Nilai.NIM = Mahasiswa.NIM
```

Hasil :

NIM	NAMA_MHS	KD_MK	MID
10296832	Nurhayati	KK021	60
10296126	Astuti	KD132	70
31296500	Budi	KK021	55
41296525	Prananigru	KU122	90
21196353	m	KU122	75
50095487	Quraish	KD132	80
10296001	Pipit	-	-
21198002	Fintri Julizar	-	-



TUGAS

NO_DEP	NAMA	LOKASI
1	KEUANGAN	MALANG
2	PEMASARAN	MALANG
3	PRODUKSI	MALANG
4	TENAGA KERJA	MALANG
5	IT	MALANG

Gambar 1.1 departemen

NO_PRO	NAMA	LOKASI	NO_DEP
1	DESAIN GAMBAR	MALANG	5
2	PEMASARAN BARANG	MALANG	3
3	JARINGAN KOMPUTER	SURABAYA	4

Gambar 1.2 proyek

NO_KER	NO_KTP	NO_PRO	LAMA_JAM
1	20104260001	1	36
2	20104340002	1	36
3	20104290001	2	60
4	20104320002	2	60
5	20104270002	3	42

Gambar 1.3 kerja

NO_KTP	NAMA	JENIS_KEL	ALAMAT	TGL_LAHIR	GAJI	NO_DEP
20104260001	DODO	1	MALANG	12-DEC-87	2000000	1
20104270002	NINI	2	SURABAYA	13-DEC-89	2000000	2
20104280002	NORMAN	1	MALANG	01-JAN-79	2500000	2
20104290001	LALA	2	MALANG	11-MAR-90	1750000	5
20104300002	DINDA	2	KEDIRI	15-FEB-80	2000000	3
20104310001	SONI	1	TULUNGAGUNG	29-OCT-70	3000000	4
20104320002	NONO	1	SURABAYA	17-SEP-86	2600000	5
20104330001	DITA	2	MALANG	01-MAR-74	2000000	3
20104340002	HARI	1	SURABAYA	01-NOV-69	2900000	5
20104350002	NINO	1	MALANG	10-NOV-72	2000000	1

Gambar 1.4 pegawai

NO_TANG	NO_KTP	NAMA	JENIS_KEL	TGL_LAHIR	HUBUNGAN
1	20104350002	HERA	2	12-NOV-72	ISTRI
2	20104350002	RAHEL	2	17-AUG-92	ANAK
3	20104350002	ROKI	1	12-NOV-98	ANAK
4	20104310001	SHINTA	2	20-MAY-72	ISTRI
5	20104290001	DINI	2	01-JUL-90	ANAK

Gambar 1.5 tanggungan

TUGAS



1. Buat query dengan menggunakan konsep JOIN untuk menyelesaikan soal dan menghasilkan output berikut : [**Database Perusahaan**]

- a) Tampilkan nama departemen yang mempunyai pegawai termuda, sertakan nama pegawai dan tanggal lahirnya.

DEPARTEMEN	PEGAWAI	TGL_LAHIR
IT	LALA	11-03-1990



- b) Tampilkan nama departemen dan jumlah pegawai yang bekerja pada masing-masing departemen tersebut.

NAMA	JML_PEGAWAI
IT	3
KELANGAN	2
PRODUKSI	2
PEMASARAN	2
TENAGA KERJA	1

- c) Tampilkan semua informasi mengenai proyek dan di departemen mana proyek tersebut dikerjakan.

NO_PRO	NAMA	LOKASI	NO_DEP	NO_DEP	NAMA	LOKASI
1	DESAIN GAMBAR	MALANG	5	5	IT	MALANG
2	PEMASARAN BARANG	MALANG	3	3	PRODUKSI	MALANG
3	JARINGAN KOMPUTER	SURABAYA	4	4	TENAGA KERJA	MALANG



TERIMAKASIH



REFERENSI

- Dwi Puspitasari, S.Kom, “**Buku Ajar Dasar Basis Data**”, *Program Studi Manajemen Informatika Politeknik Negeri Malang*, 2012.
- Fathansyah, “**Basisdata Revisi Kedua**”, Bandung: Informatika, 2015.