

**LAPORAN PRAKTIKUM**  
**MATA KULIAH PRAKTIKUM BASIS DATA**

Dosen Pengampu : Farid Angga Pribadi, S.Kom., M.Kom.

**PERTEMUAN 13 : SELECT JOIN**



Nama : Yonanda Mayla Rusdiaty

NIM : 2341760184

Prodi : D-IV Sistem Informasi Bisnis

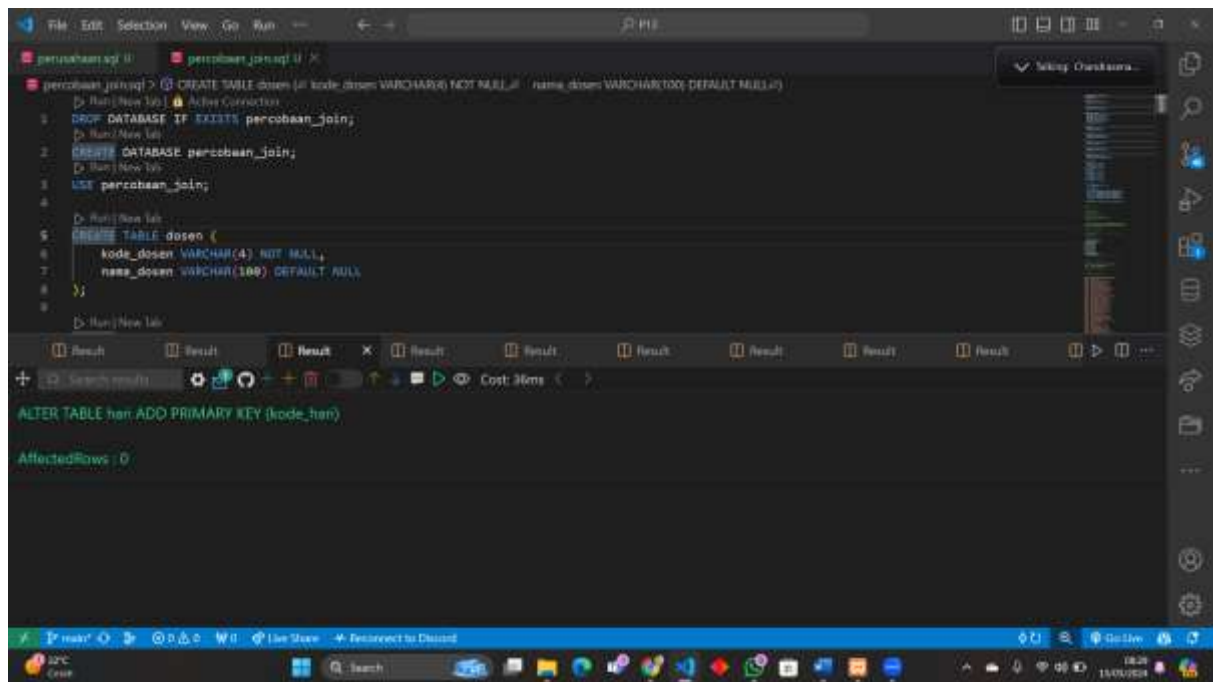
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**

**2024**

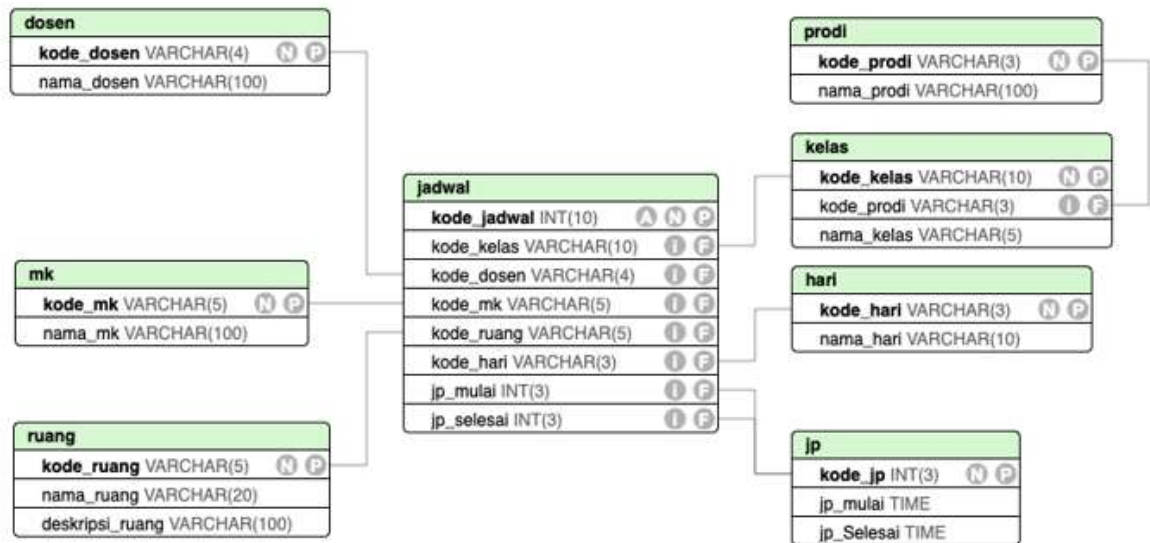
## PRAKTIKUM 1

1. Bersama jobsheet ini disertakan sebuah file bernama percobaan\_join.sql, eksekusilah file tersebut pada server MySQL Anda dengan cara apapun yang Anda bisa.

Pada contoh di bawah ini, SQL dieksekusi dengan menggunakan perintah SOURCE melalui MySQL Shell. Anda juga dapat mengimpor SQL tersebut melalui PHPMyAdmin atau MySQL Workbench, atau tools yang lain yang Anda sukai.



2. Jika Anda telah berhasil mengimpor/mengeksekusi/menjalankan file percobaan\_join.sql tersebut, maka di server MySQL Anda akan dibuatkan database dengan struktur seperti berikut.



- Pastikan pada database Anda terdapat tabel-tabel seperti di bawah ini dengan menjalankan SQL dibawah. Jika tabel-tabelnya sudah sesuai, lanjutkan ke Praktikum – Bagian 2.

```

SHOW TABLES; 13ms
--
Tables_in_percobaan_jo
varchar
--
dosen
hari
jadwal
jp
kelas
mk
prodi
ruang
  
```

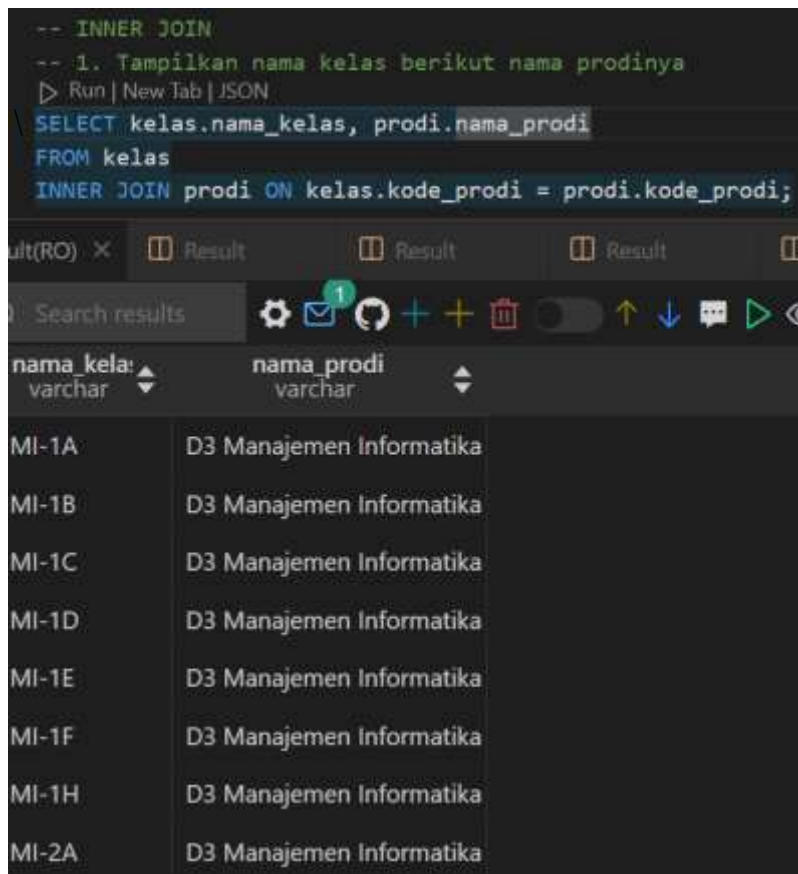
## PRAKTIKUM 2: INNER JOIN

1. Apabila kita ingin menampilkan data yang kolom-kolomnya terdapat pada tabel yang berbeda, maka kita dapat menggunakan sintaksis INNER JOIN. Sintaksis ini akan menampilkan nilai kolom pada 2 atau lebih tabel yang saling bersesuaian dalam 1 baris.

Contoh: “Tampilkan nama kelas berikut nama prodinya!”

Solusi: Jalankan query berikut ini, dan akan ditampilkan 52 baris nama kelas berikut nama prodinya yang bersesuaian (kolom kode\_prodi di tabel kelas sama nilainya dengan kolom kode\_prodi di tabel prodi).

```
-- INNER JOIN
-- 1. Tampilkan nama kelas berikut nama prodinya
> Run | New Tab | JSON
SELECT kelas.nama_kelas, prodi.nama_prodi
FROM kelas
INNER JOIN prodi ON kelas.kode_prodi = prodi.kode_prodi;
```



nama_kelas	nama_prodi
MI-1A	D3 Manajemen Informatika
MI-1B	D3 Manajemen Informatika
MI-1C	D3 Manajemen Informatika
MI-1D	D3 Manajemen Informatika
MI-1E	D3 Manajemen Informatika
MI-1F	D3 Manajemen Informatika
MI-1H	D3 Manajemen Informatika
MI-2A	D3 Manajemen Informatika

2. Pada contoh sebelumnya, jika diperhatikan baik-baik, setiap nama kolom yang ingin ditampilkan harus disebutkan nama tabel asalnya dengan notasi dot/titik (nama\_tabel.nama\_kolom). Hal ini masuk akal karena hal tersebut memiliki tujuan untuk menghilangkan ambiguitas karena bisa saja 2 tabel yang berbeda memiliki kolom dengan nama yang sama.

Namun demikian akan sedikit merepotkan apabila kita secara berulang-ulang menuliskan nama tabel di sebelah nama kolom, lagi dan lagi. Apalagi jika nama tabelnya Panjang.

Solusinya adalah dengan menggunakan alias yaitu sintaksis AS. Dengan menggunakan sintaksis ini, query kita akan menjadi lebih singkat namun dengan hasil yang sama. Jalankan SQL berikut untuk mengetahui hasilnya!

```
844
845 -- 2. Menggunakan Alias (AS)
846 SELECT k.nama_kelas, p.nama_prodi
847 FROM kelas AS k
848 INNER JOIN prodi AS p ON k.kode_prodi = p.kode_prodi;
```

Result(RO) × Result Result Result

Search results

	nama_kelas varchar	nama_prodi varchar
> 1	MI-1A	D3 Manajemen Informatika
> 2	MI-1B	D3 Manajemen Informatika
> 3	MI-1C	D3 Manajemen Informatika
> 4	MI-1D	D3 Manajemen Informatika
> 5	MI-1E	D3 Manajemen Informatika
> 6	MI-1F	D3 Manajemen Informatika
> 7	MI-1H	D3 Manajemen Informatika
> 8	MI-2A	D3 Manajemen Informatika

3. Pada contoh sebelumnya kita telah menampilkan 2 kolom yang terletak pada 2 tabel yang berbeda namun penggunaan INNER JOIN tidaklah terbatas pada 2 tabel saja. Kita juga dapat menampilkan data yang lebih banyak dari beberapa tabel sekaligus.

Contoh: “Tampilkan nama dosen berikut kelas yang diajar dan harinya!”

Solusi: Jalankan SQL berikut. Jika benar akan ditampilkan 370 baris. Pada query tersebut dilibatkan 4 tabel yaitu tabel jadwal, dosen, kelas, dan hari.

```
-- 3. JOIN beberapa tabel
-- Tampilkan dosen mengajar di kelas apa pada hari apa
▷ Run | New Tab | JSON | Copy
✓ SELECT
    j.kode_jadwal,
    d.nama_dosen,
    k.nama_kelas,
    n.nama_hari
✓ FROM
    jadwal j
    INNER JOIN dosen d ON j.kode_dosen = d.kode_dosen
    INNER JOIN kelas k ON j.kode_kelas = k.kode_kelas
    INNER JOIN hari n ON j.kode_hari = n.kode_hari; 6ms
```

kode_jadwal int	nama_dosen varchar	nama_kelas varchar	nama_hari varchar
1	Abdul Chalim, SAg., MPd.I	TI-1A	Senin
8	Ade Ismail	MI-2F	Senin
17	Ahmadi Yuli Ananta ST., MM.	MI-1F	Senin
18	Ahmadi Yuli Ananta ST., MM.	TI-2C	Senin
21	Ane Fany Novitasari, SH.MKn.	TI-1D	Senin
22	Ane Fany Novitasari, SH.MKn.	TI-1H	Senin
23	Ane Fany Novitasari, SH.MKn.	TI-1F	Senin
38	Anugrah Nur Rahmanto SSn., MDs.	MI-3F	Senin
48	Arie Rachmad Syulistyo SKom., MKom.	MI-1C	Senin

4. SELECT JOIN juga bisa difilter. Tentu saja dengan menggunakan klausa WHERE.

Contoh: “Tampilkan nama kelas berikut nama prodinya, hanya untuk kelas yang A saja!”

Solusi: Jalankan SQL berikut ini. Jika benar akan ditampilkan 7 baris yaitu semua kelas dari kelas 1-4 di masing-masing prodi yang namanya diakhiri huruf “A”

```
-- 4. INNER JOIN dengan WHERE
-- Menampilkan nama kelas dan nama prodi,
-- yang nama kelasnya adalah kelas 'A'
▷ Run | New Tab | JSON
SELECT k.nama_kelas, p.nama_prodi
FROM kelas k
INNER JOIN prodi p ON k.kode_prodi = p.kode_prodi
WHERE k.nama_kelas LIKE '%A%'; 2ms
```

nama_kelas varchar	nama_prodi varchar
MI-1A	D3 Manajemen Informatika
MI-2A	D3 Manajemen Informatika
MI-3A	D3 Manajemen Informatika
TI-1A	D4 Teknik Informatika
TI-2A	D4 Teknik Informatika
TI-3A	D4 Teknik Informatika
TI-4A	D4 Teknik Informatika



## PRAKTIKUM 3 : OUTER JOIN

1. Jika INNER JOIN hanya menampilkan baris-baris pada 2 atau lebih tabel yang saling bersesuaian. Maka untuk menampilkan data pada tabel yang saling bersesuaian ditambah yang tidak bersesuaian, kita dapat menggunakan sintaksis OUTER JOIN.

OUTER JOIN dibagi menjadi 2:

- LEFT OUTER JOIN dan;
- RIGHT OUTER JOIN

OUTER JOIN pada umumnya berguna untuk mengecek data yang tidak ada pasangannya di tabel yang di-JOIN-kan.

Contoh: “Tampilkan data semua kelas berikut nama prodinya, beserta kelas yang tidak ada prodinya!”

Solusi: Jalankan query berikut. Jika benar akan ditampilkan 57 baris data dimana 52 baris adalah nama kelas yang ada prodinya, dan 5 baris sisanya adalah nama kelas yang tidak terdaftar di prodi manapun.

```
-- 1. LEFT OUTER JOIN
-- Menampilkan kelas yang belum ada prodinya
Run | New Tab | JSON | Copy
SELECT
k.nama_kelas,
p.nama_prodi
FROM kelas k
LEFT OUTER JOIN prodi p ON k.kode_prodi = p.kode_prodi;
```

nama_kelas varchar	nama_prodi varchar
TI-4A	D4 Teknik Informatika
TI-4B	D4 Teknik Informatika
TI-4C	D4 Teknik Informatika
TI-4D	D4 Teknik Informatika
TI-4E	D4 Teknik Informatika
TI-4F	D4 Teknik Informatika
TI-4G	D4 Teknik Informatika
TRM-1A	(NULL)
TRM-1B	(NULL)
TRM-1C	(NULL)
S2TI-A	(NULL)

2. Apabila hanya ingin menampilkan data yang tidak ada pasangannya saja, maka kita bisa menggunakan filter melalui penambahan klausa **WHERE**.

Query berikut akan menampilkan data seperti sebelumnya, namun hanya yang tidak ada pasangannya saja.

```
-- 2. OUTER JOIN dengan WHERE
-- Menampilkan kelas yang belum ada prodinya saja
Run | New Tab | JSON | Copy
SELECT
  k.nama_kelas,
  p.nama_prodi
FROM
  kelas k
  LEFT OUTER JOIN prodi p ON k.kode_prodi = p.kode_prodi
WHERE
  p.kode_prodi IS NULL; 2ms
```

nama_kelas varchar	nama_prodi varchar
TRM-1A	(NULL)
TRM-1B	(NULL)
TRM-1C	(NULL)
S2TI-A	(NULL)
S2TI-B	(NULL)

- Selain LEFT OUTER JOIN juga ada RIGHT OUTER JOIN. Keduanya sama-sama menampilkan data yang bersesuaian ditambah yang tidak bersesuaian. Bedanya adalah letak data yang TIDAK NULL-nya di sebelah mana. Tabel kiri atau tabel kanan.

Bagaimana menentukan tabel kiri dan tabel kanan?

- Tabel kiri adalah yang ditulis di sebelah KIRI (SEBELUM) kata-kata JOIN.

- Tabel kanan adalah yang ditulis di sebelah KANAN (SESUDAJH) kata-kata JOIN.

LEFT OUTER JOIN → NULL-nya di tabel kanan, data yang lengkap di tabel KIRI (LEFT)

RIGHT OUTER JOIN → NULL-nya di tabel kiri, data yang lengkap di tabel KANAN (RIGHT)

Contoh: “Terdapat prodi baru yang belum ada kelasnya, tampilkan nama kelas berikut nama prodinya serta nama prodi-prodi baru yang belum ada kelasnya tersebut!”

Solusi: Jalankan query berikut. Apabila benar akan ditampilkan 55 baris dimana 52 baris adalah data yang bersesuaian (prodi dan nama kelasnya masing-masing), sedangkan 3 baris sisanya adalah prodi baru yang belum ada kelasnya. Perhatikan data yang lengkap ada di kolom nama\_prodi yang merupakan kolom dari tabel prodi yang ditulis di sebelah KANAN (RIGHT) dari kata JOIN pada sintaksis SQL-nya.

```
-- 3. RIGHT OUTER JOIN
-- sama dengan LEFT OUTER JOIN, hanya beda tempat NULL-nya saja
-- LEFT OUTER JOIN --> yang penuh di kiri (LEFT)
-- RIGHT OUTER JOIN --> yang penuh di kanan (RIGHT)
Run | New Tab | JSON | Copy
SELECT
  k.nama_kelas,
  p.nama_prodi
FROM
  kelas k
RIGHT OUTER JOIN prodi p ON k.kode_prodi = p.kode_prodi; 1ms
```

nama_kela: varchar	nama_prodi varchar
TI-4A	D4 Teknik Informatika
TI-4B	D4 Teknik Informatika
TI-4C	D4 Teknik Informatika
TI-4D	D4 Teknik Informatika
TI-4E	D4 Teknik Informatika
TI-4F	D4 Teknik Informatika
TI-4G	D4 Teknik Informatika
(NULL)	D4 Sistem Integritas Tinggi
(NULL)	D4 Kecerdasan Buatan dan
(NULL)	D4 Sistem Informasi Bisnis

4. Jika kit ingin menampilkan semua data yang bersesuaian, ditambah dengan data yang tidak bersesuaian di tabel KANAN dan KIRI sekaligus, maka kita dapat menggunakan sintaksis FULL JOIN.

Pada beberapa DBMS tertentu, sintaksis eksplisit FULL JOIN telah didukung, namun pada MySQL, sintaksis ini belum didukung.

Untuk mengakalnya kita dapat menggunakan sintaksis UNION ALL yang akan menggabungkan 2 buah himpunan hasil SELECT yang berbeda.

Namun jangan lupa bahwa:

- pada sintaksis ini, kedua buah hasil SELECT harus memiliki jumlah kolom yang sama. Jika tidak, maka datanya tidak akan dapat ditampilkan.
- Semikolon (;) harus diletakkan sekali saja di akhir statement SELECT yang paling belakang.
- Jangan gunakan UNION saja karena data yang sama (duplikat) akan dihilangkan.

Jalankan query berikut ini untuk menampilkan hasil FULL join terhadap tabel kelas dan tabel prodi. Apabila benar, maka akan ditampilkan sebanyak 112 baris data dengan rincian:

- 104 data yang lengkap ada prodi dan kelasnya.
- 5 baris data kelas yang tidak ada prodinya
- 3 baris nama prodi yang tidak ada kelasnya

```
-- 4. FULL OUTER JOIN
-- Gabungan LEFT OUTER JOIN dan RIGHT OUTER JOIN
Run | New Tab | JSON | Copy
SELECT
    k.nama_kelas,
    p.nama_prodi
FROM kelas k
    LEFT OUTER JOIN prodi p ON k.kode_prodi = p.kode_prodi
UNION ALL
SELECT
    k.nama_kelas,
    p.nama_prodi
FROM kelas k
    RIGHT OUTER JOIN prodi p ON p.kode_prodi = k.kode_prodi;
```

nama_kelas varchar	nama_prodi varchar
TI-4D	D4 Teknik Informatika
TI-4E	D4 Teknik Informatika
TI-4F	D4 Teknik Informatika
TI-4G	D4 Teknik Informatika
TRM-1A	(NULL)
TRM-1B	(NULL)
TRM-1C	(NULL)
S2TI-A	(NULL)
S2TI-B	(NULL)

## PRAKTIKUM 4 : INNER JOIN Implisit dan CROSS JOIN

1. Sintaksis JOIN yang kita pelajari sebelumnya merupakan sintaksis ANSI SQL yang lebih baru. Sekedar pengetahuan saja, bahwa sebelum distandarkannya format sintaksis tersebut, sebelumnya JOIN dilakukan dengan menggunakan sintaksis yang tidak ada kata JOIN-nya. Format ini disebut sebagai IMPLICIT JOIN. Sedangkan format sintaksis kita sebelumnya disebut sebagai EXPLICIT JOIN.

Kita dianjurkan untuk menggunakan format yang baru, yaitu yang ada kata JOIN-nya karena cenderung lebih jelas dan menghindari terjadinya kesalahan maksud pada SQL yang kita tulis. Dengan menuliskan kata-kata JOIN, berarti kita secara sadar memang ingin menampilkan data dari 2 tabel atau lebih.

Namun demikian untuk sekedar pengetahuan Anda, jalankan sintaksis SQL berikut, hasilnya akan sama dengan hasil pada Praktikum – Bagian 1 Langkah 1.

Perhatikan pada sintaksis ini tidak ada kata-kata “JOIN”-nya.

```
-- 1. SELECT JOIN Implisit
-- Tidak ada kata - kata JOIN pada SQL-nya
▷ Run | New Tab | JSON | Copy
SELECT
    k.nama_kelas,
    p.nama_prodi
FROM kelas k, prodi p
WHERE k.kode_prodi = p.kode_prodi; 5ms
```



nama_kelas varchar	nama_prodi varchar
MI-1A	D3 Manajemen Informatika
MI-1B	D3 Manajemen Informatika
MI-1C	D3 Manajemen Informatika
MI-1D	D3 Manajemen Informatika
MI-1E	D3 Manajemen Informatika
MI-1F	D3 Manajemen Informatika
MI-1H	D3 Manajemen Informatika
MI-2A	D3 Manajemen Informatika

2. INNER JOIN Implisit juga bisa diberikan alias nama tabel agar tidak terlalu Panjang.

Alias nama tabel dituliskan pada klausa FROM, sama seperti pada Explicit JOIN.

Jalankan query berikut. Hasilnya akan sama dengan langkah sebelumnya, namun dengan penulisan SQL yang lebih singkat.

```
-- 2. INNER JOIN implisit dengan alias (AS)
```

Run | New Tab | JSON | Copy

```
SELECT
```

```
    k.nama_kelas,
```

```
    p.nama_prodi
```

```
FROM kelas k, prodi p
```

```
WHERE k.kode_prodi = p.kode_prodi; 2ms
```

> 1	MI-1A	D3 Manajemen Informatika
> 2	MI-1B	D3 Manajemen Informatika
> 3	MI-1C	D3 Manajemen Informatika
> 4	MI-1D	D3 Manajemen Informatika
> 5	MI-1E	D3 Manajemen Informatika
> 6	MI-1F	D3 Manajemen Informatika
> 7	MI-1H	D3 Manajemen Informatika
> 8	MI-2A	D3 Manajemen Informatika

3. Selain INNER JOIN, OUTER JOIN, dan FULL JOIN, terdapat satu jenis JOIN lagi yaitu CROSS JOIN.

CROSS JOIN akan menampilkan kombinasi satu-satu dari setiap kolom pada semua tabel tanpa memperdulikan kesesuaian antar tabel atau tidak.

Jumlah baris yang dihasilkan adalah perkalian jumlah baris pada kedua buah tabel yang di CROSS-kan.

```
-- 3. CROSS JOIN Explisit
Run | New Tab | JSON | Copy
SELECT
    k.nama_kelas,
    p.nama_prodi
FROM kelas k
CROSS JOIN prodi p; 3ms
```

nama_kelas varchar	nama_prodi varchar
MI-1A	D3 Manajemen Informatika
MI-1A	D4 Teknik Informatika
MI-1A	D4 Sistem Integritas Tinggi
MI-1A	D4 Kecerdasan Buatan dan Robotika
MI-1A	D4 Sistem Informasi Bisnis
MI-1B	D3 Manajemen Informatika
MI-1B	D4 Teknik Informatika
MI-1B	D4 Sistem Integritas Tinggi
MI-1B	D4 Kecerdasan Buatan dan Robotika

4. CROSS JOIN juga dapat dilakukan secara IMPLICIT yaitu dengan tanpa memberikan WHERE pada sintaksis JOIN IMPLICIT.

Jalankan query di bawah ini, hasilnya akan sama dengan langkah sebelumnya.

```
-- 4. CROSS JOIN Implisit
Run | New Tab | JSON
SELECT
    k.nama_kelas,
    p.nama_prodi
FROM kelas k, prodi p; 4ms
```

nama_kelas varchar	nama_prodi varchar
MI-1A	D3 Manajemen Informatika
MI-1A	D4 Teknik Informatika
MI-1A	D4 Sistem Integritas Tinggi
MI-1A	D4 Kecerdasan Buatan dan Robotika
MI-1A	D4 Sistem Informasi Bisnis
MI-1B	D3 Manajemen Informatika
MI-1B	D4 Teknik Informatika
MI-1B	D4 Sistem Integritas Tinggi
MI-1B	D4 Kecerdasan Buatan dan Robotika

## TUGAS

1. Jalankan semua SQL pada praktikum-praktikum di atas, pahami maksudnya dan Screenshot-lah hasilnya!

**Jawab :** Outputnya sudah di screenshot di atas

2. Tampilkan nama dosen berikut mata kuliah yang mereka ampu (186 baris) dengan ketentuan:
  - a. Tidak ada data yang duplikat
  - b. Urut berdasarkan nama dosen dari A-Z.
  - c. Nama kolom yang ditampilkan harus sesuai dengan contoh di bawah.

**Jawab :**

```
-- TUGAS
-- no 1 tampilkan dosen pengampu dan nama mata kuliah
▶ Run | New Tab | JSON | Copy
SELECT DISTINCT
    d.nama_dosen,
    m.nama_mk
FROM dosen d
INNER JOIN jadwal j ON d.kode_dosen = j.kode_dosen
INNER JOIN mk m ON j.kode_mk = m.kode_mk
ORDER BY d.nama_dosen ASC; 7ms
```

	nama_dosen varchar	nama_mk varchar
> 75	Vivin Ayu Lestari, SPd.	Struktur Data
> 76	Widaningsih Condrowardhani, SH., MH.	Etika Profesi Bidang TI
> 77	Wilda Imama Sabilla, S.Kom., M.Kom.	E-Business
> 78	Wilda Imama Sabilla, S.Kom., M.Kom.	Praktikum Basis Data
> 79	Wilda Imama Sabilla, S.Kom., M.Kom.	Basis Data
> 80	Yoppy Yunhasnawa, SST., MSc.	Praktikum Basis Data
> 81	Yoppy Yunhasnawa, SST., MSc.	Teknologi Data
> 82	Yoppy Yunhasnawa, SST., MSc.	Basis Data
> 83	Yuri Ariyanto, SKom., MKom.	Proyek 1_P2
> 84	Yuri Ariyanto, SKom., MKom.	Proyek 1_P1
> 85	Yuri Ariyanto, SKom., MKom.	Manajemen Proyek
> 86	Zulmy Faqihuddin Putera, S.Pd., M.Pd	Bahasa Indonesia

3. Tampilkan ruang yang digunakan untuk perkuliahan pada hari 'Selasa' berikut jam-nya (79 baris) dengan ketentuan:
- Harus ditampilkan ruangan tersebut dipakai mulai dari jam berapa sampai dengan jam berapa.
  - Nama kolom yang ditampilkan harus sesuai contoh di bawah.

**Jawab :**

```
-- no 2 tampilkan kode ruang, nama ruang, hari dipakai, dari jam, sampai jam Pada hari selasa
Run | New Tab | JSON | Copy
SELECT |
    r.kode_ruang,
    r.nama_ruang,
    h.nama_hari,
    j.jp_mulai,
    j.jp_selesai
FROM ruang r
INNER JOIN jadwal j ON r.kode_ruang = j.kode_ruang
INNER JOIN hari h ON j.kode_hari = h.kode_hari
WHERE h.nama_hari = 'Selasa';
```

kode_ruang varchar	nama_ruang varchar	nama_hari varchar	jp_mulai int	jp_selesai int
0503	RT03	Selasa	4	6
0503	RT03	Selasa	4	6
0705	LPR5	Selasa	8	12
0506	RT06	Selasa	1	4
0503	RT03	Selasa	1	3
0503	RT03	Selasa	1	3
0806	RT10	Selasa	1	3
0707	LKJ1	Selasa	4	6
0707	LKJ1	Selasa	4	6

4. Tampilkan dosen yang tidak mendapatkan jadwal mengajar! Catatan: Nama kolom harus sesuai contoh.

```
-- no 3 tampilkan dosen yang tidak mendapatkan jadwal mengajar
Run | New Tab | JSON | Copy
SELECT
  d.nama_dosen,
  j.kode_jadwal
FROM dosen d
LEFT OUTER JOIN jadwal j ON d.kode_dosen = j.kode_dosen
WHERE j.kode_jadwal IS NULL;
```

nama_dosen varchar	kode_jadwal int
Kamado Tanjiro, S.Kom., M.Kom.	(NULL)
Rei Ayanami, S.ST., M.Sc.	(NULL)
Soryu Asuka Langley, M.Eng, Ph.D.	(NULL)

5. Tampilkan nama mata kuliah berikut dosen pengampunya berikut (192 baris) yang tidak ada dosen pengampunya! Ketentuan:
- Nama Kolom harus sesuai contoh
  - Anda bisa menggunakan RIGHT OUTER JOIN agar lebih mudah

**Jawab :**



```
-- no 4 tampilkan field mata kuliah dan pengampu mata kuliah.
-- |Tampilkan semua mata kuliah meskipun tidak ada pengampu nya
Run | New Tab | JSON | Copy
```

```
SELECT
  m.nama_mk,
  d.nama_dosen
FROM mk m
LEFT OUTER JOIN jadwal j ON m.kode_mk = j.kode_mk
LEFT OUTER JOIN dosen d ON j.kode_dosen = d.kode_dosen;
```

nama_mk varchar	nama_dosen varchar
Teknologi Data	Pramana Yoga Saputra, SKom., MMT.
Teknologi Data	Yoppy Yunhasnawa, SST., MSc.
Teknologi Data	Yoppy Yunhasnawa, SST., MSc.
Cyber Physical System	(NULL)
Komputasi Awan	(NULL)
Komputasi Kuantum	(NULL)

6. Tampilkan nama-nama dosen (88 baris) berikut jumlah jadwal mereka! Ketentuan:
- Nama kolom harus sesuai contoh
  - Apabila ada dosen yang tidak mendapatkan jadwal, jumlah\_jadwal\_mengajarnya haruslah = 0

```
-- no 5 tampilkan nama-nama dosen beserta jumlah jadwal mereka,  
-- jika dosen tidak mendapatkan jadwal, maka jumlah_jadwal_mengajarnya = 0  
Run | New Tab | JSON | Copy
```

```
SELECT  
    d.nama_dosen,  
    COUNT(j.kode_jadwal) AS jumlah_jadwal_mengajar  
FROM dosen d  
LEFT OUTER JOIN jadwal j ON d.kode_dosen = j.kode_dosen  
GROUP BY d.nama_dosen;
```

nama_dosen varchar	jumlah_jadwal_mengaja bigint
Shohib Muslim	6
Siti Romlah, Dra., M.M.	1
Sofyan Noor Arief, S.ST., M.I	6
Soryu Asuka Langley, M.Eng	0
Ulla Delfiana Rosiani, ST., M	5
Usman Nurhasan, S.Kom., M	4
Very Sugiarto, SPd., MKom.	2
Vipkas Al Hadid Firdaus, ST.	4
Vivi Nur Wijyaningrum, S.K	5
Vivin Ayu Lestari, SPd.	5
Widaningsih Condrowardha	7
Wilda Imama Sabilla, S.Kom	5