

LAPORAN PRAKTIKUM

MATA KULIAH PRAKTIKUM BASIS DATA

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

PERTEMUAN - 13



Nama : M. Zidna Billah Faza
NIM : 2341760030
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.

```
ThinkPad@FAZA C:\Users\ThinkPad\OneDrive\2. Kuliah\01. Praktikum Basis Data\Jobsheet 13
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 13
Server version: 10.4.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> source percobaan_join.sql;
Query OK, 0 rows affected, 1 warning (0.001 sec)

Query OK, 1 row affected (0.004 sec)

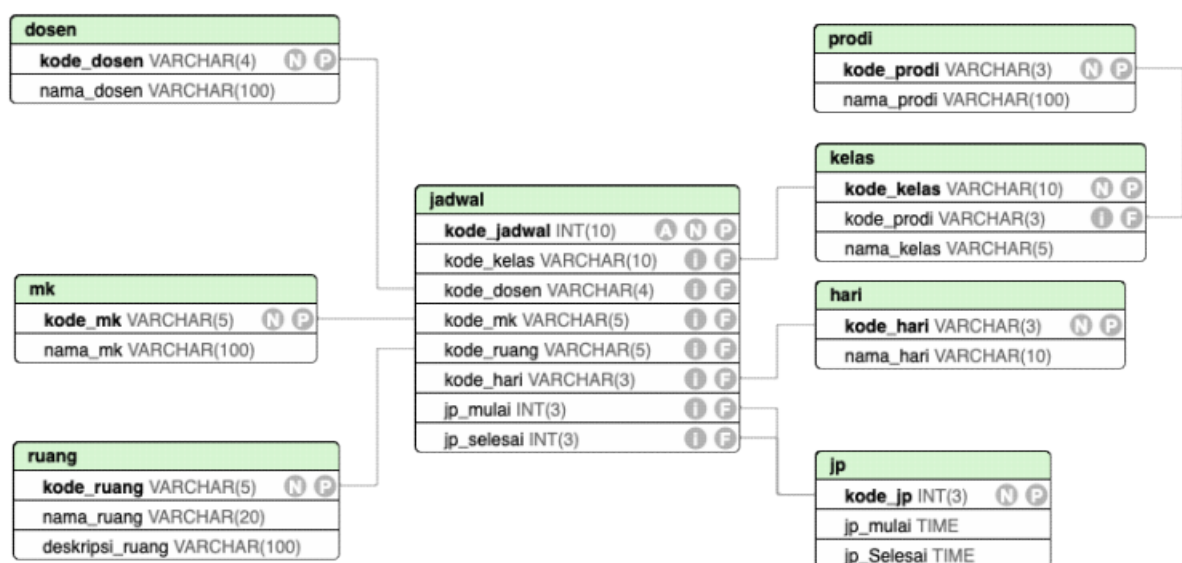
Database changed
Query OK, 0 rows affected (0.015 sec)

Query OK, 0 rows affected (0.010 sec)

Query OK, 0 rows affected (0.010 sec)

Query OK, 0 rows affected (0.009 sec)
```

- 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.



Praktikum 2

- 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).

```
Run | New Tab | JSON
SELECT kelas.nama_kelas, prodi.nama_prodi
FROM kelas
INNER JOIN prodi ON kelas.kode_prodi = prodi.kode_prodi; 10ms
```

> 48	TI-4C	D4 Teknik Informatika
> 49	TI-4D	D4 Teknik Informatika
> 50	TI-4E	D4 Teknik Informatika
> 51	TI-4F	D4 Teknik Informatika
> 52	TI-4G	D4 Teknik 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!

```
Run | New Tab | JSON
SELECT k.nama_kelas, p.nama_prodi
FROM kelas k
INNER JOIN prodi p ON p.kode_prodi = k.kode_prodi; 4ms
```

Q	nama_kelas varchar	nama_prodi varchar
> 42	TI-3E	D4 Teknik Informatika
> 43	TI-3F	D4 Teknik Informatika
> 44	TI-3G	D4 Teknik Informatika
> 45	TI-3H	D4 Teknik Informatika
> 46	TI-4A	D4 Teknik Informatika
> 47	TI-4B	D4 Teknik Informatika
> 48	TI-4C	D4 Teknik Informatika
> 49	TI-4D	D4 Teknik Informatika
> 50	TI-4E	D4 Teknik Informatika
> 51	TI-4F	D4 Teknik Informatika
> 52	TI-4G	D4 Teknik 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 320 baris. Pada query tersebut dilibatkan 4 tabel yaitu tabel jadwal, dosen, kelas, dan hari.

```
Run | New Tab | JSON | Copy
SELECT
  j.kode_jadwal,
  d.nama_dosen,
  k.nama_kelas,
  h.nama_hari
FROM jadwal j
  INNER JOIN dosen d ON d.kode_dosen = j.kode_dosen
  INNER JOIN kelas k ON k.kode_kelas = j.kode_kelas
  INNER JOIN hari h ON h.kode_hari = j.kode_hari; 3ms
```

	kode_jadwal int	nama_dosen varchar	nama_kelas varchar	nama_hari varchar	
> 90	2	Abdul Chalim, SAg., MPd.I	MI-1C	Selasa	
> 91	12	Agung Nugroho Pramudhit	TI-3A	Selasa	
> 92	13	Agung Nugroho Pramudhit	TI-3B	Selasa	
> 93	19	Ahmadi Yuli Ananta ST., MM	MI-2A	Selasa	

- 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”

```
Run | New Tab | JSON | Copy
SELECT k.nama_kelas, p.nama_prodi
FROM kelas k
      INNER JOIN prodi p ON p.kode_prodi = k.kode_prodi
WHERE k.nama_kelas LIKE '%A'; 7ms
```

	nama_kelas varchar	nama_prodi varchar
	Filter	Filter
> 1	MI-1A	D3 Manajemen Informatika
> 2	MI-2A	D3 Manajemen Informatika
> 3	MI-3A	D3 Manajemen Informatika
> 4	TI-1A	D4 Teknik Informatika
> 5	TI-2A	D4 Teknik Informatika
> 6	TI-3A	D4 Teknik Informatika
> 7	TI-4A	D4 Teknik Informatika

- 5) Lanjutkan ke Praktikum – Bagian 3.

Praktikum 3

- 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.

```
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; 4ms
```

Q	nama_kelas varchar	nama_prodi varchar
> 47	TI-4B	D4 Teknik Informatika
> 48	TI-4C	D4 Teknik Informatika
> 49	TI-4D	D4 Teknik Informatika
> 50	TI-4E	D4 Teknik Informatika
> 51	TI-4F	D4 Teknik Informatika
> 52	TI-4G	D4 Teknik Informatika
> 53	TRM-1A	(NULL)
> 54	TRM-1B	(NULL)
> 55	TRM-1C	(NULL)
> 56	S2TI-A	(NULL)
> 57	S2TI-B	(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.

```
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; 4ms
```

Q	nama_kelas varchar	nama_prodi varchar
	Filter	Filter
> 1	TRM-1A	(NULL)
> 2	TRM-1B	(NULL)
> 3	TRM-1C	(NULL)
> 4	S2TI-A	(NULL)
> 5	S2TI-B	(NULL)

- 3) 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.

```

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; 3ms

```

	nama_kelas varchar	nama_prodi varchar
> 45	TI-3H	D4 Teknik Informatika
> 46	TI-4A	D4 Teknik Informatika
> 47	TI-4B	D4 Teknik Informatika
> 48	TI-4C	D4 Teknik Informatika
> 49	TI-4D	D4 Teknik Informatika
> 50	TI-4E	D4 Teknik Informatika
> 51	TI-4F	D4 Teknik Informatika
> 52	TI-4G	D4 Teknik Informatika
> 53	(NULL)	D4 Sistem Integritas Tinggi
> 54	(NULL)	D4 Kecerdasan Buatan dan Robotika
> 55	(NULL)	D4 Sistem Informasi Bisnis

- 4) Jika kita 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 mengakalinya 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

```
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; 22ms
```

Q	nama_kelas varchar	nama_prodi varchar
> 48	TI-4C	D4 Teknik Informatika
> 49	TI-4D	D4 Teknik Informatika
> 50	TI-4E	D4 Teknik Informatika
> 51	TI-4F	D4 Teknik Informatika
> 52	TI-4G	D4 Teknik Informatika
> 53	TRM-1A	(NULL)
> 54	TRM-1B	(NULL)
> 55	TRM-1C	(NULL)
> 56	S2TI-A	(NULL)
> 57	S2TI-B	(NULL)
> 58	MI-1A	D3 Manajemen Informatika
> 59	MI-1B	D3 Manajemen Informatika

> 9	TI-4G	D4 Teknik Informatika
> 10	(NULL)	D4 Sistem Integritas Tinggi
> 11	(NULL)	D4 Kecerdasan Buatan dan
> 12	(NULL)	D4 Sistem Informasi Bisnis

5) Lanjutkan ke Praktikum – Bagian 5.

Praktikum 4

- 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.

```
Run | New Tab | JSON | Copy
SELECT
    kelas.nama_kelas,
    prodi.nama_prodi
FROM
    kelas, prodi
WHERE
    kelas.kode_prodi = prodi.kode_prodi; 1ms
```

	nama_kelas varchar	nama_prodi varchar
	Filter	Filter
> 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
> 9	MI-2B	D3 Manajemen Informatika
> 10	MI-2C	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.

```
Run | New Tab | JSON | Copy
SELECT
  k.nama_kelas,
  p.nama_prodi
FROM
  kelas k, prodi p
WHERE
  k.kode_prodi = p.kode_prodi; 3ms
```

	nama_kelas varchar	nama_prodi varchar
	Filter	Filter
> 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
> 9	MI-2B	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 mempedulikan kesesuaian antar tabel atau tidak. Jumlah baris yang dihasilkan adalah perkalian jumlah baris pada kedua buah tabel yang di CROSS-kan.

```
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
	Filter	Filter
> 1	MI-1A	D3 Manajemen Informatika
> 2	MI-1A	D4 Teknik Informatika
> 3	MI-1A	D4 Sistem Integritas Tinggi
> 4	MI-1A	D4 Kecerdasan Buatan dan
> 5	MI-1A	D4 Sistem Informasi Bisnis
> 6	MI-1B	D3 Manajemen Informatika
> 7	MI-1B	D4 Teknik Informatika
> 8	MI-1B	D4 Sistem Integritas Tinggi
> 9	MI-1B	D4 Kecerdasan Buatan dan
> 10	MI-1B	D4 Sistem Informasi Bisnis

- 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.

```

Run | New Tab | JSON | Copy
SELECT
  k.nama_kelas,
  p.nama_prodi
FROM
  kelas k,
  prodi p; 4ms

```

	nama_kelas: varchar	nama_prodi: varchar
	Filter	Filter
> 1	MI-1A	D3 Manajemen Informatika
> 2	MI-1A	D4 Teknik Informatika
> 3	MI-1A	D4 Sistem Integritas Tinggi
> 4	MI-1A	D4 Kecerdasan Buatan dan
> 5	MI-1A	D4 Sistem Informasi Bisnis
> 6	MI-1B	D3 Manajemen Informatika
> 7	MI-1B	D4 Teknik Informatika

TUGAS

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

```
Run | New Tab | JSON | Copy
SELECT DISTINCT
  d.nama_dosen AS 'Dosen Pengampu',
  mk.nama_mk AS 'Nama Mata Kuliah'
FROM
  dosen d
  INNER JOIN jadwal j ON d.kode_dosen = j.kode_dosen
  INNER JOIN mk ON j.kode_mk = mk.kode_mk
ORDER BY
  d.nama_dosen ASC; 6ms
```

	Dosen Pengampu varchar	Nama Mata Kuliah varchar
> 76	Widaningsih Condrowardhani, SH., MH.	Etika Profesi Bidang TI
> 77	Wilda Imama Sabilla, S.Kom., M.Kom.	Praktikum Basis Data
> 78	Wilda Imama Sabilla, S.Kom., M.Kom.	Basis Data
> 79	Wilda Imama Sabilla, S.Kom., M.Kom.	E-Business
> 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.

```

> Run | New Tab | JSON | Copy
SELECT
  r.kode_ruang AS 'Kode Ruang',
  r.nama_ruang AS 'Nama Ruang',
  'Selasa' AS 'Hari Dipakai',
  jp_mulai.jp_mulai AS 'Dari Jam',
  jp_selesai.jp_selesai AS 'Sampai Jam'
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
  INNER JOIN jp jp_mulai ON j.jp_mulai = jp_mulai.kode_jp
  INNER JOIN jp jp_selesai ON j.jp_selesai = jp_selesai.kode_jp
WHERE
  h.nama_hari = 'Selasa'; 3ms

```

Q	Kode Ruan varchar	Nama Ruai varchar	Hari Dipakai varchar	Dari Jam time	Sampai Jar time
> 69	0705	LPR5	Selasa	13:40:00	18:00:00
> 70	0506	RT06	Selasa	07:00:00	10:30:00
> 71	0503	RT03	Selasa	07:00:00	09:30:00
> 72	0503	RT03	Selasa	07:00:00	09:30:00
> 73	0806	RT10	Selasa	07:00:00	09:30:00
> 74	0707	LKJ1	Selasa	09:40:00	12:10:00
> 75	0707	LKJ1	Selasa	09:40:00	12:10:00
> 76	0720	LAI1	Selasa	12:50:00	18:00:00
> 77	0508	LPY1	Selasa	12:50:00	18:00:00
> 78	0504	RT04	Selasa	12:50:00	15:20:00
> 79	0806	RT10	Selasa	15:30:00	18:00:00

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

```
Run | New Tab | JSON | Copy
SELECT
  d.nama_dosen AS 'Nama Dosen',
  j.kode_jadwal AS 'Kode Jadwal'
FROM
  dosen d
  LEFT JOIN jadwal j ON d.kode_dosen = j.kode_dosen
WHERE
  j.kode_jadwal IS NULL; 2ms
```

	Nama Dosen varchar	Kode Jadwal int
	Filter	Filter
> 1	Kamado Tanjiro, S.Kom., M.Kom.	(NULL)
> 2	Rei Ayanami, S.ST., M.Sc.	(NULL)
> 3	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

```
Run | New Tab | JSON | Copy
SELECT DISTINCT
  mk.nama_mk AS 'Mata Kuliah',
  d.nama_dosen AS 'Pengampu'
FROM
  dosen d
RIGHT JOIN
  jadwal j ON d.kode_dosen = j.kode_dosen
RIGHT JOIN
  mk ON j.kode_mk = mk.kode_mk
WHERE
  j.kode_jadwal IS NULL OR j.kode_jadwal IS NOT NULL; 16ms
```

	Mata Kuliah varchar	Pengampu varchar
> 82	Teknologi Data	Dodit Supriyanto SKom., M
> 83	Teknologi Data	Gunawan Budi Prasetyo, ST.
> 84	Teknologi Data	Odhitya Desta Triswidranan
> 85	Teknologi Data	Pramana Yoga Saputra, SKo
> 86	Teknologi Data	Yoppy Yunhasnawa, SST., M
> 87	Cyber Physical System	(NULL)
> 88	Komputasi Awan	(NULL)
> 89	Komputasi Kuantum	(NULL)
> 90	Swarm Robotics	(NULL)
> 91	Collaborative Thought	(NULL)
> 92	Matematika Transendental	(NULL)

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

```
Run | New Tab | JSON | Copy
SELECT
  d.nama_dosen AS 'Nama Dosen',
  COUNT(j.kode_jadwal) AS 'Jumlah_Jadwal_Mengajar'
FROM
  dosen d
LEFT JOIN
  jadwal j ON d.kode_dosen = j.kode_dosen
GROUP BY
  d.nama_dosen; 4ms
```

Q	Nama Dosen varchar	Jumlah_Jadwal_Mengajar bigint
> 78	Ulla Delfiana Rosiani, ST., MT.	5
> 79	Usman Nurhasan, S.Kom., MT.	4
> 80	Very Sugianto, SPd., MKom.	2
> 81	Vipkas Al Hadid Firdaus, ST.,MT.	4
> 82	Vivi Nur Wijayaningrum, S.Kom, M.Kom	5
> 83	Vivin Ayu Lestari, SPd.	5
> 84	Widaningsih Condrowardhani, SH., MH.	7
> 85	Wilda Imama Sabilla, S.Kom., M.Kom.	5
> 86	Yoppy Yunhasnawa, SST., MSc.	4
> 87	Yuri Ariyanto, SKom., MKom.	5
> 88	Zulmy Faqihuddin Putera, S.Pd., M.Pd	6

Github :
[zidnafaz/Praktikum-Basis-Data \(github.com\)](https://github.com/zidnafaz/Praktikum-Basis-Data)

Google Drive :
https://drive.google.com/drive/folders/1Ex7eiTKLfSRM44yVtCAgqTeDfL4_75Wt?usp=sharing