



Topik

SELECT pada lebih dari satu tabel

Tujuan

Mahasiswa diharapkan memahami dan mampu menggunakan statement:

1. SELECT - INNER JOIN.
2. SELECT - LEFT OUTER JOIN.
3. SELECT - RIGHT OUTER JOIN.
4. SELECT - CROSS JOIN
5. Implicit JOIN

Pendahuluan

Relationship adalah suatu hubungan antara beberapa entitas. Konsep ini sangat penting sekali di dalam basis data, di mana memungkinkan entitas-entitas untuk saling berhubungan satu sama lain. Di dalam sebuah relationship, primary key memiliki peran penting untuk mengaitkan entitas. Selain itu, primary key juga digunakan untuk mendefinisikan batasan keterhubungan.

JOIN merupakan salah satu konstruksi dasar dari SQL dan basis data. Join dapat didefinisikan sebagai kombinasi record dari dua atau lebih tabel di dalam basis data relasional dan menghasilkan sebuah tabel (temporary) baru yang disebut sebagai joined table. Join dapat diklasifikasikan ke dalam dua jenis: INNER dan OUTER.

1. INNER JOIN

Inner join pada dasarnya adalah menemukan persimpangan (intersection) antara dua buah tabel. Sintaks *inner join* diperlihatkan sebagai berikut:

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

Cara 1

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

Inner join juga dapat direpresentasikan dalam bentuk implisit.

```
SELECT A1, A2, ..., An
FROM r1, r2
WHERE r1.key = r2.key
```

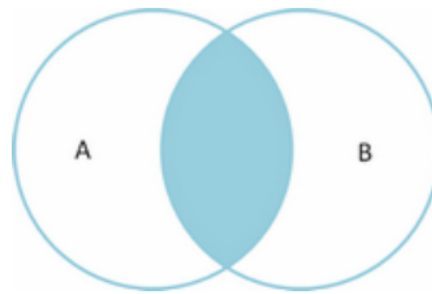
Cara 2

```
SELECT A1, A2, ..., An
```

FROM r1, r2

WHERE r1.key = r2.key

Misalkan terdapat tabel A dan B, maka hasil *inner join* dapat diperlihatkan—sebagai bidang terarsir—dalam diagram Venn seperti Gambar 1.



Gambar 1. Representasi INNER JOIN

2. OUTER JOIN

LEFT OUTER JOIN

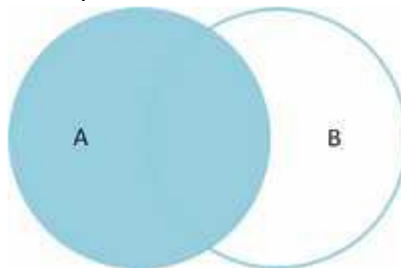
Left outer join (atau left join) mengembalikan semua nilai dari tabel kiri ditambah dengan nilai dari tabel kanan yang sesuai (atau NULL jika tidak ada nilai yang sesuai).

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

Syntax

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

Left outer join antara tabel A dan B dapat diilustrasikan dalam diagram Venn seperti Gambar 2.



Gambar 2. Representasi LEFT OUTER JOIN

RIGHT OUTER JOIN

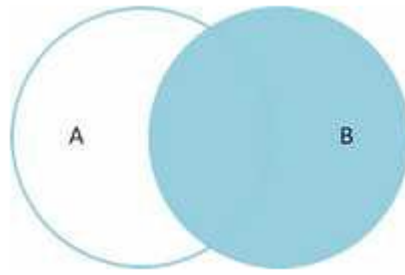
Right outer join (atau right join) pada dasarnya sama seperti left join, namun dalam bentuk terbalik—kanan dan kiri. Sintaks *right outer join* diperlihatkan sebagai berikut:

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

Syntax

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

Right outer join antara tabel A dan B dapat diilustrasikan dalam diagram Venn seperti Gambar 3.



Gambar 3. Representasi RIGHT OUTER JOIN

FULL OUTER JOIN

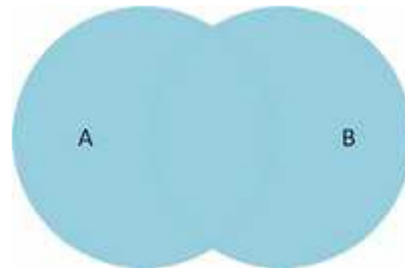
Full outer join (atau full join) pada hakekatnya merupakan kombinasi dari left dan right join. Sintaks *full outer join* diperlihatkan sebagai berikut:

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

Syntax

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

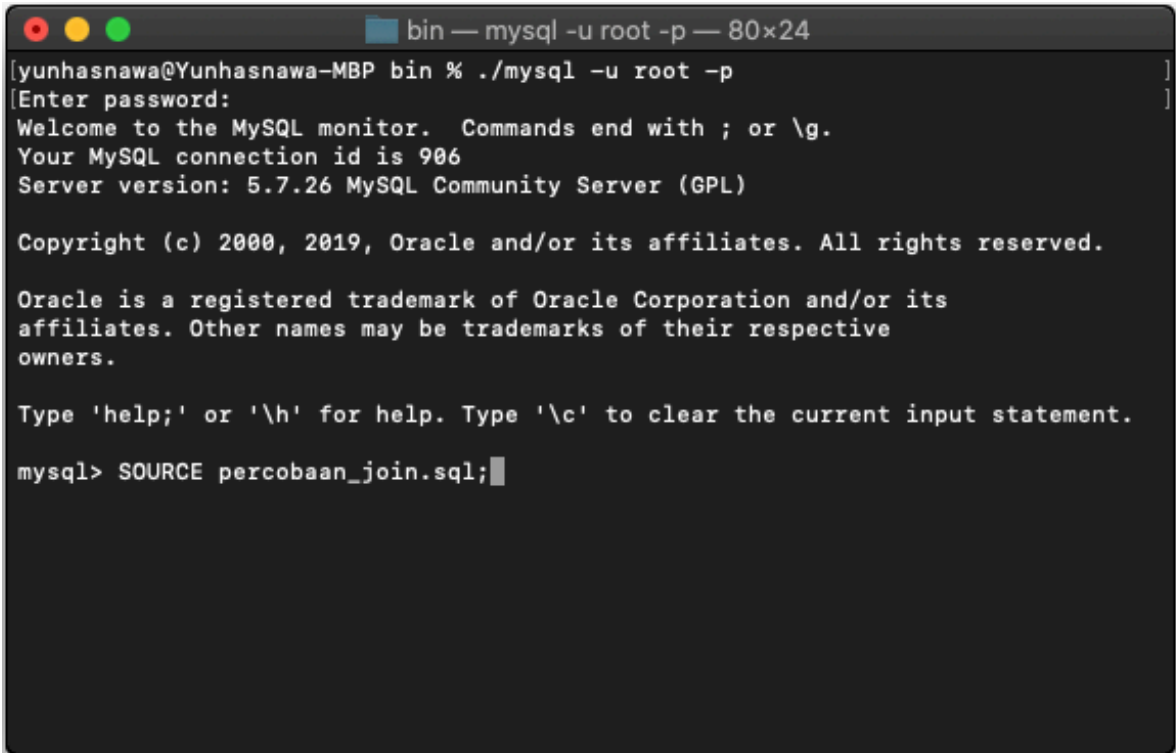
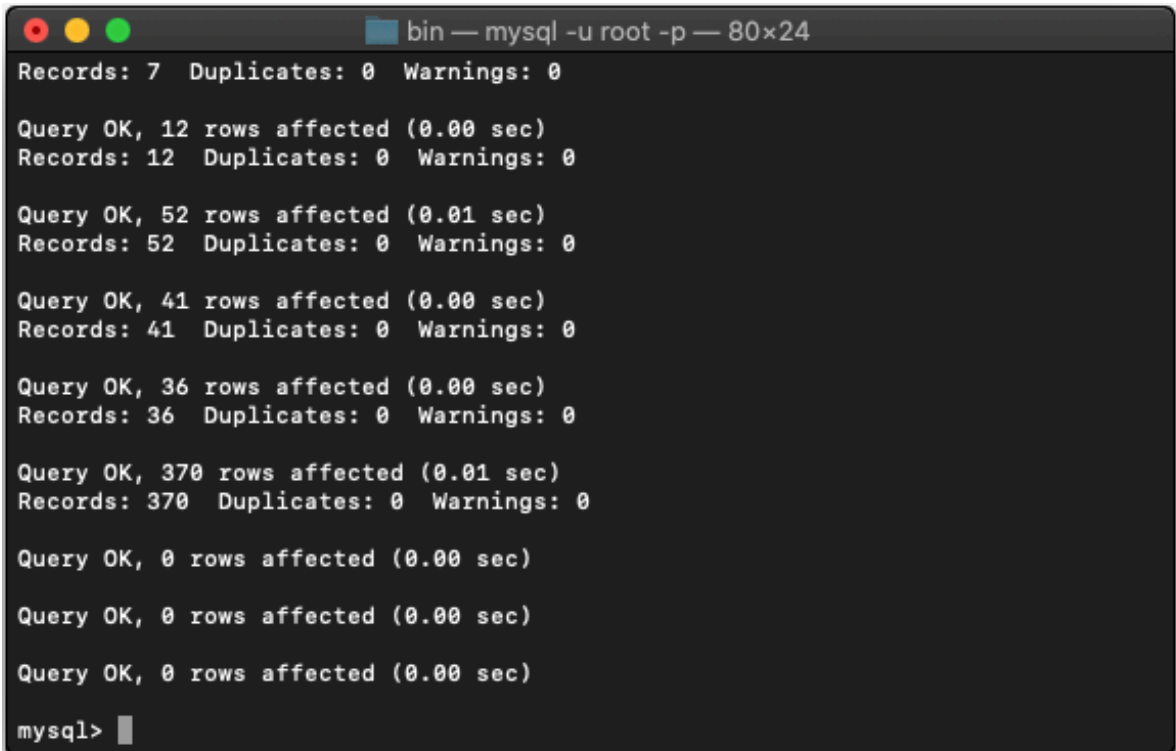
Bentuk visual dari *full outer join* dapat diperlihatkan menggunakan diagram Venn seperti Gambar 4.



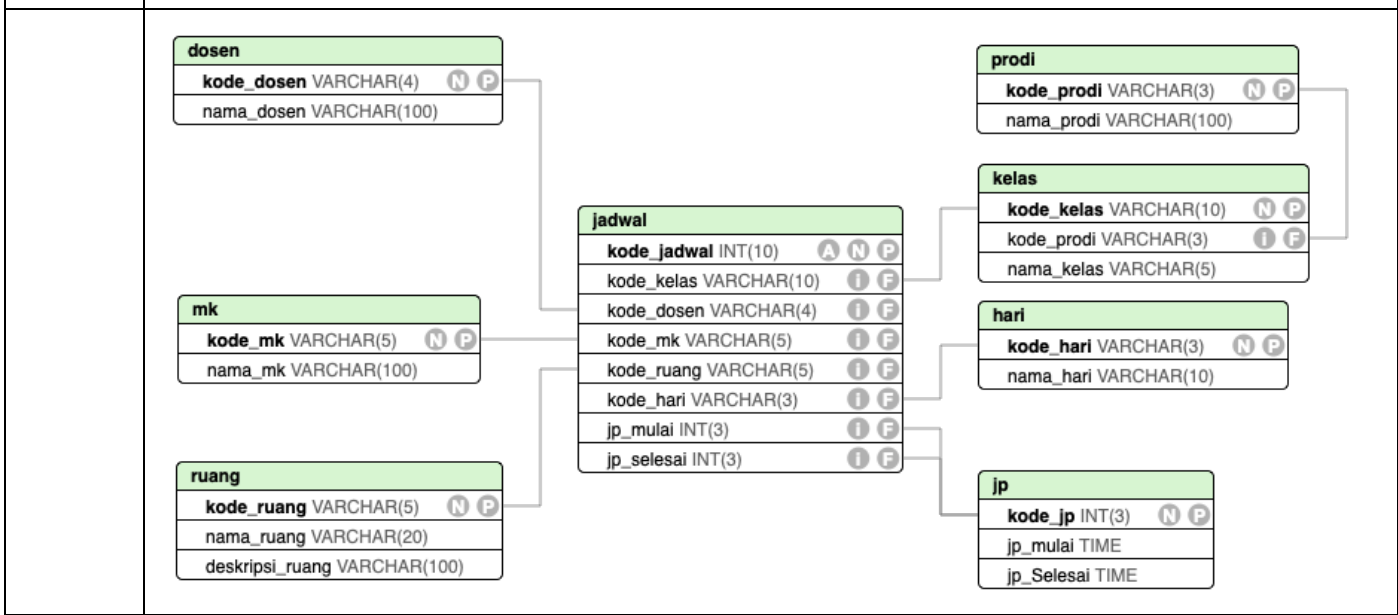
Gambar 4. Representasi FULL OUTER JOIN

Selain empat jenis join yang utama di atas, masih ada beberapa variasi join lainnya, seperti CROSS JOIN (cartesian product), NATURAL JOIN, dan sebagainya. Perlu juga diperhatikan, join bisa diimplementasikan dalam bentuk bersarang (nested join). Jadi, di dalam sebuah operasi join bisa terdapat operasi join lainnya.

Praktikum – Bagian 1: Menyiapkan Database untuk Percobaan

Langkah	Keterangan
1	<p>Bersama jobsheet ini disertakan sebuah file bernama percobaan_join.sql, eksekusilah file tersebut pada server MySQL Anda dengan cara apapun yang Anda bisa.</p> <p>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.</p>
	 <pre>bin — mysql -u root -p — 80x24 [yunhasnawa@Yunhasnawa-MBP bin % ./mysql -u root -p] [Enter password:] Welcome to the MySQL monitor. Commands end with ; or \g. Your MySQL connection id is 906 Server version: 5.7.26 MySQL Community Server (GPL) Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement. mysql> SOURCE percobaan_join.sql;</pre>  <pre>bin — mysql -u root -p — 80x24 Records: 7 Duplicates: 0 Warnings: 0 Query OK, 12 rows affected (0.00 sec) Records: 12 Duplicates: 0 Warnings: 0 Query OK, 52 rows affected (0.01 sec) Records: 52 Duplicates: 0 Warnings: 0 Query OK, 41 rows affected (0.00 sec) Records: 41 Duplicates: 0 Warnings: 0 Query OK, 36 rows affected (0.00 sec) Records: 36 Duplicates: 0 Warnings: 0 Query OK, 370 rows affected (0.01 sec) Records: 370 Duplicates: 0 Warnings: 0 Query OK, 0 rows affected (0.00 sec) Query OK, 0 rows affected (0.00 sec) Query OK, 0 rows affected (0.00 sec) mysql></pre>

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.



3 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**.

1 ✓
2
3 ✓

```
USE percobaan_join;
-- Cek tabel
SHOW TABLES;
```

OutputCek tabel

8 rows

Tables_in_percobaan_join

1 dosen

2 hari

3 jadwal

4 jp

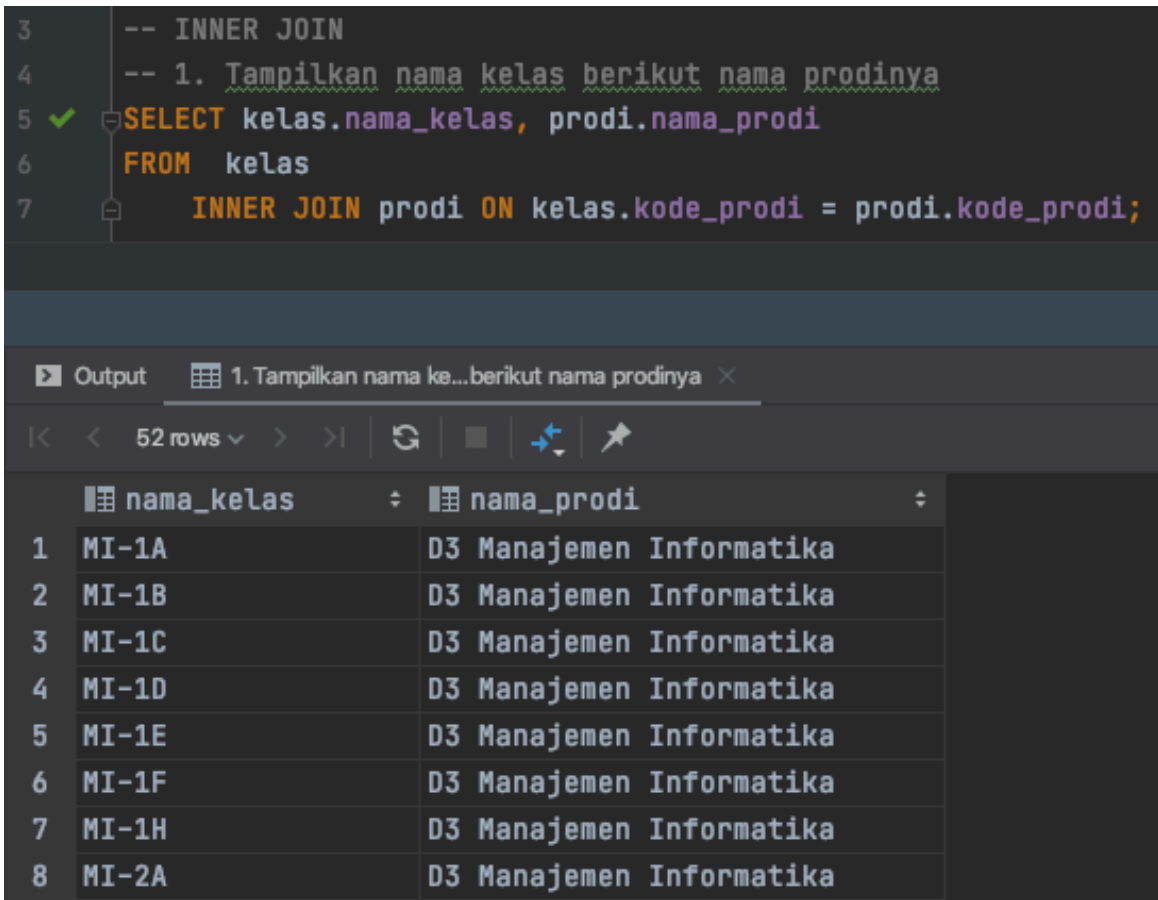
5 kelas

6 mk

7 prodi

8 ruang

Praktikum – Bagian 2: INNER JOIN

Langkah	Keterangan																											
1	<p>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.</p> <p>Contoh: “Tampilkan nama kelas berikut nama prodinya!”</p> <p>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).</p> <pre>3 -- INNER JOIN 4 -- 1. Tampilkan nama kelas berikut nama prodinya 5 ✓ SELECT kelas.nama_kelas, prodi.nama_prodi 6 FROM kelas 7 INNER JOIN prodi ON kelas.kode_prodi = prodi.kode_prodi;</pre>  <table><thead><tr><th></th><th>nama_kelas</th><th>nama_prodi</th></tr></thead><tbody><tr><td>1</td><td>MI-1A</td><td>D3 Manajemen Informatika</td></tr><tr><td>2</td><td>MI-1B</td><td>D3 Manajemen Informatika</td></tr><tr><td>3</td><td>MI-1C</td><td>D3 Manajemen Informatika</td></tr><tr><td>4</td><td>MI-1D</td><td>D3 Manajemen Informatika</td></tr><tr><td>5</td><td>MI-1E</td><td>D3 Manajemen Informatika</td></tr><tr><td>6</td><td>MI-1F</td><td>D3 Manajemen Informatika</td></tr><tr><td>7</td><td>MI-1H</td><td>D3 Manajemen Informatika</td></tr><tr><td>8</td><td>MI-2A</td><td>D3 Manajemen Informatika</td></tr></tbody></table>		nama_kelas	nama_prodi	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
	nama_kelas	nama_prodi																										
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																										
2	<p>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.</p> <p>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.</p> <p>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!</p>																											

```

9      -- 2. Menggunakan alias (AS)
10     ✓ SELECT k.nama_kelas, p.nama_prodi
11         FROM kelas k
12         INNER JOIN prodi p ON p.kode_prodi = k.kode_prodi;

```

Output Result 3

52 rows

	nama_kelas	nama_prodi
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

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 melibatkan 4 tabel yaitu tabel jadwal, dosen, kelas, dan hari.

```

14     -- 3. JOIN beberapa tabel
15     -- Tampilkan dosen mengajar di kelas apa pada hari apa
16     ✓ SELECT
17         j.kode_jadwal,
18         d.nama_dosen,
19         k.nama_kelas,
20         h.nama_hari
21     FROM
22         jadwal j
23         INNER JOIN dosen d ON d.kode_dosen = j.kode_dosen
24         INNER JOIN kelas k ON k.kode_kelas = j.kode_kelas
25         INNER JOIN hari h ON h.kode_hari = j.kode_hari;

```

Output Result 4

370 rows

	kode_jadwal	nama_dosen	nama_kelas	nama_hari
1	6	Abdul Chalim, SAg., MPd.I	MI-1A	Kamis
2	37	Annisa Taufika Firdausi ST., MT.	MI-1A	Jumat
3	165	Hairus	MI-1A	Jumat
4	170	Ika Kusumaning Putri, MT.	MI-1A	Senin
5	174	Ika Kusumaning Putri, MT.	MI-1A	Kamis
6	178	Indra Dharma Wijaya, ST., MMT.	MI-1A	Senin
7	180	Indra Dharma Wijaya, ST., MMT.	MI-1A	Selasa
8	276	Ridwan Rismanto, SST., MKom.	MI-1A	Rabu

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”

```
30 -- 4. INNER JOIN dengan WHERE
31 --     Menampilkan nama kelas dan nama prodi,
32 --     yang nama kelasnya adalah kelas 'A'
33 ✓ SELECT k.nama_kelas, p.nama_prodi
34 FROM kelas k
35     INNER JOIN prodi p ON p.kode_prodi = k.kode_prodi
36 WHERE
37     k.nama_kelas LIKE '%A';
```

Output Result 31

7 rows

	nama_kelas	nama_prodi
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

4

5

Lanjutkan ke **Praktikum – Bagian 3**.

Praktikum - Bagian 3: OUTER JOIN

Langkah	Keterangan																														
1	<p>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.</p> <p>OUTER JOIN dibagi menjadi 2:</p> <ul style="list-style-type: none">- LEFT OUTER JOIN dan;- RIGHT OUTER JOIN <p>OUTER JOIN pada umumnya berguna untuk mengecek data yang tidak ada pasangannya di tabel yang di-JOIN-kan.</p> <p>Contoh: “Tampilkan data semua kelas berikut nama prodinya, beserta kelas yang tidak ada prodinya!”</p> <p>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.</p> <pre>32 -- 1. LEFT OUTER JOIN 33 -- Menampilkan kelas yang belum ada prodinya 34 ✓ SELECT 35 k.nama_kelas, 36 p.nama_prodi 37 FROM 38 kelas k 39 LEFT OUTER JOIN prodi p ON k.kode_prodi = p.kode_prodi;</pre> <p>Output: Menampilkan kelas ya...ng belum ada prodinya</p> <p>57 rows</p> <table><thead><tr><th></th><th>nama_kelas</th><th>nama_prodi</th></tr></thead><tbody><tr><td>49</td><td>TI-4D</td><td>D4 Teknik Informatika</td></tr><tr><td>50</td><td>TI-4E</td><td>D4 Teknik Informatika</td></tr><tr><td>51</td><td>TI-4F</td><td>D4 Teknik Informatika</td></tr><tr><td>52</td><td>TI-4G</td><td>D4 Teknik Informatika</td></tr><tr><td>53</td><td>TRM-1A</td><td><null></td></tr><tr><td>54</td><td>TRM-1B</td><td><null></td></tr><tr><td>55</td><td>TRM-1C</td><td><null></td></tr><tr><td>56</td><td>S2TI-A</td><td><null></td></tr><tr><td>57</td><td>S2TI-B</td><td><null></td></tr></tbody></table>		nama_kelas	nama_prodi	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>
	nama_kelas	nama_prodi																													
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>																													

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.

```
41 -- 2. OUTER JOIN dengan WHERE
42 -- Menampilkan kelas yang belum ada prodinya saja
43 ✓ SELECT
44     k.nama_kelas,
45     p.nama_prodi
46 FROM
47     kelas k
48     LEFT OUTER JOIN prodi p ON k.kode_prodi = p.kode_prodi
49 WHERE
50     p.kode_prodi IS NULL;
```

Output Result 15 x

5 rows

	nama_kelas	nama_prodi
1	TRM-1A	<null>
2	TRM-1B	<null>
3	TRM-1C	<null>
4	S2TI-A	<null>
5	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.

```

52 -- 3. RIGHT OUTER JOIN
53 -- Sama dengan LEFT OUTER JOIN, hanya beda tempat NULL-nya saja.
54 -- LEFT OUTER JOIN --> Yang penuh di KIRI (LEFT)
55 -- RIGHT OUTER JOIN --> Yang penuh di KANAN (RIGHT)
56 ✓ SELECT
57     k.nama_kelas,
58     p.nama_prodi
59 FROM
60     kelas k
61     RIGHT OUTER JOIN prodi p ON p.kode_prodi = k.kode_prodi;

```

Output Result 16 x

55 rows

	nama_kelas	nama_prodi
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

```

63 -- 4. FULL JOIN
64 -- Gabungan LEFT OUTER JOIN dan RIGHT OUTER JOIN
65 ✓ SELECT k.nama_kelas, p.nama_prodi
66 FROM kelas k
67 LEFT OUTER JOIN prodi p ON k.kode_prodi = p.kode_prodi
68 UNION ALL
69 SELECT k.nama_kelas, p.nama_prodi
70 FROM kelas k
71 RIGHT OUTER JOIN prodi p ON p.kode_prodi = k.kode_prodi;

```

Output

Result 18

112 rows

	nama_kelas	nama_prodi
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

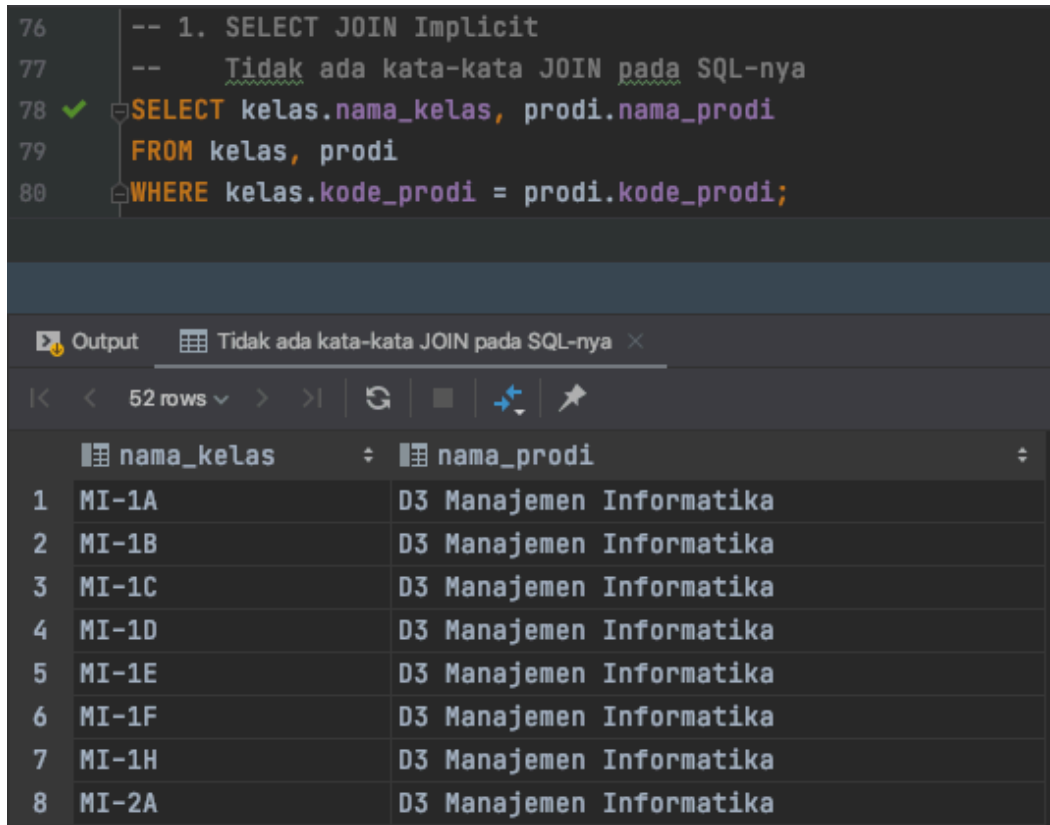
...

	nama_kelas	nama_prodi
108	TI-4F	D4 Teknik Informatika
109	TI-4G	D4 Teknik Informatika
110	<null>	D4 Sistem Integritas Tinggi
111	<null>	D4 Kecerdasan Buatan dan Rob...
112	<null>	D4 Sistem Informasi Bisnis

5

Lanjutkan ke **Praktikum – Bagian 5**.

Praktikum - Bagian 4: INNER JOIN Implisit dan CROSS JOIN

Langkah	Keterangan																											
1	<p>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.</p> <p>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.</p> <p>Namun demikian untuk sekedar pengetahuan Anda, jalankan sintaksis SQL berikut, hasilnya akan sama dengan hasil pada Praktikum – Bagian 1 Langkah 1.</p> <p>Perhatikan pada sintaksis ini tidak ada kata-kata “JOIN”-nya.</p>  <pre>76 -- 1. SELECT JOIN Implicit 77 -- Tidak ada kata-kata JOIN pada SQL-nya 78 ✓ SELECT kelas.nama_kelas, prodi.nama_prodi 79 FROM kelas, prodi 80 WHERE kelas.kode_prodi = prodi.kode_prodi;</pre> <p>Output: Tidak ada kata-kata JOIN pada SQL-nya</p> <p>52 rows</p> <table><thead><tr><th></th><th>nama_kelas</th><th>nama_prodi</th></tr></thead><tbody><tr><td>1</td><td>MI-1A</td><td>D3 Manajemen Informatika</td></tr><tr><td>2</td><td>MI-1B</td><td>D3 Manajemen Informatika</td></tr><tr><td>3</td><td>MI-1C</td><td>D3 Manajemen Informatika</td></tr><tr><td>4</td><td>MI-1D</td><td>D3 Manajemen Informatika</td></tr><tr><td>5</td><td>MI-1E</td><td>D3 Manajemen Informatika</td></tr><tr><td>6</td><td>MI-1F</td><td>D3 Manajemen Informatika</td></tr><tr><td>7</td><td>MI-1H</td><td>D3 Manajemen Informatika</td></tr><tr><td>8</td><td>MI-2A</td><td>D3 Manajemen Informatika</td></tr></tbody></table>		nama_kelas	nama_prodi	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
	nama_kelas	nama_prodi																										
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																										
2	<p>INNER JOIN Implisit juga bisa diberikan alias nama tabel agar tidak terlalu Panjang.</p> <p>Alias nama tabel dituliskan pada klausa FROM, sama seperti pada Explicit JOIN.</p> <p>Jalankan query berikut. Hasilnya akan sama dengan langkah sebelumnya, namun dengan penulisan SQL yang lebih singkat.</p>																											

82	-- 2. INNER JOIN implicit dengan alias (AS)
83	✓ <code>SELECT k.nama_kelas, p.nama_prodi</code>
84	<code>FROM kelas k, prodi p</code>
85	<code>WHERE k.kode_prodi = p.kode_prodi;</code>
86	

Output	2. INNER JOIN implicit dengan alias (AS) X
52 rows	
nama_kelas	nama_prodi
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

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 mempertimbangkan kesesuaian antar tabel atau tidak.

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

87	-- 3. CROSS JOIN Explicit
88	✓ <code>SELECT k.nama_kelas, p.nama_prodi</code>
89	<code>FROM kelas k CROSS JOIN prodi p;</code>

Output	3. CROSS JOIN Explicit X
285 rows	
nama_kelas	nama_prodi
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 Robotika
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 Robotika

4	<p>CROSS JOIN juga dapat dilakukan secara IMPLICIT yaitu dengan tanpa memberikan WHERE pada sintaksis JOIN IMPLICIT.</p> <p>Jalankan query di bawah ini, hasilnya akan sama dengan langkah sebelumnya.</p>
	<pre>91 -- 4. CROSS JOIN Implicit 92 SELECT k.nama_kelas, p.nama_prodi 93 FROM kelas k, prodi p;</pre>

Tugas

- Jalankan semua SQL pada praktikum-praktikum di atas, pahami maksudnya dan *Screenshot*-lah hasilnya!
- Tampilkan nama dosen berikut mata kuliah yang mereka ampu (186 baris) dengan ketentuan:
 - Tidak ada data yang duplikat
 - Urut berdasarkan nama dosen dari A-Z.
 - Nama kolom yang ditampilkan harus sesuai dengan contoh di bawah.

	📄 `Dosen Pengampu`	÷ 📄 `Nama Mata Kuliah`
176	Widaningsih Condrowardhani, SH., MH.	Etika Profesi Bidang TI
177	Wilda Imama Sabilla, S.Kom., M.Kom.	E-Business
178	Wilda Imama Sabilla, S.Kom., M.Kom.	Praktikum Basis Data
179	Wilda Imama Sabilla, S.Kom., M.Kom.	Basis Data
180	Yoppy Yunhasnawa, SST., MSc.	Teknologi Data
181	Yoppy Yunhasnawa, SST., MSc.	Praktikum Basis Data
182	Yoppy Yunhasnawa, SST., MSc.	Basis Data
183	Yuri Ariyanto, SKom., MKom.	Manajemen Proyek
184	Yuri Ariyanto, SKom., MKom.	Proyek 1_P1
185	Yuri Ariyanto, SKom., MKom.	Proyek 1_P2
186	Zulmy Faqihuddin Putera, S.Pd., M.Pd	Bahasa Indonesia

- 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.

	📄 `Kode Ruang`	÷ 📄 `Nama Ruang`	÷ 📄 `Hari Dipakai`	÷ 📄 `Dari Jam`	÷ 📄 `Sampai Jam`
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

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

	📄 `Nama Dosen`	÷ 📄 `Kode Jadwal`
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

Mata Kuliah	Pengampu
Struktur Data	Vivian Ayu Lestari, Sru.
Teknologi Data	Dodit Supriyanto SKom., MT.
Teknologi Data	Gunawan Budi Prasetyo, ST., MMT., Ph.D.
Teknologi Data	Odhitia Desta Triswidrananta, SPd., MPd.
Teknologi Data	Pramana Yoga Saputra, SKom., MMT.
Teknologi Data	Yoppy Yunhasnawa, SST., MSc.
Cyber Physical System	<null>
Komputasi Awan	<null>
Komputasi Kuantum	<null>
Swarm Robotics	<null>
Collaborative Thought	<null>
Matematika Transendental	<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

nama_dosen	jumlah_jadwal_mengajar
Rosa Andrie Asmara, ST., MT., Dr. Eng.	2
Deasy Sandhya Elya Ikawati SSi., MSi.	2
Siti Romlah, Dra., M.M.	1
Farid Angga Pribadi, SKom., MKom.	1
Robby Anggriawan SE., ME.	1
Kamado Tanjiro, S.Kom., M.Kom.	0
Rei Ayanami, S.ST., M.Sc.	0
Soryu Asuka Langley, M.Eng, Ph.D.	0

-- Selamat Mengerjakan --