LAPORAN BASIS DATA
PERTEMUAN 12

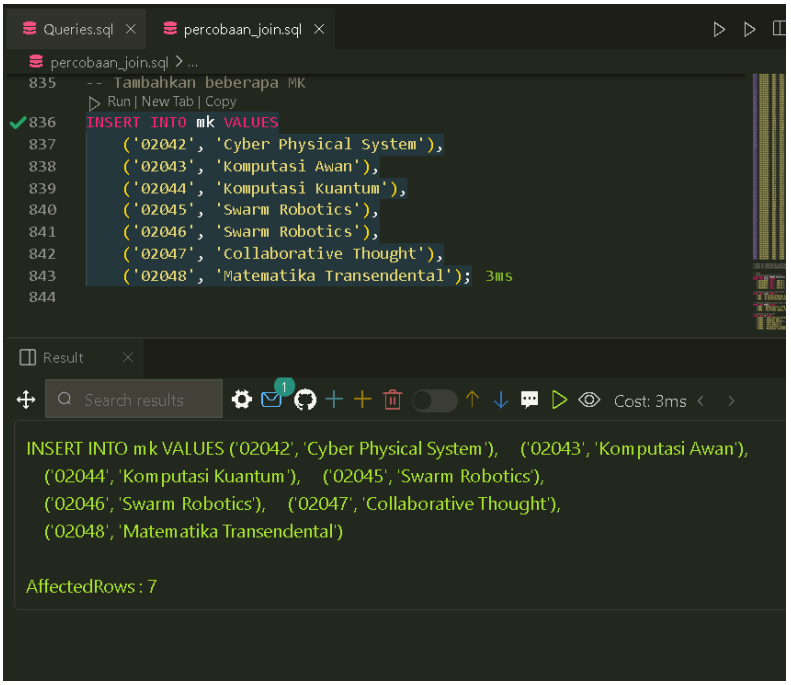
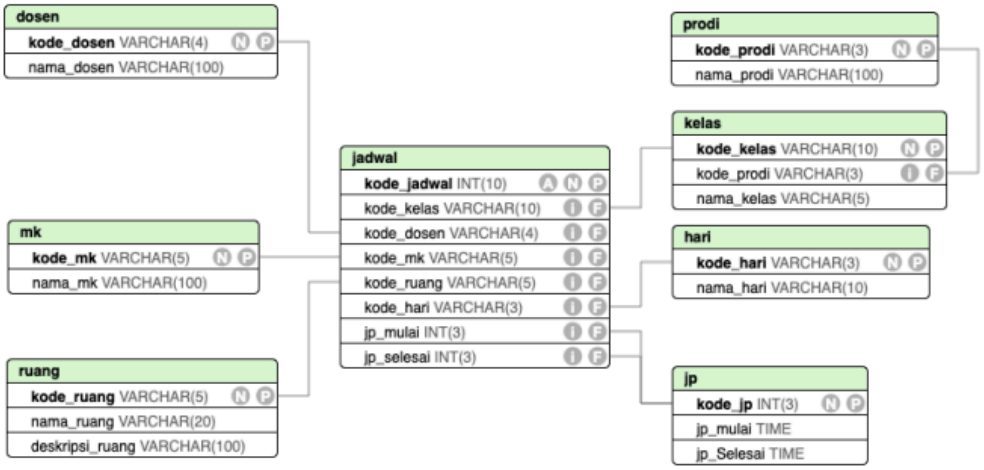
Nama
Muhammad Afif Al Ghifari

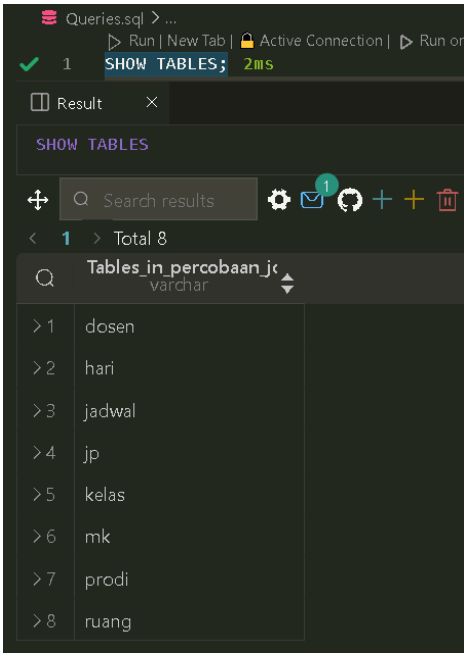
Kelas / Absen
TI-1H / 17

NIM
2341720168

Tanggal
13 Mei 2024

Praktikum 1

Langkah	Keterangan
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.
	

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.
	

Praktikum 2: INNER JOIN

Jadwal	Keterangan
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).

<table>
<tr><td></td><td>

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

Result(RO)

Search results

1 Total 52

	nama_kela 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

</td></tr>
<tr><td>2</td><td>

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!

</td></tr>
<tr><td></td><td>

```
SELECT k.nama_kelas, p.nama_prodi
FROM kelas k
INNER JOIN prodi p ON p.kode_prodi = k.kode_prodi  3ms
```

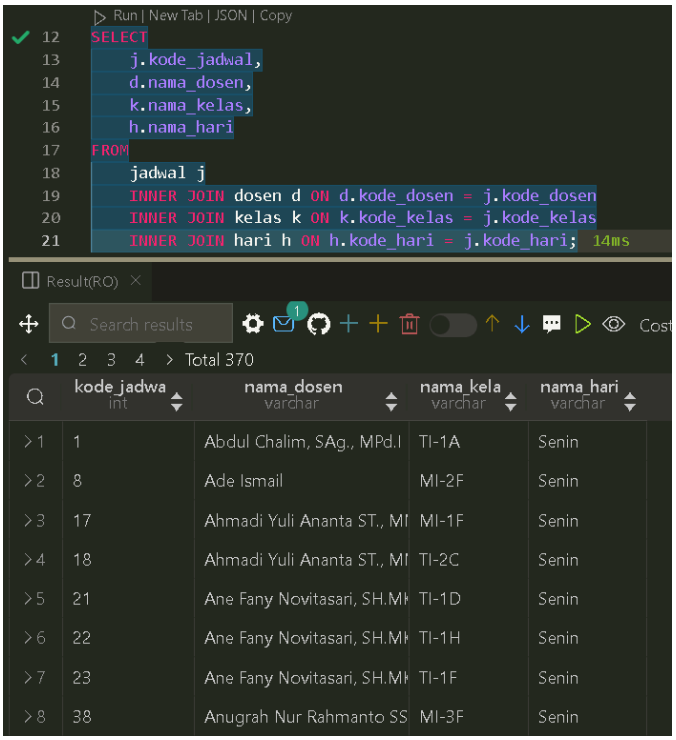
Result(RO)

Search results

1 Total 52

	nama_kela 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

</td></tr>
</table>

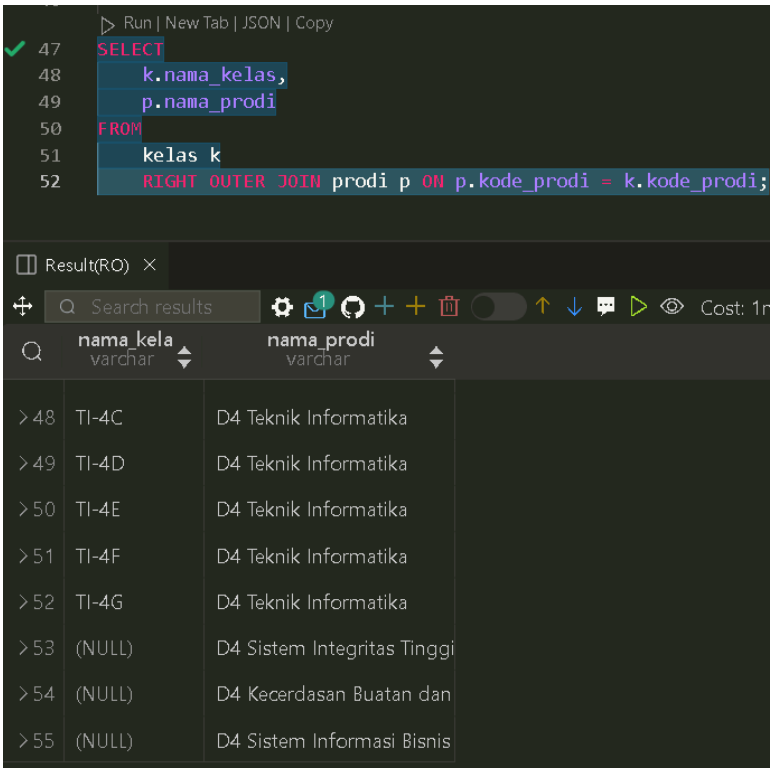
3	Pada contoh sebelumnya kita telah menampilkan 2 kolom yang terletak pada 2 tabel yang berbeda namun penggunaan INNER JOIN tidaklah terbatas pada 2 tabel saja. Kita juga dapat menampilkan data yang lebih banyak dari beberapa tabel sekaligus. Contoh: “Tampilkan nama dosen berikut kelas yang diajar dan harinya!” Solusi: Jalankan SQL berikut. Jika benar akan ditampilkan 320 baris. Pada query tersebut dilibatkan 4 tabel yaitu tabel jadwal, dosen, kelas, dan hari.
	
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 23 SELECT k.nama_kelas, p.nama_prodi 24 FROM kelas k 25 INNER JOIN prodi p ON p.kode_prodi = k.kode_prodi 26 WHERE 27 k.nama_kelas LIKE '%A'; 1ms Result(RO) Search results 1 Total 7 nama_kela varchar / nama_prodi varchar 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: OUTER JOIN

Langkah	Keterangan
1	Jika INNER JOIN hanya menampilan 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 beguna 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 31 SELECT 32 k.nama_kelas, 33 p.nama_prodi 34 FROM 35 kelas k 36 LEFT OUTER JOIN prodi p ON k.kode_prodi = p.kode_prodi; 2ms Result(RO) Search results Cost: 2ms 1 Total 57 nama_kela 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)
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 38 SELECT 39 k.nama_kelas, 40 p.nama_prodi 41 FROM 42 kelas k 43 LEFT OUTER JOIN prodi p ON k.kode_prodi = p.kode_prodi 44 WHERE 45 p.kode_prodi IS NULL; 4ms Result(RO) Search results Cost: 4ms 1 Total 5 nama_kela varchar | nama_proc varchar 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

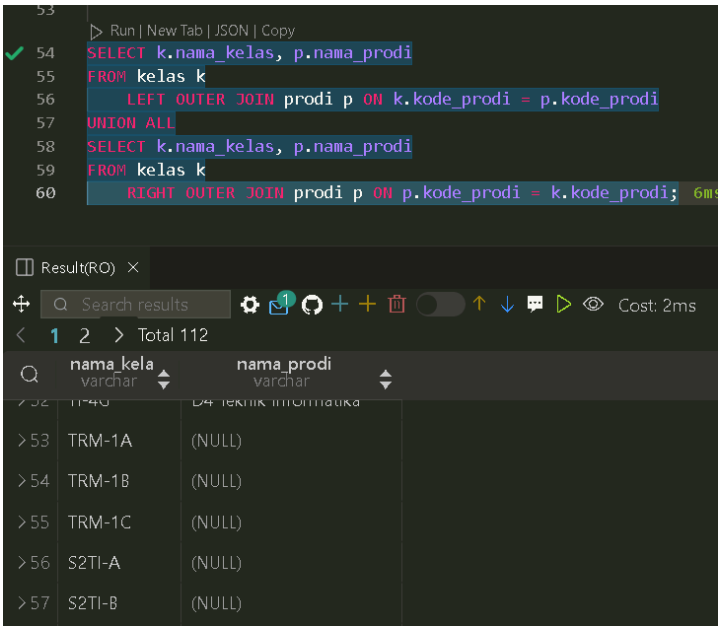
	sama-sama menampilkan data yang bersesuaian ditambah yang tidak bersesuaian. Bedanya dalah letak data yang TIDAK NULL-nya di sebelah mana. Tabel kiri atau tabel kanan. Bagaimana menentukan tabel kiri dan tabel kanan? - Tabel kiri adalah yang ditulis di sebelah KIRI (SEBELUM) kata-kata JOIN. - Tabel kanan 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.
	
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 sintaks 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

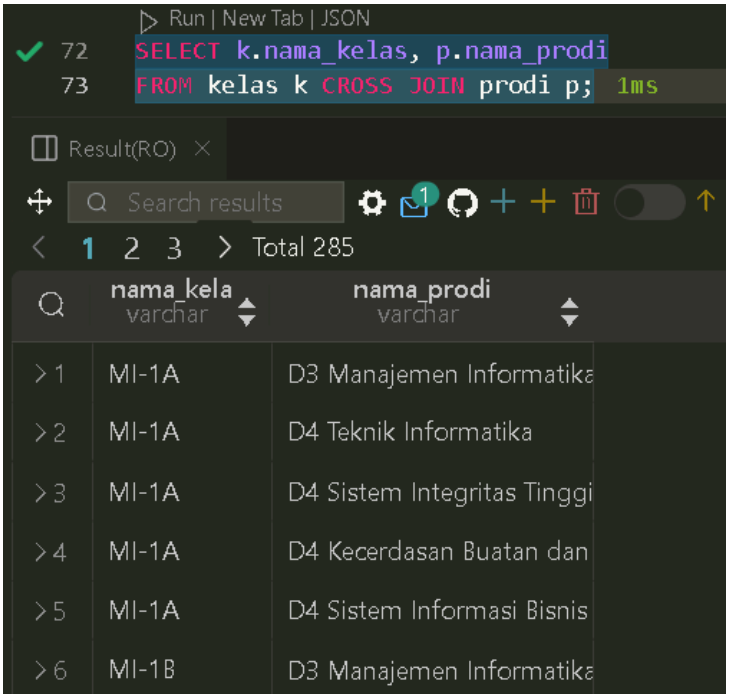
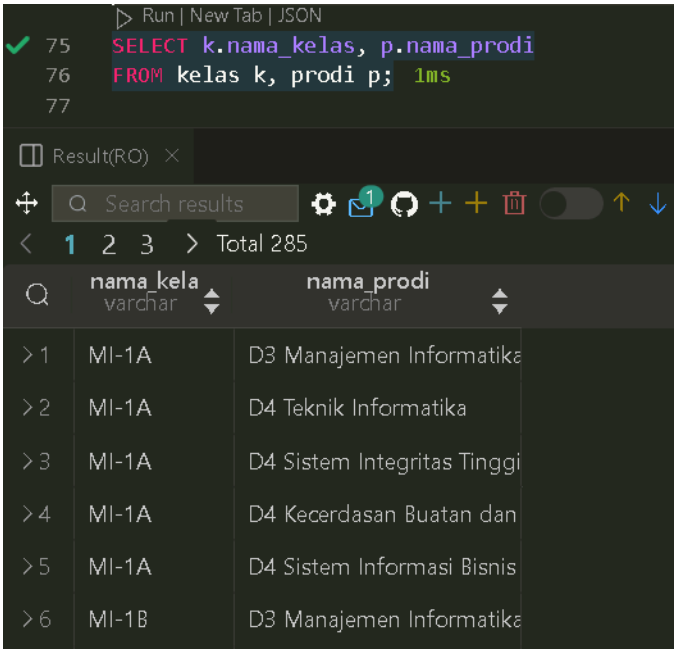


	Result(RO) Search results 1 2 Total 112 nama_kela varchar / nama_prodi varchar 6 TI-4D D4 Teknik Informatika 7 TI-4E D4 Teknik Informatika 8 TI-4F D4 Teknik Informatika 9 TI-4G D4 Teknik Informatika 10 (NULL) D4 Sistem Integritas Tinggi 11 (NULL) D4 Kecerdasan Buatan dan 12 (NULL) D4 Sistem Informasi Bisnis
5	Lanjutkan ke Praktikum – Bagian 5.

Praktikum 4: INNER JOIN Implisit dan CROSS JOIN

Langkah	Keterangan
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 64 SELECT kelas.nama_kelas, prodi.nama_prodi 65 FROM kelas, prodi 66 WHERE kelas.kode_prodi = prodi.kode_prodi; 2ms Result(RO) Search results nama_kela varchar | nama_prodi varchar 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 11 MI-2D 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 68 SELECT k.nama_kelas, p.nama_prodi 69 FROM kelas k, prodi p 70 WHERE k.kode_prodi = p.kode_prodi 1ms Result(RO) Search results nama_kela 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
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.
	
4	CROSS JOIN juga dapt dilakukan secara IMPLICIT yaitu dengan tanpa memberikan WHERE pada sintaksis JOIN IMPLICIT. Jalankan query di bawah ini, hasilnya akan sama dengan langkah sebelumnya.
	

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.

186 rows

	`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

Jawab:

Run | New Tab | JSON | Copy

```sql
SELECT DISTINCT d.nama_dosen
    AS 'Dosen Pengampu', m.nama_mk AS 'Nama Matakuliah'
FROM jadwal j
    JOIN dosen d ON j.kode_dosen = d.kode_dosen
    JOIN mk m ON j.kode_mk = m.kode_mk
ORDER BY d.nama_dosen ASC;  13ms
```

Result(RO)

Search results Cost: 13ms

1 2 > Total 186

	Dosen Pengampu (varchar)	Nama Matakuliah (varchar)
25	Arwin Sumari ST., MT., DR.	Manajemen Proyek
26	Atiqah Nurul Asri SPd., MPd	Bahasa Inggris Persiapan K
27	Atiqah Nurul Asri SPd., MPd	Bahasa Inggris
28	Atiqah Nurul Asri SPd., MPd	Bahasa Inggris 2
29	Bagas Satya Dian Nugraha,	Komputasi Multimedia
30	Banni Satria Andoko, S. Kom	Proyek 1_P2
31	Banni Satria Andoko, S. Kom	Analisis Dan Desan Berorien
32	Banni Satria Andoko, S. Kom	Proyek 1_P1
33	Budi Harijanto ST., MMKom	Proyek 1_P1
34	Budi Harijanto ST., MMKom	Proyek 1_P2
35	Budi Harijanto ST., MMKom	Sistem Operasi
36	Cahya Rahmad ST., MKom.	Pengolahan Citra dan Visi K
37	Cahya Rahmad ST., MKom.	Alajabar Linier

Result(RO)

Search results | Cost: 2ms

1 2 Total 186

	Dosen Pengampu (varchar)	Nama Matakuliah (varchar)
74	Vivin Ayu Lestari, SPd.	Praktikum Struktur Data
75	Vivin Ayu Lestari, SPd.	Rekayasa Perangkat Lunak
76	Widaningsih Condrowardh	Etika Profesi Bidang TI
77	Wilda Imama Sabilla, S.Kon	Praktikum Basis Data
78	Wilda Imama Sabilla, S.Kon	Basis Data
79	Wilda Imama Sabilla, S.Kon	E-Business
80	Yoppy Yunhasnawa, SST., M	Praktikum Basis Data
81	Yoppy Yunhasnawa, SST., M	Teknologi Data
82	Yoppy Yunhasnawa, SST., M	Basis Data
83	Yuri Ariyanto, SKom., MKor	Proyek 1_P1
84	Yuri Ariyanto, SKom., MKor	Manajemen Proyek
85	Yuri Ariyanto, SKom., MKor	Proyek 1_P2
86	Zulmy Faqihuddin Putera, S	Bahasa Indonesia

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

79 rows | CSV

	`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

Jawab:

Run | New Tab | JSON | Copy

```
87  SELECT
88      r.kode_ruang AS 'Kode Ruang',
89      r.nama_ruang AS 'Nama Ruang',
90      h.nama_hari AS 'Hari Dipakai',
91      jm.jp_mulai AS 'Dari Jam',
92      js.jp_selesai AS 'Sampai Jam'
93  FROM jadwal j
94  INNER JOIN ruang r ON j.kode_ruang = r.kode_ruang
95  INNER JOIN hari h ON j.kode_hari = h.kode_hari
96  INNER JOIN jp jm ON j.jp_mulai = jm.kode_jp
97  INNER JOIN jp js ON j.jp_selesai = js.kode_jp
98  WHERE h.nama_hari = 'Selasa'; 10ms
```

Result(RO)

Search results | Cost: 10ms

1 Total 79

	Kode Ruar (varchar)	Nama Rua (varchar)	Hari Dipakai (varchar)	Dari Jam (time)	Sampai Jar (time)
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

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

3 rows

	`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>

Jawab:

Run | New Tab | JSON

```
100 SELECT d.nama_dosen AS 'Nama Dosen', j.kode_jadwal AS 'Kode Jadwal'
101 FROM dosen d
102 LEFT JOIN jadwal j ON d.kode_dosen = j.kode_dosen
103 WHERE j.kode_dosen IS NULL;  5ms
```

Result(RO)

Search results Cost: 5ms 1

	Nama Dosen varchar	Kode Jadwa int
1	Kamado Tanjiro, S.Kom., M.	(NULL)
2	Rei Ayanami, S.ST., M.Sc.	(NULL)
3	Soryu Asuka Langley, M.En	(NULL)

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

192 rows CSV

	`Mata Kuliah`	Pengampu
182	Teknologi Data	Dodit Supriyanto SKom., MT.
183	Teknologi Data	Gunawan Budi Prasetyo, ST., MMT., Ph.D.
184	Teknologi Data	Odhitya Desta Triswidrananta, SPd., MPd.
185	Teknologi Data	Pramana Yoga Saputra, SKom., MMT.
186	Teknologi Data	Yoppy Yunhasnawa, SST., MSc.
187	Cyber Physical System	<null>
188	Komputasi Awan	<null>
189	Komputasi Kuantum	<null>
190	Swarm Robotics	<null>
191	Collaborative Thought	<null>
192	Matematika Transendental	<null>

Jawab:

```sql
SELECT DISTINCT
    m.nama_mk AS 'Mata Kuliah',
    d.nama_dosen AS 'Dosen Pengampu'
FROM jadwal j
RIGHT OUTER JOIN dosen d ON d.kode_dosen = j.kode_dosen
RIGHT OUTER JOIN mk m ON m.kode_mk = j.kode_mk;  10ms
```

Result(RO)

Cost: 3ms

1 2 Total 192

	Mata Kuliah (varchar)	Dosen Pengampu (varchar)
80	Struktur Data	Vivi Nur Wijayaningrum, S.Kom, M.Kom
81	Struktur Data	Vivin Ayu Lestari, SPd.
82	Teknologi Data	Dodit Supriyanto SKom., MT.
83	Teknologi Data	Gunawan Budi Prasetyo, ST., MMT., Ph.D.
84	Teknologi Data	Odhitya Desta Triswidrananta, SPd., MPd.
85	Teknologi Data	Pramana Yoga Saputra, SKom., MMT.
86	Teknologi Data	Yoppy Yunhasnawa, SST., MSc.
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

88 rows

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

Jawab:

Run | New Tab | JSON | Copy

```sql
SELECT d.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.kode_dosen, d.nama_dosen;  15ms
```

Result(RO)

Search results

Cost: 15ms

1 Total 88

	nama_dosen varchar	Jumlah_Jadwal_Menga bigint
81	Widaningsih Condrowardh	7
82	Wilda Imama Sabilla, S.Kon	5
83	Yoppy Yunhasnawa, SST., M	4
84	Yuri Ariyanto, SKom., MKor	5
85	Zulmy Faqihuddin Putera, S	6
86	Kamado Tanjiro, S.Kom., M.	0
87	Rei Ayanami, S.ST., M.Sc.	0
88	Soryu Asuka Langley, M.En	0