

## BASIS DATA

### Dasar MySQL Data Retrieval Language

Lavina 2341760062

#### Praktikum 1

74 `SELECT * FROM jadwal;` 3ms

jadwal

`SELECT * FROM jadwal LIMIT 100`

Search results

Cost: 9ms < 1 2 3 4 > Total 370

		kode_jad int(10)	kode_kel varchar(10)	kode_dos varchar(4)	kode_mk varchar(5)	kode_rua varchar(5)	kode_hari varchar(3)	jp_mulai int(3)	jp_selesai int(3)
> 1	1		2021020101	D001	02001	0504	001	7	9
> 2	2		2021010103	D001	02001	0506	002	9	11
> 3	3		2021010105	D001	02001	0806	003	10	12
> 4	4		2021010102	D001	02001	0506	004	1	3
> 5	5		2021010106	D001	02001	0806	004	4	6
> 6	6		2021010101	D001	02001	0506	004	7	9
> 7	7		2021010104	D001	02001	0506	005	10	12
> 8	8		2021010206	D002	02037	0702	001	7	12
> 9	9		2021020202	D002	02036	0708	003	2	4
> 10	10		2021010205	D002	02037	0713	004	1	6
> 11	11		2021020209	D002	02025	0719	004	7	12
> 12	12		2021020301	D003	02012	0508	002	1	4

74 `SELECT * FROM prodi;` 2ms

prodi

`SELECT * FROM prodi LIMIT 100`

Search results

Cost: 1ms

		kode_pi varchar(3)	nama_prodi varchar(100)
> 1	001		D3 Manajemen Informatika
> 2	002		D4 Teknik Informatika

✓ 74 `SELECT * FROM kelas;` 2ms

kelas

`SELECT * FROM kelas LIMIT 100`

Search results

		kode_kelas varchar(10)	kode_prodi varchar(3)	nama_kel varchar(5)
<input type="checkbox"/>	> 1	2021010101	001	MI-1A
<input type="checkbox"/>	> 2	2021010102	001	MI-1B
<input type="checkbox"/>	> 3	2021010103	001	MI-1C
<input type="checkbox"/>	> 4	2021010104	001	MI-1D
<input type="checkbox"/>	> 5	2021010105	001	MI-1E
<input type="checkbox"/>	> 6	2021010106	001	MI-1F
<input type="checkbox"/>	> 7	2021010107	001	MI-1H
<input type="checkbox"/>	> 8	2021010201	001	MI-2A
<input type="checkbox"/>	> 9	2021010202	001	MI-2B
<input type="checkbox"/>	> 10	2021010203	001	MI-2C
<input type="checkbox"/>	> 11	2021010204	001	MI-2D
<input type="checkbox"/>	> 12	2021010205	001	MI-2E

✓ 74 `SELECT * FROM ruang;` 1ms

ruang

`SELECT * FROM ruang LIMIT 100`

Search results

		kode_ruangan varchar(5)	nama_ruangan varchar(20)	deskripsi_ruangan varchar(100)
<input type="checkbox"/>	> 1	0501	RT01	Ruang Teori 1
<input type="checkbox"/>	> 2	0502	RT02	Ruang Teori 2
<input type="checkbox"/>	> 3	0503	RT03	Ruang Teori 3
<input type="checkbox"/>	> 4	0504	RT04	Ruang Teori 4
<input type="checkbox"/>	> 5	0505	RT05	Ruang Teori 5
<input type="checkbox"/>	> 6	0506	RT06	Ruang Teori 6
<input type="checkbox"/>	> 7	0507	RT07	Ruang Teori 7
<input type="checkbox"/>	> 8	0508	LPY1	Laboratorium Proyek 1
<input type="checkbox"/>	> 9	0615	LSI1	Laboratorium Sistem Inf
<input type="checkbox"/>	> 10	0617	LSI2	Laboratorium Sistem Inf
<input type="checkbox"/>	> 11	0618	LSI3	Laboratorium Sistem Inf

✓ 74 `SELECT * FROM dosen;` 2ms

`SELECT * FROM dosen LIMIT 100`

		kode_dosen varchar(4)	nama_dosen varchar(100)
<input type="checkbox"/>	> 1	D001	Abdul Chalim, SAg., MPd.I
<input type="checkbox"/>	> 2	D002	Ade Ismail
<input type="checkbox"/>	> 3	D003	Agung Nugroho Pramudhita ST., MT.
<input type="checkbox"/>	> 4	D004	Ahmadi Yuli Ananta ST., MM.
<input type="checkbox"/>	> 5	D005	Ane Fany Novitasari, SH.MKn.
<input type="checkbox"/>	> 6	D006	Annisa Puspa Kirana MKom.
<input type="checkbox"/>	> 7	D007	Annisa Taufika Firdausi ST., MT.
<input type="checkbox"/>	> 8	D008	Anugrah Nur Rahmanto SSn., MDs.
<input type="checkbox"/>	> 9	D009	Ariadi Retno Ririd SKom., MKom.
<input type="checkbox"/>	> 10	D010	Arie Rachmad Syulistyo SKom., MKom.
<input type="checkbox"/>	> 11	D011	Arief Prasetyo SKom., MKom.
<input type="checkbox"/>	> 12	D012	Arwin Sumari ST., MT., DR.

✓ 74 `SELECT * FROM hari;` 2ms

`SELECT * FROM hari LIMIT 100`

		kode_hari varchar(3)	nama_hari varchar(10)
<input type="checkbox"/>	> 1	001	Senin
<input type="checkbox"/>	> 2	002	Selasa
<input type="checkbox"/>	> 3	003	Rabu
<input type="checkbox"/>	> 4	004	Kamis
<input type="checkbox"/>	> 5	005	Jumat
<input type="checkbox"/>	> 6	006	Sabtu
<input type="checkbox"/>	> 7	007	Minggu

74 `SELECT * FROM mk;` 2ms

mk

`SELECT * FROM mk LIMIT 100`

Search results

		kode_m varchar(5)	nama_mk varchar(100)
<input type="checkbox"/>	> 1	02001	Agama
<input type="checkbox"/>	> 2	02002	Aljabar Linier
<input type="checkbox"/>	> 3	02003	Algoritma dan Struktur Data
<input type="checkbox"/>	> 4	02004	Aljabar Linier
<input type="checkbox"/>	> 5	02005	Analisis Dan Desain Berorientasi Objek
<input type="checkbox"/>	> 6	02006	Bahasa Indonesia
<input type="checkbox"/>	> 7	02007	Bahasa Inggris
<input type="checkbox"/>	> 8	02008	Bahasa Inggris 2
<input type="checkbox"/>	> 9	02009	Bahasa Inggris Persiapan Kerja
<input type="checkbox"/>	> 10	02010	Basis Data
<input type="checkbox"/>	> 11	02011	Desain Pemrograman Web
<input type="checkbox"/>	> 12	02012	Digital Entrepreneurship

74 `SELECT * FROM jp;` 1ms

jp

`SELECT * FROM jp LIMIT 100`

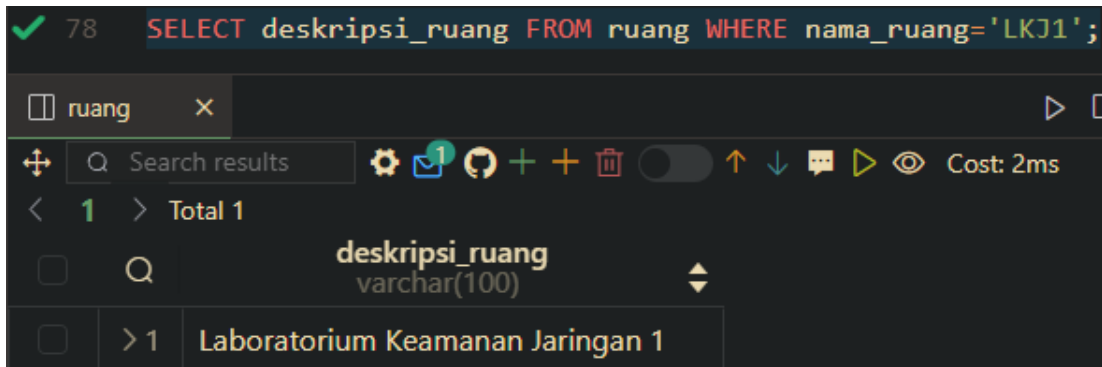
Search results

		kode_jp int(3)	jp_mulai time	jp_selesai time
<input type="checkbox"/>	> 1	1	07:00:00	07:50:00
<input type="checkbox"/>	> 2	2	07:50:00	08:40:00
<input type="checkbox"/>	> 3	3	08:40:00	09:30:00
<input type="checkbox"/>	> 4	4	09:40:00	10:30:00
<input type="checkbox"/>	> 5	5	10:30:00	11:20:00
<input type="checkbox"/>	> 6	6	11:20:00	12:10:00
<input type="checkbox"/>	> 7	7	12:50:00	13:40:00
<input type="checkbox"/>	> 8	8	13:40:00	14:30:00
<input type="checkbox"/>	> 9	9	14:30:00	15:20:00
<input type="checkbox"/>	> 10	10	15:30:00	15:30:00
<input type="checkbox"/>	> 11	11	16:20:00	17:10:00
<input type="checkbox"/>	> 12	12	17:10:00	18:00:00

## Praktikum 2

1. Untuk menampilkan baris dengan suatu kondisi tertentu, digunakan statement **SELECT** dengan **WHERE**. Eksekusi SQL berikut untuk menampilkan (record) pada tabel **Ruang** yang memiliki kolom nama\_ruang = LKJ1.

```
✓ 78 SELECT deskripsi_ruang FROM ruang WHERE nama_ruang='LKJ1';
```

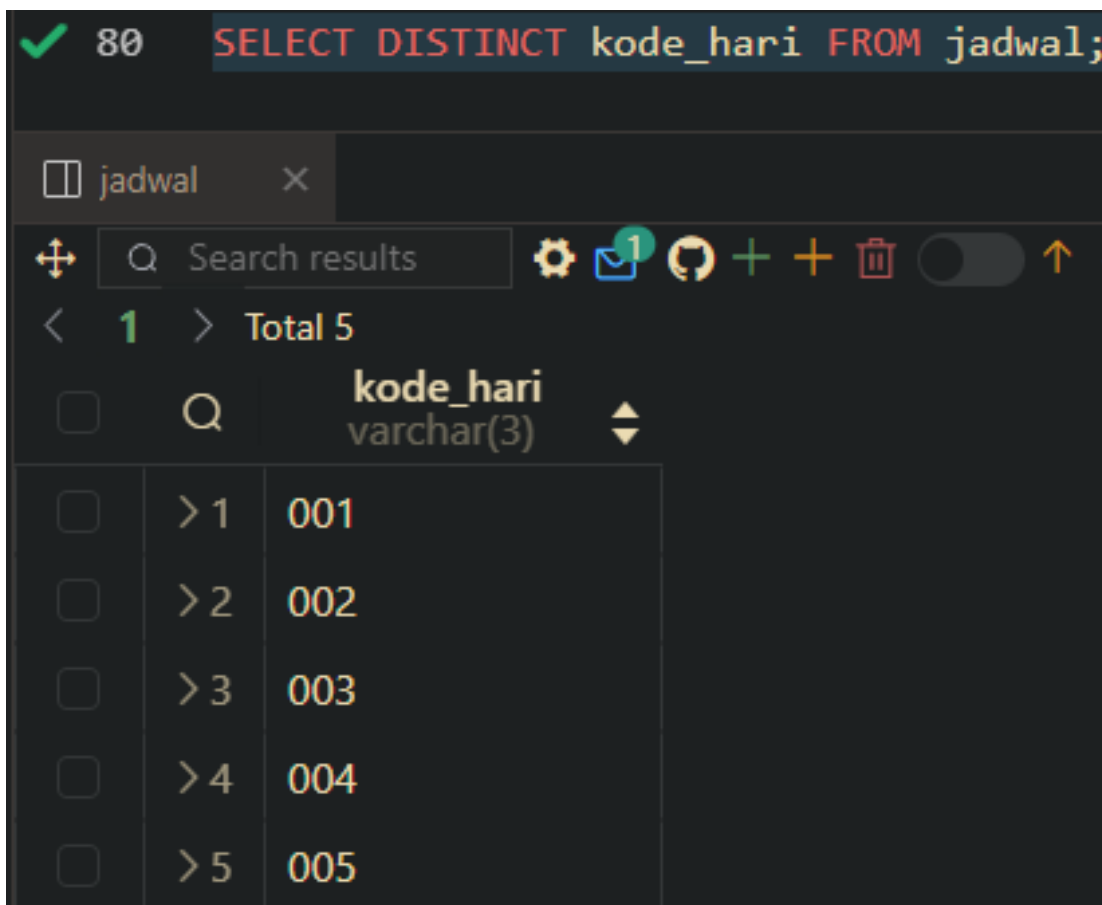


The screenshot shows a database interface with a query window titled 'ruang'. The query is `SELECT deskripsi_ruang FROM ruang WHERE nama_ruang='LKJ1';`. The results pane shows a single record with the column 'deskripsi\_ruang' (varchar(100)) containing the value 'Laboratorium Keamanan Jaringan 1'. The total number of records is 1.

deskripsi_ruang varchar(100)
Laboratorium Keamanan Jaringan 1

2. Untuk menampilkan data dengan eliminasi data yang sama (duplicate), digunakan statement **SELECT** dengan **DISTINCT**. Eksekusi SQL berikut untuk menampilkan kolom kode\_hari dari tabel jadwal yang bernilai tidak sama.

```
✓ 80 SELECT DISTINCT kode_hari FROM jadwal;
```



The screenshot shows a database interface with a query window titled 'jadwal'. The query is `SELECT DISTINCT kode_hari FROM jadwal;`. The results pane shows five distinct records for the column 'kode\_hari' (varchar(3)), with values 001, 002, 003, 004, and 005. The total number of records is 5.

kode_hari varchar(3)
001
002
003
004
005



kode\_dosen, dan nama\_dosen, pada tabel **dosen** yang memiliki nama dengan huruf awal 'E'.

✓ 86 `SELECT * FROM dosen WHERE nama_dosen LIKE 'E%';`

dosen

Search results

< 1 > Total 4

		kode_dosen varchar(4)	nama_dosen varchar(100)
<input type="checkbox"/>	> 1	D028	Eka Larasati Amalia, SST., MT.
<input type="checkbox"/>	> 2	D029	Ekojono, ST., M.Kom.
<input type="checkbox"/>	> 3	D030	Elok Nur Hamdana, ST., MT
<input type="checkbox"/>	> 4	D031	Erfan Rohadi, ST., MEng., PhD.

6. Untuk menampilkan susunan data dalam bentuk grup, digunakan **SELECT** dengan **GROUP BY**. Eksekusi SQL berikut untuk menampilkan kolom kode\_dosen, kode\_mk, kode\_ruang dan kode\_hari pada tabel **jadwal** yang dikelompokkan berdasarkan kode\_hari.

✓ 88 `SELECT kode_dosen, kode_mk, kode_ruang, kode_hari`  
89 `FROM jadwal GROUP BY kode_hari; 2ms`

jadwal

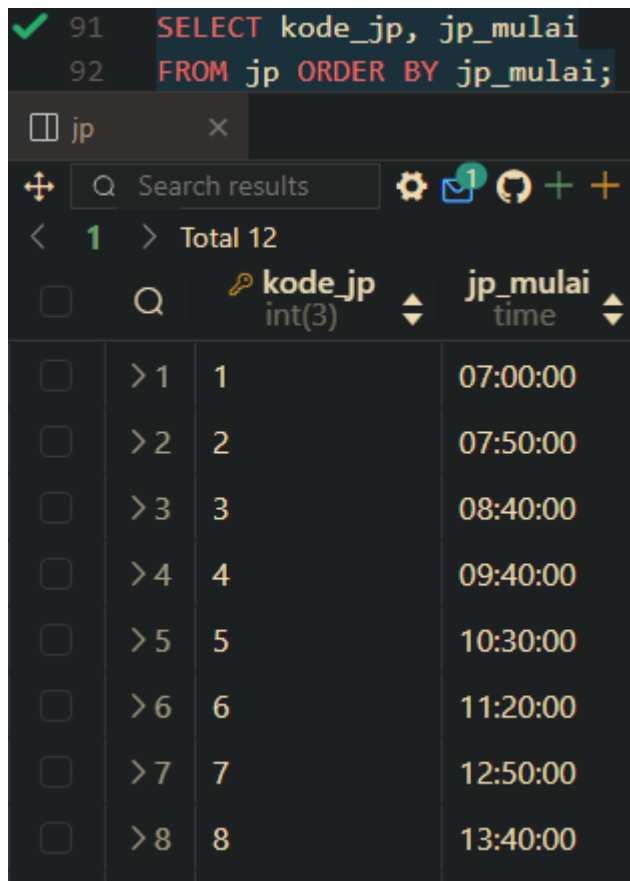
Search results

< 1 > Total 5

		kode_dosen varchar(4)	kode_mk varchar(5)	kode_ruang varchar(5)	kode_hari varchar(3)
<input type="checkbox"/>	> 1	D001	02001	0504	001
<input type="checkbox"/>	> 2	D001	02001	0506	002
<input type="checkbox"/>	> 3	D001	02001	0806	003
<input type="checkbox"/>	> 4	D001	02001	0506	004
<input type="checkbox"/>	> 5	D001	02001	0506	005

Cost: 4ms

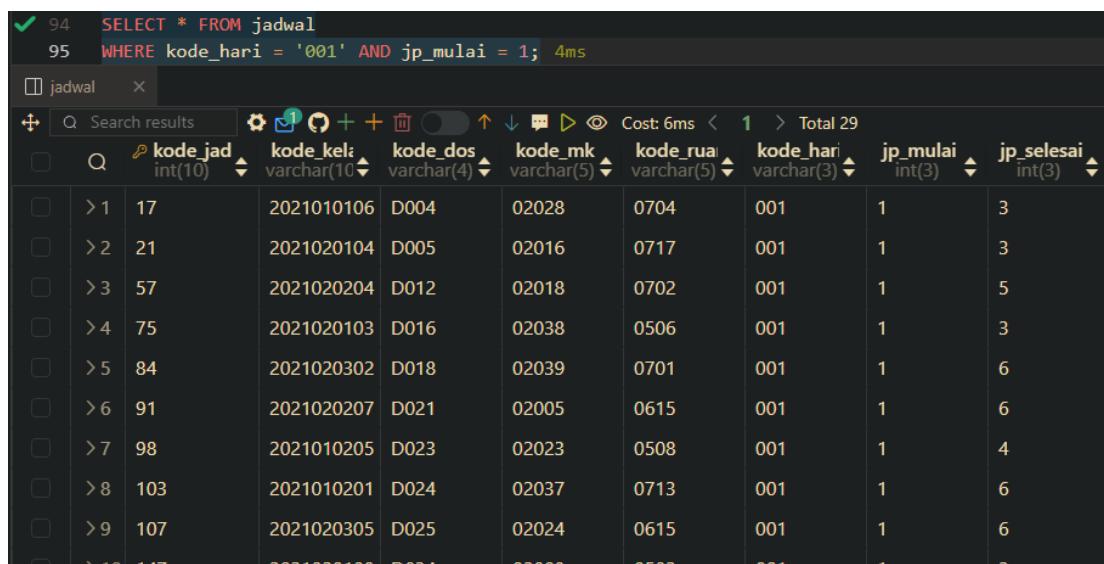
7. Untuk menampilkan baris secara spesifik dan terurut maju atau mundur, digunakan **SELECT** dengan **ORDER BY**. Eksekusi SQL berikut untuk menampilkan kolom `kode_jp` dan `jp_mulai` pada tabel `jp` dari jam yang paling mulai hingga berakhir.



```
91 SELECT kode_jp, jp_mulai
92 FROM jp ORDER BY jp_mulai;
```

		kode_jp int(3)	jp_mulai time
> 1	1		07:00:00
> 2	2		07:50:00
> 3	3		08:40:00
> 4	4		09:40:00
> 5	5		10:30:00
> 6	6		11:20:00
> 7	7		12:50:00
> 8	8		13:40:00

8. Untuk menampilkan data dengan kondisi dan atau atau tidak, digunakan **SELECT** dengan **AND**, **OR** and **NOT**. Eksekusi SQL berikut untuk menampilkan semua kolom pada tabel `jadwal` dengan `kode_hari = '001'` dan `jp_mulai = 1`.



```
94 SELECT * FROM jadwal
95 WHERE kode_hari = '001' AND jp_mulai = 1; 4ms
```

	kode_jad int(10)	kode_kel varchar(10)	kode_dos varchar(4)	kode_mk varchar(5)	kode_ruai varchar(5)	kode_hari varchar(3)	jp_mulai int(3)	jp_selesai int(3)
> 1	17	2021010106	D004	02028	0704	001	1	3
> 2	21	2021020104	D005	02016	0717	001	1	3
> 3	57	2021020204	D012	02018	0702	001	1	5
> 4	75	2021020103	D016	02038	0506	001	1	3
> 5	84	2021020302	D018	02039	0701	001	1	6
> 6	91	2021020207	D021	02005	0615	001	1	6
> 7	98	2021010205	D023	02023	0508	001	1	4
> 8	103	2021010201	D024	02037	0713	001	1	6
> 9	107	2021020305	D025	02024	0615	001	1	6
> 10	117	2021020108	D024	02008	0503	001	1	3



9. Untuk menampilkan data dari kolom yang terlibat dalam dua tabel dapat digunakan **SELECT** dengan **UNION**. **UNION** secara otomatis akan menghilangkan duplikasi. Eksekusi SQL berikut untuk menampilkan kolom kode\_hari yang ada di tabel hari atau jadwal.

```
✓ 97 SELECT kode_hari FROM hari
98 UNION
99 SELECT kode_hari FROM jadwal;
```

Result(RO) ×

Search results

1

< 1 > Total 7

kode\_hari  
varchar

> 1	001
> 2	002
> 3	003
> 4	004
> 5	005
> 6	006
> 7	007

10. Untuk menampilkan data dari kolom yang terlibat dalam dua tabel dapat digunakan **SELECT** dengan **UNION ALL**. **UNION ALL** akan menampilkan duplikasi data. Eksekusi SQL berikut untuk menampilkan kolom nik yang ada di tabel **penugasan** atau **departemen**.

```
✓ 101 SELECT kode_hari FROM hari
102 UNION ALL
103 SELECT kode_hari FROM jadwal;
```

Result(RO) ×

Search results

1

< 1 2 3 4 > Total 377

kode\_hari  
varchar

> 1	001
> 2	002
> 3	003
> 4	004
> 5	005
> 6	006
> 7	007
> 8	001
> 9	001

### Praktikum 3

1. Untuk menampilkan data (test keanggotaan sub-query) yang berasal dari pemilihan tampilan data lain digunakan tambahan statement IN. Eksekusi SQL berikut untuk menampilkan kolom kode\_dosen, kode\_mk, kode\_hari, jp\_mulai dan jp\_selesai pada tabel jadwal dimana jp\_selesai adalah jp\_selesai yang paling lama pada tabel jadwal.

```
✓ 105 SELECT kode_dosen, kode_mk, kode_hari, jp_mulai, jp_selesai
106 FROM jadwal WHERE jp_selesai
107 IN (SELECT MAX(jp_selesai) FROM jadwal); 7ms
```

Result(RO) ×

Search results

1

Cost: 7ms

< 1 > Total 99

	kode_dos varchar	kode_mk varchar	kode_hari varchar	jp_mulai int	jp_selesai int
> 1	D001	02001	003	10	12
> 2	D001	02001	005	10	12
> 3	D002	02037	001	7	12
> 4	D002	02025	004	7	12
> 5	D004	02032	001	10	12
> 6	D004	02034	002	7	12
> 7	D005	02016	001	10	12
> 8	D006	02037	004	7	12
> 9	D006	02037	005	7	12
> 10	D007	02011	004	8	12

2. Untuk menampilkan data (perbandingan himpunan sub-query) yang berasal dari pemilihan tampilan data lain digunakan tambahan operator. Eksekusi SQL berikut untuk menampilkan kode\_dosen, kode\_mk, kode\_hari, jp\_mulai dan jp\_selesai pada tabel jadwal berdasarkan semua record kolom jp\_selesai harus bernilai kurang dari jp\_selesai yang dimiliki oleh jp\_selesai = 6 dari tabel jadwal.

```

109 SELECT kode_dosen, kode_mk, kode_hari, jp_mulai, jp_selesai
110 FROM jadwal WHERE jp_selesai < ALL
111 (SELECT jp_selesai FROM jadwal WHERE jp_selesai = 6); 3ms

```

Result(RO) X

Search results

Cost: 3ms

< 1 2 > Total 112

	kode_dos varchar	kode_mk varchar	kode_hari varchar	jp_mulai int	jp_selesai int
> 1	D001	02001	004	1	3
> 2	D002	02036	003	2	4
> 3	D003	02012	002	1	4
> 4	D003	02012	002	1	4
> 5	D003	02017	003	2	5
> 6	D003	02017	003	2	5
> 7	D003	02017	005	2	5

#### Praktikum 4

1. Untuk menghitung rata-rata, digunakan tambahan statement AVG. Eksekusi SQL berikut untuk menampilkan rata-rata durasi perkuliahan dalam satu jadwal perkuliahan sesuai kolom jp\_mulai, jp\_selesai dari tabel jadwal.

```

113 SELECT AVG(jp_selesai - jp_mulai) FROM jadwal;

```

jadwal X

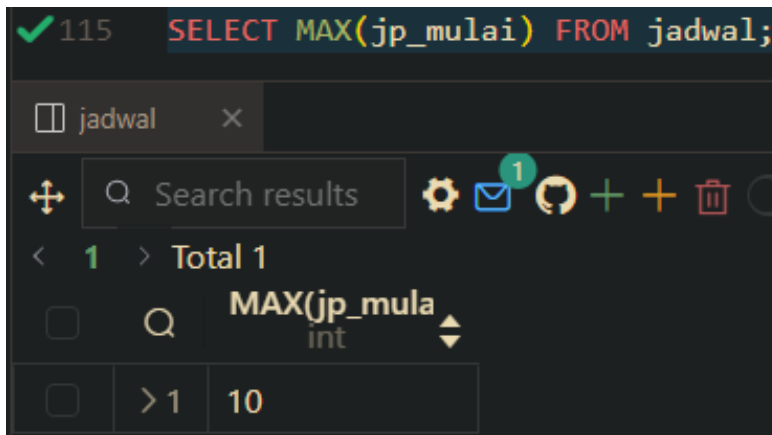
Search results

Cost: 1

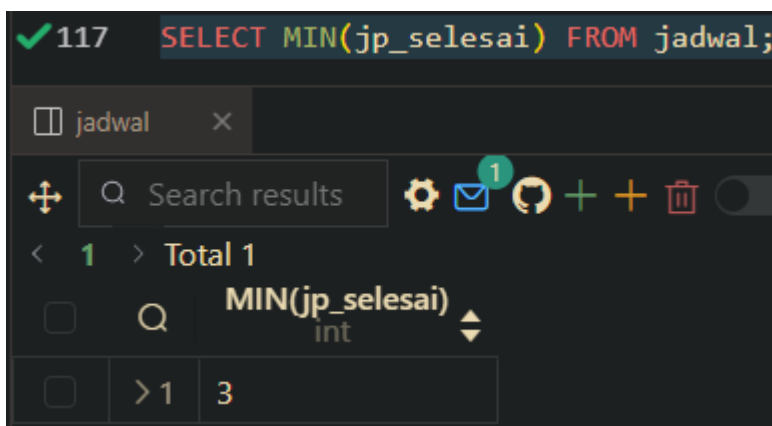
< 1 > Total 1

	AVG(jp_selesai - jp_r newdecimal
> 1	3.2216

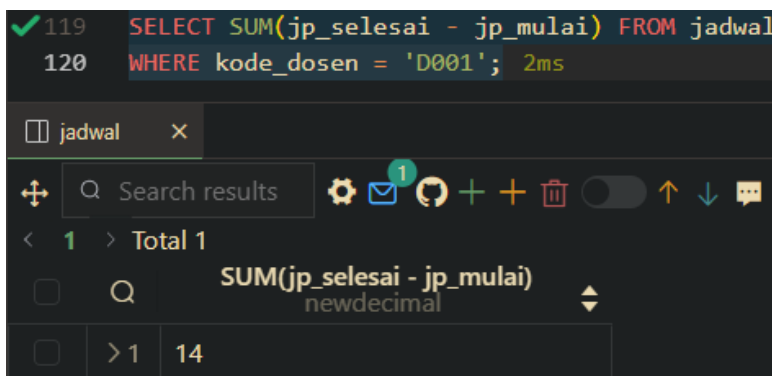
2. Untuk menghitung nilai tertinggi, digunakan tambahan statement MAX. Eksekusi SQL berikut untuk menampilkan dimulainya jam perkuliahan yang paling siang sesuai kolom jp\_mulai dari tabel jadwal.



- Untuk menghitung nilai terendah, digunakan tambahan statement MIN. Eksekusi SQL berikut untuk menampilkan berakhirnya jam perkuliahan yang paling cepat sesuai kolom jp\_selesai dari tabel jadwal.



- Untuk menghitung total jam ajar digunakan tambahan statement SUM. Eksekusi SQL berikut untuk menampilkan total jam ajar dosen dengan kode 'D001' sesuai jp\_mulai, jp\_selesai, dan kode\_dosen dari tabel jadwal.



✓ 122 `SELECT * FROM jadwal WHERE kode_dosen = 'D001';` 1ms

jadwal

Search results

Cost: 3ms < 1 > Total 7

		kode_jad int(10)	kode_kel: varchar(10)	kode_dos varchar(4)	kode_mk varchar(5)	kode_rua: varchar(5)	kode_hari varchar(3)	jp_mulai int(3)	jp_selesai int(3)
> 1	1		2021020101	D001	02001	0504	001	7	9
> 2	2		2021010103	D001	02001	0506	002	9	11
> 3	3		2021010105	D001	02001	0806	003	10	12
> 4	4		2021010102	D001	02001	0506	004	1	3
> 5	5		2021010106	D001	02001	0806	004	4	6
> 6	6		2021010101	D001	02001	0506	004	7	9
> 7	7		2021010104	D001	02001	0506	005	10	12

5. Untuk menunjukkan jumlah baris pada kolom yang diinginkan digunakan COUNT(). Eksekusi SQL berikut untuk menampilkan jumlah baris pada kolom kode\_ruang dari tabel ruang.

✓ 124 `SELECT count(kode_ruang) FROM ruang;`

ruang

Search results

< 1 > Total 1

		count(kode_ruang) bigint
> 1	36	

## Tugas

1. Himpunlah screenshot sintaksis SQL berikut hasil SELECT yang Anda dapat di komputer Anda pada bagian Praktikum dalam sebuah laporan dengan format PDF!
2. Bagaimana sintaksis untuk menampilkan semua kolom jadwal pada hari Kamis yang dimulai pada pukul 08:40 ? Tunjukkan outputnya!

✓ 126 `SELECT * FROM jadwal WHERE kode_hari = '004' AND jp_mulai = 3;` 1ms

jadwal

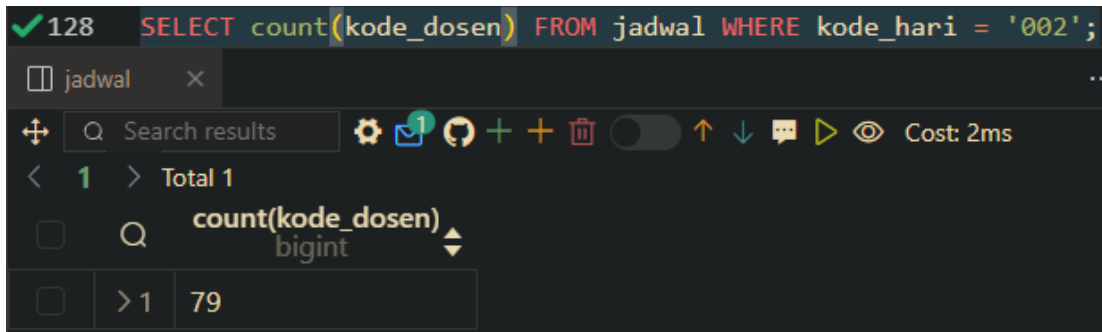
Search results

Cost: 3ms < 1 > Total 3

		kode_jad int(10)	kode_kel: varchar(10)	kode_dos varchar(4)	kode_mk varchar(5)	kode_rua: varchar(5)	kode_hari varchar(3)	jp_mulai int(3)	jp_selesai int(3)
> 1	80		2021020106	D017	02004	0805	004	3	6
> 2	194		2021020206	D044	02020	0707	004	3	5
> 3	195		2021020207	D044	02020	0707	004	3	5

3. Berapakah jumlah dosen yang mengajar pada hari selasa! Tunjukkan sintaksis dan outputnya! Kemudian tampilkan list semua kode dosen yang sesuai dengan kondisi tersebut! Tunjukkan sintaksis dan outputnya!

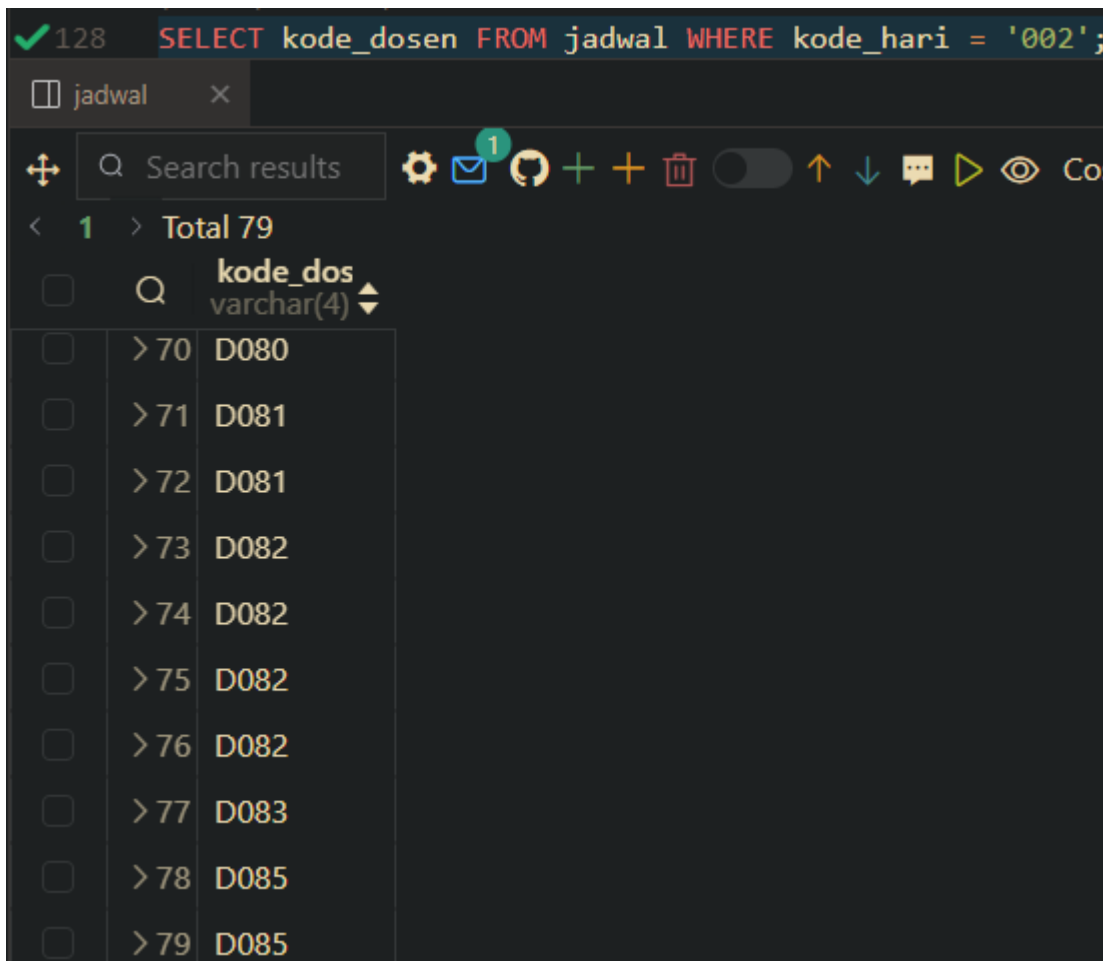
```
✓ 128 SELECT count(kode_dosen) FROM jadwal WHERE kode_hari = '002';
```



The screenshot shows the results of the query `SELECT count(kode_dosen) FROM jadwal WHERE kode_hari = '002';`. The interface includes a search bar, a toolbar with icons for settings, email, refresh, and other functions, and a status bar indicating a cost of 2ms. The results table has one column, `count(kode_dosen)`, with a data type of `bigint`. The first row shows a value of 79.

count(kode_dosen)
79

```
✓ 128 SELECT kode_dosen FROM jadwal WHERE kode_hari = '002';
```



The screenshot shows the results of the query `SELECT kode_dosen FROM jadwal WHERE kode_hari = '002';`. The interface is similar to the first screenshot. The results table has one column, `kode_dosen`, with a data type of `varchar(4)`. The first row shows a value of D080. The total number of rows is 79.

kode_dosen
D080
D081
D081
D082
D082
D082
D082
D082
D083
D085
D085

4. Tampilkan kode dosen yang mengajar pada hari kamis dari tabel jadwal yang nama depannya diawali dengan huruf A. gunakan operasi intersect! Tunjukkan sintaksis dan output dari pencarian tersebut.

```

✓ 132 (SELECT DISTINCT kode_dosen FROM jadwal WHERE kode_hari = '004')
133 INTERSECT
134 (SELECT DISTINCT kode_dosen FROM jadwal
135 WHERE kode_dosen IN (
136     SELECT kode_dosen FROM dosen WHERE nama_dosen LIKE 'A%'
137 )); 3ms

```

Result(RO) X

Search results

kode\_dosen  
varchar

> 2	D002
> 3	D004
> 4	D005
> 5	D006
> 6	D007
> 7	D008
> 8	D011
> 9	D013

5. Tampilkan kode\_ruang tertentu pada hari tertentu dengan diurutkan berdasarkan dengan kode jam yang ter awal.

```

✓ 130 SELECT kode_ruang, kode_hari, jp_mulai FROM jadwal WHERE kode_hari = '001' ORDER BY jp_mulai;

```

jadwal X

```

SELECT kode_ruang, kode_hari, jp_mulai FROM jadwal WHERE kode_hari = '001' ORDER BY jp_mulai LIMIT 1

```

Search results

kode\_ruang  
varchar(5)

kode\_hari  
varchar(3)

jp\_mulai  
int(3)

> 28	0708	001	1
> 29	0504	001	1
> 30	0619	001	2
> 31	0705	001	2
> 32	0703	001	2
> 33	0703	001	2
> 34	0806	001	3
> 35	0706	001	3
> 36	0504	001	4
> 37	0716	001	4
> 38	0704	001	4
> 39	0507	001	4
> 40	0507	001	4

Cost: 3ms < 1 > Total 89