

JOBSHEET

PRAKTIKUM BASIS DATA

Jurusan Teknologi Informasi
POLITEKNIK NEGERI MALANG



PERTEMUAN 12

MYSQL – DATA RETRIEVAL LANGUAGE

Team Teaching:

Dwi Puspitasari, S.Kom., M.Kom.

Yan Watequlis Syaifudin, ST., MMT., PhD.

Annisa Puspa Kirana, S. Kom, M.Kom Yoppy

Yunhasnawa, S.ST., M.Sc.

Candra Bella VistaRetno Damayanti, S.Pd., M.T.



Jurusan Teknologi Informasi Politeknik Negeri Malang
**Jobsheet-12: MySQL – Data Retrieval Language
(DRL) Mata Kuliah Basis Data**

Pengampu: Tim Ajar Basis Data

NAME : Fitri cahyaniati(11)

Mei 2024

Topik

Data Retrieval Language (DRL) pada DBMS MySQL

Tujuan

Mahasiswa diharapkan dapat memahami penggunaan dan dapat membuat SQL statement SELECT dengan berbagai klausa.

Pendahuluan

DRL atau DSL (Data Selection Language) adalah set perintah yang digunakan untuk mengambil data dari server basis data. DRL terdiri dari klausa yang dapat dikembangkan lebih lanjut, antara lain:

1. **FROM** : Memilih nama sebuah tabel
2. **WHERE** : Menunjukkan secara spesifik suatu row/baris yang akan dicari
3. **GROUP BY** : Menyusung data ke dalam grup
4. **HAVING** : Memilih diantara kelompok-kelompok data yang didefinisikan klausa GROUP BY
5. **ORDER BY** : Secara spesifik digunakan untuk menentukan suatu baris pada kolom tertentu
6. **AS** : Memberikan nama alias sementara untuk tabel atau kolom

DRL lebih diterapkan dalam beberapa hal dalam beberapa hal seperti :

1. **QUERY**
2. **SUB-QUERY**
3. **FUNGSI AGREGASI**

QUERY

Query merupakan operasi yang melibatkan satu atau lebih tabel untuk melakukan retrieval data.

Pengambilan data dilakukan dengan beberapa klausa berikut :

1. **SELECT *** untuk memilih semua kolomFormat

SELECT * **FROM** *table_name*

```
MariaDB [jadwal_perkuliahan]> select * from dosen;
```

kode_dosen	nama_dosen
D001	Abdul Chalim, SAg., MPd.I
D002	Ade Ismail
D003	Agung Nugroho Pramudhita ST., MT.
D004	Ahmadi Yuli Ananta ST., MM.
D005	Ane Fany Novitasari, SH.MKn.
D006	Annisa Puspa Kirana MKom.
D007	Annisa Taufika Firdausi ST., MT.
D008	Anugrah Nur Rahmanto SSn., MDs.
D009	Ariadi Retno Ririd SKom., MKom.
D010	Arie Rachmad Syulistyo SKom., MKom.
D011	Arief Prasetyo SKom., MKom.
D012	Arwin Sumari ST., MT., DR.
D013	Atiqah Nurul Asri SPd., MPd.
D014	Bagas Satya Dian Nugraha, ST., MT.
D015	Banni Satria Andoko, S. Kom., M.MSI
D016	Budi Harijanto ST., MMKom.
D017	Cahya Rahmad ST., MKom. DR.Eng
D018	Candra Bella Vista SKom., MT.
D019	Candrasena Setiadi ST., MMT.
D020	Deasy Sandhya Elya Ikawati SSi., MSi.
D021	Deddy Kusbianto PA Ir. MMKom.
D022	Dhebys Suryani SKom. MT.
D023	Dian Hanifudin Subhi SKom., MT.
D024	Dika Rizky Yuniarto SKom., MKom.
D025	Dimas Wahyu Wibowo ST., MT.

2. **SELECT** dengan **WHERE** untuk menampilkan baris dengan suatu kondisiFormat

SELECT *column1, column2,...*
FROM *table_name*
WHERE *condition;*

```
MariaDB [jadwal_perkuliahan]> SELECT kode_dosen, nama_dosen  
-> FROM dosen  
-> WHERE nama_dosen = 'Abdul Chalim, SAg., MPd.I';
```

kode_dosen	nama_dosen
D001	Abdul Chalim, SAg., MPd.I

1 row in set (0.001 sec)

3. **SELECT** dengan **DISTINCT** untuk menampilkan data dengan eliminasi data yang sama (duplicate)Format

SELECT DISTINCT *column* **FROM** *table_name;*

```
MariaDB [jadwal_perkuliahan]> SELECT DISTINCT nama_dosen  
-> FROM dosen;
```

nama_dosen
Abdul Chalim, SAg., MPd.I
Ade Ismail
Agung Nugroho Pramudhita ST., MT.
Ahmadi Yuli Ananta ST., MM.
Ane Fany Novitasari, SH.MKn.
Annisa Puspa Kirana MKom.
Annisa Taufika Firdausi ST., MT.
Anugrah Nur Rahmanto SSn., MDs.
Ariadi Retno Ririd SKom., MKom.
Arie Rachmad Syulistyo SKom., MKom.
Arief Prasetyo SKom., MKom.

4. **SELECT** dengan **IN** untuk menampilkan data yang spesifikFormat

SELECT *column_name(s)*

FROM *table_name*
WHERE *column_name* **IN** (*value1*, *value2*, ...);

```
MariaDB [jadwal_perkuliahan]> SELECT kode_dosen, nama_dosen  
-> FROM dosen  
-> WHERE kode_dosen IN ('D001', 'D002');
```

kode_dosen	nama_dosen
D001	Abdul Chalim, SAg., MPd.I
D002	Ade Ismail

```
2 rows in set (0.001 sec)
```

5. SELECT dengan BETWEEN untuk menampilkan data pada jarak (range) tertentu

Format

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

```
MariaDB [jadwal_perkuliahan]> SELECT kode_dosen, nama_dosen
-> FROM dosen
-> WHERE kode_dosen BETWEEN 'D001' AND 'D005';
+-----+-----+
| kode_dosen | nama_dosen |
+-----+-----+
| D001      | Abdul Chalim, SAg., MPd.I |
| D002      | Ade Ismail |
| D003      | Agung Nugroho Pramudhita ST., MT. |
| D004      | Ahmadi Yuli Ananta ST., MM. |
| D005      | Ane Fany Novitasari, SH.MKn. |
+-----+-----+
5 rows in set (0.001 sec)
```

6. SELECT dengan LIKE untuk menampilkan data yang memiliki kemiripan dengan keyword yang diinginkan

Format

```
SELECT column1,column2,...
FROM table_name
WHERE columnN LIKE pattern;
```

```
MariaDB [jadwal_perkuliahan]> SELECT kode_dosen, nama_dosen
-> FROM dosen
-> WHERE nama_dosen LIKE '%Abdul%';
+-----+-----+
| kode_dosen | nama_dosen |
+-----+-----+
| D001      | Abdul Chalim, SAg., MPd.I |
| D049      | Moch. Zawaruddin Abdullah, S.ST., M.Kom |
+-----+-----+
2 rows in set (0.001 sec)
```

7. SELECT dengan GROUP BY untuk menampilkan susunan data dalam bentuk grup

Format

```
SELECT column1, column2,...
condition
FROM table_name
GROUP BY column1, column2, ...;
```

```
MariaDB [jadwal_perkuliahan]> SELECT departemen, AVG(gaji) AS rata_gaji
-> FROM dosen
-> GROUP BY departemen;
ERROR 1054 (42S22): Unknown column 'departemen' in 'field list'
MariaDB [jadwal_perkuliahan]> |
```

8. SELECT dengan ORDER BY untuk menampilkan baris secara spesifik dan terurut maju atau mundur

Format

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

```
MariaDB [jadwal_perkuliahan]> SELECT *
-> FROM dosen
-> ORDER BY gaji DESC;
ERROR 1054 (42S22): Unknown column 'gaji' in 'order clause'
```

9. SELECT dengan AND, OR and NOT untuk menampilkan data dengan kondisi dan atau atau tidak

Format

AND

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;
```

```
OR
SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;
```

```
NOT
SELECT column1, column2, ...
FROM table_name
WHERE NOT condition;
```

```
MariaDB [jadwal_perkuliahan]> SELECT *
-> FROM dosen
-> WHERE departemen = 'Teknik Informatika' AND gaji > 5000000;
ERROR 1054 (42S22): Unknown column 'departemen' in 'where clause'
MariaDB [jadwal_perkuliahan]> SELECT *
-> FROM dosen
-> WHERE departemen = 'Teknik Informatika' OR departemen = 'Sistem I
nformasi';
ERROR 1054 (42S22): Unknown column 'departemen' in 'where clause'
MariaDB [jadwal_perkuliahan]> SELECT *
-> FROM dosen
-> WHERE NOT departemen = 'Teknik Informatika';
ERROR 1054 (42S22): Unknown column 'departemen' in 'where clause'
MariaDB [jadwal_perkuliahan]> SELECT *
-> FROM dosen
-> WHERE (departemen = 'Teknik Informatika' OR departemen = 'Sistem
Informasi') AND gaji > 5000000;
ERROR 1054 (42S22): Unknown column 'departemen' in 'where clause'
```

10. SELECT dengan UNION, INTERSECT dan EXCEPT untuk menampilkan data dengan operasi himpunan yang melibatkan lebih dari satu tabel

Format

UNION

Tanpa duplikasi

```
SELECT column_name(s) FROM table1
UNION
SELECT column_name(s) FROM table2;
```

Dengan duplikasi

```
SELECT column_name(s) FROM table1
UNION ALL
SELECT column_name(s) FROM table2;
```

INTERSECT

Tanpa duplikasi

```
SELECT column_name(s) FROM table1
INTERSECT
SELECT column_name(s) FROM table2;
```

Dengan duplikasi

```
SELECT column_name(s) FROM table1
INTERSECT ALL
SELECT column_name(s) FROM table2;
```

EXCEPT

Tanpa duplikasi

```
SELECT column_name(s) FROM table1
EXCEPT
SELECT column_name(s) FROM table2;
```

Dengan duplikasi

```
SELECT column_name(s) FROM table1
EXCEPT ALL
SELECT column_name(s) FROM table2;
```

```
MariaDB [jadwal_perkuliahan]> SELECT nama_dosen FROM dosen
-> UNION
-> SELECT nama_dosen FROM dosen_lain;
ERROR 1146 (42S02): Table 'jadwal_perkuliahan.dosen_lain' doesn't exist
MariaDB [jadwal_perkuliahan]> SELECT nama_dosen FROM dosen
-> INTERSECT
-> SELECT nama_dosen FROM dosen_lain;
ERROR 1146 (42S02): Table 'jadwal_perkuliahan.dosen_lain' doesn't exist
MariaDB [jadwal_perkuliahan]> SELECT nama_dosen FROM dosen
-> EXCEPT
-> SELECT nama_dosen FROM dosen_lain;
ERROR 1146 (42S02): Table 'jadwal_perkuliahan.dosen_lain' doesn't exist
MariaDB [jadwal_perkuliahan]> SELECT column_name(s) FROM table1
-> EXCEPT ALL
-> SELECT column_name(s) FROM table2;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '(s) FROM table1
EXCEPT ALL
SELECT column_name(s) FROM table2' at line 1
MariaDB [jadwal_perkuliahan]>
```

SUB-QUERY

Sub-query adalah adanya query di dalam query lain. Sub-query juga disebut dengan perintah SELECT bersarang (nested SELECT). Retrieval data dalam sub-query dilakukan oleh SELECT untuk menemukan data pada SELECT utama. Klausa GROUP BY dan HAVING juga dapat digunakan untuk retrieval data pada sub-query. Sub-query biasanya digunakan untuk test keanggotaan himpunan atau perbandingan himpunan dan kardinalitas.

Format

Untuk test keanggotaan

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (SELECT STATEMENT);
```

atau

```
SELECT column_name(s)
FROM table_name
GROUP BY column1, column2, ...;
HAVING column_name IN (SELECT STATEMENT);
```

Untuk perbandingan himpunan
SELECT *column_name(s)*
FROM *table_name operator*
(SELECT STATEMENT);

FUNGSI AGREGASI

Fungsi agregasi akan melakukan perhitungan kepada sekumpulan nilai dan menghasilkan suatu nilai tunggal.

Jenis:

1. **AVG** untuk menghitung rata-rata

Format

```
SELECT AVG(column_name)  
FROM table_name  
WHERE condition;
```


2. **MAX** untuk mencari nilai tertinggi
Format

```
SELECT MAX(column_name)
FROM table_name
WHERE condition;
```
3. **MIN** untuk mencari nilai terendah
Format

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```
4. **SUM** untuk menjumlahkan sekumpulan nilai
Format

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```
5. **COUNT()** untuk menunjukkan jumlah baris pada kolom yang diinginkan
Format

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

Praktikum – Bagian 1: Membuat Database untuk Percobaan

Langkah	Keterangan
1	Studi kasus yang digunakan sama dengan jobsheet DML dengan skema/model relasional/EER diagram dari database berikut.
	<pre> graph TD subgraph "jadwal_perkuliahan_prodi" kprodi[kode_prodi : varchar(3)] nprodi[nama_prodi : varchar(100)] end subgraph "jadwal_perkuliahan_kelas" kkelas[kode_kelas : varchar(10)] kprodi --> kkelas kprodi --> kkelas kprodi --> kkelas kprodi --> kkelas kprodi --> kkelas end subgraph "jadwal_perkuliahan_ruang" kruang[kode_ruang : varchar(5)] nruang[nama_ruang : varchar(20)] druang[deskripsi_ruang : varchar(100)] end subgraph "jadwal_perkuliahan_hari" khari[kode_hari : varchar(3)] nhari[nama_hari : varchar(10)] end subgraph "jadwal_perkuliahan_mahasiswa" nmah[nim : int(10)] nmahasiswa[nama_mahasiswa : varchar(100)] mkelas[kode_kelas : varchar(10)] end subgraph "jadwal_perkuliahan_jadwal" kjadwal[kode_jadwal : int(10)] mkelas --> kjadwal kdosen[kode_dosen : varchar(4)] kmk[kode_mk : varchar(5)] kruang --> kjadwal khari --> kjadwal jpmulai[jp_mulai : int(3)] jpselesai[jp_selesai : int(3)] end subgraph "jadwal_perkuliahan_dosen" kdosen ndosen[nama_dosen : varchar(100)] end subgraph "jadwal_perkuliahan_mk" kmk nmk[nama_mk : varchar(100)] end subgraph "jadwal_perkuliahan_jp" kjp[kode_jp : int(3)] jpulai[jp_mulai : time] jpselesai[jp_selesai : time] end kdosen --> kjadwal kdosen --> kjadwal kdosen --> kjadwal kdosen --> kjadwal kdosen --> kjadwal kmk --> kjadwal kmk --> kjadwal kmk --> kjadwal kmk --> kjadwal kmk --> kjadwal kruang --> kjadwal kruang --> kjadwal kruang --> kjadwal kruang --> kjadwal kruang --> kjadwal khari --> kjadwal khari --> kjadwal khari --> kjadwal khari --> kjadwal khari --> kjadwal nmahasiswa --> mkelas nmahasiswa --> mkelas nmahasiswa --> mkelas nmahasiswa --> mkelas nmahasiswa --> mkelas </pre>
2	Skema tersebut adalah skema database pada sebuah sistem informasi penjadwalan di jurusan Teknologi Informasi. Sesuai dengan percobaan pada jobsheet DML. Hapuslah tabel mahasiswa pada database jadwal_perkuliahan dengan menggunakan perintah DROP TABLE , sehingga

menghasilkan *record* keseluruhan tabel sebagai berikut (*record* tidak harus sama persis dengan tampilan ini):

```
SELECT * FROM jadwal;
```

kode_jadwal	kode_kelas	kode_dosen	kode_mk	kode_ruang	kode_hari	jp_mulai	jp_selesai
1	2021020101	D001	02001	0504	001	7	9
2	2021010103	D001	02001	0506	002	9	11
3	2021010105	D001	02001	0806	003	10	12
4	2021010102	D001	02001	0506	004	1	3
5	2021010106	D001	02001	0806	004	4	6
6	2021010101	D001	02001	0506	004	7	9
7	2021010104	D001	02001	0506	005	10	12
8	2021010206	D002	02037	0702	001	7	12
9	2021020202	D002	02036	0708	003	2	4
10	2021010205	D002	02037	0713	004	1	6
11	2021020209	D002	02025	0719	004	7	12
12	2021020301	D003	02012	0508	002	1	4
13	2021020302	D003	02012	0508	002	1	4
14	2021010201	D003	02017	0719	003	2	5
15	2021010202	D003	02017	0719	003	2	5

```
MariaDB [(none)]> show databases;
```

Database
db_akademik
db_polinema
information_schema
jadwal_perkuliahan
jual_boneka
matakuliah
mysql
pegawai
penjualan_produk
performance_schema
perusahaan
phpmyadmin
test

```
13 rows in set (0.046 sec)
```

```
MariaDB [(none)]> use jadwal_perkuliahan;
```

```
Database changed
```

```
MariaDB [jadwal_perkuliahan]> SELECT * FROM jadwal;
```

```
SELECT COLUMN_NAME(S) FROM TABLES at Line 1
```

kode_jadwal	kode_kelas	kode_dosen	kode_mk	kode_ruang	kode_hari	jp_mulai	jp_selesai
1	2021020101	D001	02001	0504	001	7	9
2	2021010103	D001	02001	0506	002	9	11
3	2021010105	D001	02001	0806	003	10	12
4	2021010102	D001	02001	0506	004	1	3
5	2021010106	D001	02001	0806	004	4	6
6	2021010101	D001	02001	0506	004	7	9
7	2021010104	D001	02001	0506	005	10	12
8	2021010206	D002	02037	0702	001	7	12
9	2021020202	D002	02036	0708	003	2	4
10	2021010205	D002	02037	0713	004	1	6
11	2021020209	D002	02025	0719	004	7	12
12	2021020301	D003	02012	0508	002	1	4
13	2021020302	D003	02012	0508	002	1	4
14	2021010201	D003	02017	0719	003	2	5
15	2021010202	D003	02017	0719	003	2	5

```
SELECT * FROM prodi;
```

kode_prodi	nama_prodi
001	D3 Manajemen Informatika
002	D4 Teknik Informatika

```
MariaDB [jadwal_perkuliahan]> SELECT * FROM prodi;  
+-----+-----+  
| kode_prodi | nama_prodi |  
+-----+-----+  
| 001        | D3 Manajemen Informatika |  
| 002        | D4 Teknik Informatika    |  
+-----+-----+  
2 rows in set (0.001 sec)
```

```
SELECT * FROM kelas;
```

kode_kelas	kode_prodi	nama_kelas
abc Filter...	abc Filter...	abc Filter...
2021010101	001	MI-
2021010102	001	MI-
2021010103	001	MI-
2021010104	001	MI-
2021010105	001	MI-
2021010106	001	MI-
2021010107	001	MI-
2021010201	001	MI-
2021010202	001	MI-
2021010203	001	MI-
2021010204	001	MI-
2021010205	001	MI-
2021010206	001	MI-
2021010301	001	MI-

```

MariaDB [jadwal_perkuliahahan]> SELECT * FROM kelas;
+-----+-----+-----+
| kode_kelas | kode_prodi | nama_kelas |
+-----+-----+-----+
| 2021010101 | 001        | MI-1A      |
| 2021010102 | 001        | MI-1B      |
| 2021010103 | 001        | MI-1C      |
| 2021010104 | 001        | MI-1D      |
| 2021010105 | 001        | MI-1E      |
| 2021010106 | 001        | MI-1F      |
| 2021010107 | 001        | MI-1H      |
| 2021010201 | 001        | MI-2A      |
| 2021010202 | 001        | MI-2B      |
| 2021010203 | 001        | MI-2C      |
| 2021010204 | 001        | MI-2D      |
| 2021010205 | 001        | MI-2E      |
| 2021010206 | 001        | MI-2F      |
| 2021010301 | 001        | MI-3A      |

```

```
SELECT * FROM ruang;
```

kode_ruang	nama_ruang	deskripsi_ruang
abc Filter...	abc Filter...	abc Filter...
0501	RT01	Ruang Teori 1
0502	RT02	Ruang Teori 2
0503	RT03	Ruang Teori 3
0504	RT04	Ruang Teori 4
0505	RT05	Ruang Teori 5
0506	RT06	Ruang Teori 6
0507	RT07	Ruang Teori 7
0508	LPY1	Laboratorium Proyek 1
0615	LSI1	Laboratorium Sistem I...
0617	LSI2	Laboratorium Sistem I...
0618	LSI3	Laboratorium Sistem I...
0619	LPY2	Laboratorium Proyek 2
0620	LPY3	Laboratorium Proyek 3
0701	LPR1	Laboratorium Pemrogr...
0702	LPR2	Laboratorium Pemrogr...
0703	LPR3	Laboratorium Pemrogr...
0704	LPR4	Laboratorium Pemrogr...
0705	LPR5	Laboratorium Pemrogr...
0706	LPR6	Laboratorium Pemrogr...
0707	LKJ1	Laboratorium Keaman...
0708	LPR7	Laboratorium Pemrogr...
0713	LKJ2	Laboratorium Keaman...
0714	LPR8	Laboratorium Pemrogr...
0715	LKJ3	Laboratorium Keaman...
0716	LIG1	Laboratorium Visi Ko...

```
MariaDB [jadwal_perkuliahan]> SELECT * FROM ruang;
```

kode_ruang	nama_ruang	deskripsi_ruang
0501	RT01	Ruang Teori 1
0502	RT02	Ruang Teori 2
0503	RT03	Ruang Teori 3
0504	RT04	Ruang Teori 4
0505	RT05	Ruang Teori 5
0506	RT06	Ruang Teori 6
0507	RT07	Ruang Teori 7
0508	LPY1	Laboratorium Proyek 1
0615	LSI1	Laboratorium Sistem Informasi 1
0617	LSI2	Laboratorium Sistem Informasi 2
0618	LSI3	Laboratorium Sistem Informasi 3
0619	LPY2	Laboratorium Proyek 2
0620	LPY3	Laboratorium Proyek 3
0701	LPR1	Laboratorium Pemrograman 1
0702	LPR2	Laboratorium Pemrograman 2
0703	LPR3	Laboratorium Pemrograman 3
0704	LPR4	Laboratorium Pemrograman 4
0705	LPR5	Laboratorium Pemrograman 5
0706	LPR6	Laboratorium Pemrograman 6
0707	LKJ1	Laboratorium Keamanan Jaringan 1
0708	LPR7	Laboratorium Pemrograman 7
0713	LKJ2	Laboratorium Keamanan Jaringan 2
0714	LPR8	Laboratorium Pemrograman 8
0715	LKJ3	Laboratorium Keamanan Jaringan 3
0716	LIG1	Laboratorium Visi Komputer 1

```
SELECT * FROM hari;
```

kode_hari	nama_hari
001	Senin
002	Selasa
003	Rabu
004	Kamis
005	Jumat
006	Sabtu
007	Minggu


```
MariaDB [jadwal_perkuliahan]> SELECT * FROM hari;
```

kode_hari	nama_hari
001	Senin
002	Selasa
003	Rabu
004	Kamis
005	Jumat
006	Sabtu
007	Minggu

7 rows in set (0.001 sec)

```
SELECT * FROM dosen;
```

kode_dosen	nama_dosen
abc Filter...	abc Filter...
D001	Abdul Chalim, SAg., MPd.I
D002	Ade Ismail
D003	Agung Nugroho Pramudhita ST., MT.
D004	Ahmadi Yuli Ananta ST., MM.
D005	Ane Fany Novitasari, SH.MKn.
D006	Annisa Puspa Kirana MKom.
D007	Annisa Taufika Firdausi ST., MT.
D008	Anugrah Nur Rahmanto SSn., MDs.
D009	Ariadi Retno Ririd SKom., MKom.
D010	Arie Rachmad Syulistyo SKom., MKom.
D011	Arief Prasetyo SKom., MKom.
D012	Arwin Sumari ST., MT., DR.
D013	Atiqah Nurul Asri SPd., MPd.
D014	Bagas Satya Dian Nugraha, ST., MT.
D015	Banni Satria Andoko, S. Kom., M.MSI
D016	Budi Harijanto ST., MMKom.
D017	Cahya Rahmad ST., MKom. DR.Eng
D018	Candra Bella Vista SKom., MT.

```

MariaDB [jadwal_perkuliahan]> select * from dosen;
+-----+-----+
| kode_dosen | nama_dosen |
+-----+-----+
| D001      | Abdul Chalim, SAg., MPd.I |
| D002      | Ade Ismail |
| D003      | Agung Nugroho Pramudhita ST., MT. |
| D004      | Ahmadi Yuli Ananta ST., MM. |
| D005      | Ane Fany Novitasari, SH.MKn. |
| D006      | Annisa Puspa Kirana MKom. |
| D007      | Annisa Taufika Firdausi ST., MT. |
| D008      | Anugrah Nur Rahmanto SSn., MDs. |
| D009      | Ariadi Retno Ririd SKom., MKom. |
| D010      | Arie Rachmad Syulistyo SKom., MKom. |
| D011      | Arief Prasetyo SKom., MKom. |
| D012      | Arwin Sumari ST., MT., DR. |
| D013      | Atiqah Nurul Asri SPd., MPd. |
| D014      | Bagas Satya Dian Nugraha, ST., MT. |
| D015      | Banni Satria Andoko, S. Kom., M.MSI |
| D016      | Budi Harijanto ST., MMKom. |
| D017      | Cahya Rahmad ST., MKom. DR.Eng |
| D018      | Candra Bella Vista SKom., MT. |
| D019      | Candrasena Setiadi ST., MMT. |
| D020      | Deasy Sandhya Elya Ikawati SSi., MSi. |
| D021      | Deddy Kusbianto PA Ir. MMKom. |
| D022      | Dhebys Suryani SKom. MT. |
| D023      | Dian Hanifudin Subhi SKom., MT. |
| D024      | Dika Rizky Yunianto SKom., MKom. |
| D025      | Dimas Wahyu Wibowo ST., MT. |

```

```
SELECT * FROM mk;
```

kode_mk	nama_mk
abc Filter...	abc Filter...
02001	Agama
02002	Aljabar Linier
02003	Algoritma dan Struktur Data
02004	Aljabar Linier
02005	Analisis Dan Desain Berorientasi Objek
02006	Bahasa Indonesia
02007	Bahasa Inggris
02008	Bahasa Inggris 2
02009	Bahasa Inggris Persiapan Kerja
02010	Basis Data
02011	Desain Pemrograman Web
02012	Digital Entrepreneurship
02013	E-Business
02014	Etika Profesi Bidang TI
02015	Internet Of Things
02016	Kewarganegaraan
02017	Komputasi Multimedia
02018	Machine Learning
02019	Manajemen Jaringan Komputer

```
MariaDB [jadwal_perkuliahan]> SELECT * FROM mk;
```

kode_mk	nama_mk
02001	Agama
02002	Aljabar Linier
02003	Algoritma dan Struktur Data
02004	Aljabar Linier
02005	Analisis Dan Desain Berorientasi Objek
02006	Bahasa Indonesia
02007	Bahasa Inggris
02008	Bahasa Inggris 2
02009	Bahasa Inggris Persiapan Kerja
02010	Basis Data
02011	Desain Pemrograman Web
02012	Digital Entrepreneurship
02013	E-Business
02014	Etika Profesi Bidang TI
02015	Internet Of Things
02016	Kewarganegaraan
02017	Komputasi Multimedia
02018	Machine Learning
02019	Manajemen Jaringan Komputer

```
SELECT * FROM jp;
```

kode_jp	jp_mulai	jp_selesai
abc Filter...	abc Filter...	abc Filter...
1	07:00:00	07:50:00
2	07:50:00	08:40:00
3	08:40:00	09:30:00
4	09:40:00	10:30:00
5	10:30:00	11:20:00
6	11:20:00	12:10:00
7	12:50:00	13:40:00
8	13:40:00	14:30:00
9	14:30:00	15:20:00
10	15:30:00	15:30:00
11	16:20:00	17:10:00
12	17:10:00	18:00:00

```
MariaDB [jadwal_perkuliahan]> SELECT * FROM jp;
```

kode_jp	jp_mulai	jp_selesai
1	07:00:00	07:50:00
2	07:50:00	08:40:00
3	08:40:00	09:30:00
4	09:40:00	10:30:00
5	10:30:00	11:20:00
6	11:20:00	12:10:00
7	12:50:00	13:40:00
8	13:40:00	14:30:00
9	14:30:00	15:20:00
10	15:30:00	15:30:00
11	16:20:00	17:10:00
12	17:10:00	18:00:00

12 rows in set (0.001 sec)

3 Setelah selesai membuat database diatas, lanjutkan ke **Praktikum – Bagian 2.**

Praktikum – Bagian 2: Percobaan SELECT QUERY

Langkah	Keterangan
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.
	<pre>SELECT deskripsi_ruang FROM ruang WHERE nama_ruang = 'LKJ1';</pre>

	<pre> MariaDB [jadwal_perkuliahan]> SELECT * FROM Ruang -> WHERE nama_ruang = 'LKJ1'; +-----+-----+-----+ kode_ruang nama_ruang deskripsi_ruang +-----+-----+-----+ 0707 LKJ1 Laboratorium Keamanan Jaringan 1 +-----+-----+-----+ 1 row in set (0.003 sec) </pre> <div> <div>deskripsi_ruang</div> <div>abc Filter...</div> <div>Laboratorium Keamanan Jaringan 1</div> </div>	
2	<p>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.</p>	
	<pre> SELECT DISTINCT kode_hari FROM jadwal; </pre>	

	<div><div><div>kode_hari</div><div>abc Filter...</div><div>001</div><div>002</div><div>003</div><div>004</div><div>005</div></div><div>MariaDB [jadwal_perkuliahan]> SELECT DISTINCT kode_hari FROM jadwal; +-----+ kode_hari +-----+ 001 002 003 004 005 +-----+ 5 rows in set (0.001 sec)</div></div>
3	Untuk menampilkan data yang spesifik digunakan statement SELECT dengan IN . Eksekusi SQL berikut untuk menampilkan kolom kode_ruang, nama_ruang dan deskripsi_ruang dari tabel ruang yang memiliki nama_ruang RT01 atau RT10.
	<pre>SELECT * FROM ruang WHERE nama_ruang IN ('RT01','RT10');</pre>
	<div><div><div>kode_ruang</div><div>abc Filter...</div><div>0501</div><div>0806</div></div><div><div>nama_ruang</div><div>abc Filter...</div><div>RT01</div><div>RT10</div></div><div><div>deskripsi_ruang</div><div>abc Filter...</div><div>Ruang Teori 1</div><div>Ruang Teori 10</div></div></div> <div>MariaDB [jadwal_perkuliahan]> SELECT kode_ruang, nama_ruang, deskripsi_ruang -> FROM ruang -> WHERE nama_ruang IN ('RT01', 'RT10'); +-----+ +-----+ +-----+ kode_ruang nama_ruang deskripsi_ruang +-----+ +-----+ +-----+ 0501 RT01 Ruang Teori 1 0806 RT10 Ruang Teori 10 +-----+ +-----+ +-----+ 2 rows in set (0.001 sec)</div>
4	Untuk menampilkan data pada jarak (range) tertentu digunakan statement SELECT dengan BETWEEN . Eksekusi SQL berikut untuk menampilkan kolom kode_ruang dan nama_ruang dan deskripsi_ruang dari tabel ruang yang memiliki kode_ruang antara 0501 dan 0508.
	<pre>SELECT * FROM ruang WHERE kode_ruang BETWEEN '0501' AND '0508';</pre>

kode_ruang	nama_ruang	deskripsi_ruang
abc Filter...	abc Filter...	abc Filter...
0501	RT01	Ruang Teori 1
0502	RT02	Ruang Teori 2
0503	RT03	Ruang Teori 3
0504	RT04	Ruang Teori 4
0505	RT05	Ruang Teori 5
0506	RT06	Ruang Teori 6
0507	RT07	Ruang Teori 7
0508	LPY1	Laboratorium Proyek 1

```
MariaDB [jadwal_perkuliah]> SELECT kode_ruang, nama_ruang, deskripsi_ruang
-> FROM ruang
-> WHERE kode_ruang BETWEEN '0501' AND '0508';
```

kode_ruang	nama_ruang	deskripsi_ruang
0501	RT01	Ruang Teori 1
0502	RT02	Ruang Teori 2
0503	RT03	Ruang Teori 3
0504	RT04	Ruang Teori 4
0505	RT05	Ruang Teori 5
0506	RT06	Ruang Teori 6
0507	RT07	Ruang Teori 7
0508	LPY1	Laboratorium Proyek 1

```
8 rows in set (0.001 sec)
```

5

Untuk menampilkan data yang memiliki kemiripan dengan keyword yang diinginkan digunakan **SELECT** dengan **LIKE**. Eksekusi SQL berikut untuk menampilkan kolom kode_dosen, dan nama_dosen, pada tabel **dosen** yang memiliki nama dengan huruf awal 'E'.

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

kode_dosen	nama_dosen
abc Filter...	abc Filter...
D028	Eka Larasati Amalia, SST., MT.
D029	Ekojono, ST., M.Kom.
D030	Elok Nur Hamdana, ST., MT
D031	Erfan Rohadi, ST., MEng., PhD.

```
MariaDB [jadwal_perkuliahan]> SELECT kode_dosen, nama_dosen FROM
dosen WHERE nama_dosen LIKE 'E%';
+-----+-----+
| kode_dosen | nama_dosen |
+-----+-----+
| D028      | Eka Larasati Amalia, SST., MT. |
| D029      | Ekojono, ST., M.Kom. |
| D030      | Elok Nur Hamdana, ST., MT |
| D031      | Erfan Rohadi, ST., MEng., PhD. |
+-----+-----+
4 rows in set (0.001 sec)
```

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.

```
SELECT kode_dosen, kode_mk, kode_ruang, kode_hari
FROM jadwal
GROUP BY kode_hari;
```

kode_dosen	kode_mk	kode_ruang	kode_hari
abc Filter...	abc Filter...	abc Filter...	abc Filter...
D001	02001	0504	001
D001	02001	0506	002
D001	02001	0806	003
D001	02001	0506	004
D001	02001	0506	005

```
MariaDB [jadwal_perkuliahan]> SELECT kode_dosen, kode_mk, kode_r
uang, kode_hari FROM jadwal GROUP BY kode_hari;
+-----+-----+-----+-----+
| kode_dosen | kode_mk | kode_ruang | kode_hari |
+-----+-----+-----+-----+
| D001      | 02001  | 0504      | 001      |
| D001      | 02001  | 0506      | 002      |
| D001      | 02001  | 0806      | 003      |
| D001      | 02001  | 0506      | 004      |
| D001      | 02001  | 0506      | 005      |
+-----+-----+-----+-----+
5 rows in set (0.002 sec)
```

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.
	<pre>SELECT kode_jp, jp_mulai FROM jp ORDER BY jp_mulai;</pre>

kode_jp	jp_mulai
1	07:00:00
2	07:50:00
3	08:40:00
4	09:40:00
5	10:30:00
6	11:20:00
7	12:50:00
8	13:40:00
9	14:30:00
10	15:30:00
11	16:20:00
12	17:10:00

```
MariaDB [jadwal_perkuliahan]> SELECT kode_jp, jp_mulai
-> FROM jp
-> ORDER BY jp_mulai ASC;
+-----+-----+
| kode_jp | jp_mulai |
+-----+-----+
| 1       | 07:00:00 |
| 2       | 07:50:00 |
| 3       | 08:40:00 |
| 4       | 09:40:00 |
| 5       | 10:30:00 |
| 6       | 11:20:00 |
| 7       | 12:50:00 |
| 8       | 13:40:00 |
| 9       | 14:30:00 |
| 10      | 15:30:00 |
| 11      | 16:20:00 |
| 12      | 17:10:00 |
+-----+-----+
12 rows in set (0.001 sec)
```

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.

```
SELECT *
FROM jadwal
WHERE kode_hari = '001' AND jp_mulai=1;
```

kode_jadwal	kode_kelas	kode_dosen	kode_mk	kode_ruang	kode_hari	jp_mulai	jp_selesai
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
17	2021010106	D004	02028	0704	001	1	3
21	2021020104	D005	02016	0717	001	1	3
57	2021020204	D012	02018	0702	001	1	5
75	2021020103	D016	02038	0506	001	1	3
84	2021020302	D018	02039	0701	001	1	6
91	2021020207	D021	02005	0615	001	1	6
98	2021010205	D023	02023	0508	001	1	4
103	2021010201	D024	02037	0713	001	1	6
107	2021020305	D025	02024	0615	001	1	6
147	2021020108	D034	02008	0502	001	1	3
170	2021010101	D039	02040	0503	001	1	3
171	2021010102	D039	02040	0503	001	1	3
197	2021020208	D045	02037	0718	001	1	6
206	2021020308	D047	02022	0620	001	1	6
227	2021020206	D051	02037	0715	001	1	6
242	2021020106	D055	02003	0716	001	1	3
256	2021010203	D059	02023	0719	001	1	4
279	2021010301	D064	02006	0805	001	1	3

MariaDB [jadwal_perkuliahan]> SELECT * FROM jadwal WHERE kode_hari = '001' AND jp_mulai = 1;

kode_jadwal	kode_kelas	kode_dosen	kode_mk	kode_ruang	kode_hari	jp_mulai	jp_selesai
17	2021010106	D004	02028	0704	001	1	3
21	2021020104	D005	02016	0717	001	1	3
57	2021020204	D012	02018	0702	001	1	5
75	2021020103	D016	02038	0506	001	1	3
84	2021020302	D018	02039	0701	001	1	6
91	2021020207	D021	02005	0615	001	1	6
98	2021010205	D023	02023	0508	001	1	4
103	2021010201	D024	02037	0713	001	1	6
107	2021020305	D025	02024	0615	001	1	6
147	2021020108	D034	02008	0502	001	1	3
170	2021010101	D039	02040	0503	001	1	3
171	2021010102	D039	02040	0503	001	1	3
197	2021020208	D045	02037	0718	001	1	6
206	2021020308	D047	02022	0620	001	1	6
227	2021020206	D051	02037	0715	001	1	6
242	2021020106	D055	02003	0716	001	1	3
256	2021010203	D059	02023	0719	001	1	4
279	2021010301	D064	02006	0805	001	1	3

9	<p>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.</p>
	<pre>SELECT kode_hari FROM hari UNION SELECT kode_hari FROM jadwal;</pre>
	<div> <div> kode_hari <input type="text"/> Filter... 001 002 003 004 005 006 007 </div> <div> <pre>MariaDB [jadwal_perkuliahan]> SELECT kode_hari FROM hari UNION SELECT kode_hari FROM jadwal; +-----+ kode_hari +-----+ 001 002 003 004 005 006 007 +-----+ 7 rows in set (0.001 sec)</pre> </div> </div>
10	<p>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.</p>
	<pre>SELECT kode_hari FROM hari UNION ALL SELECT kode_hari FROM jadwal;</pre> <pre>MariaDB [jadwal_perkuliahan]> SELECT nik FROM penugasan UNION ALL SELECT nik FROM departemen; ERROR 1146 (42S02): Table 'jadwal_perkuliahan.penugasan' doesn't exist</pre>
	Silakan Capture hasil output dari query di atas
11	Setelah berhasil mengeksekusi SQL tersebut, lanjutkan ke <u>Praktikum - Bagian 3</u> .

Praktikum - Bagian 3: Percobaan SELECT Sub-Query

Langkah	Keterangan
---------	------------

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 .
	<pre>SELECT kode_dosen, kode_mk, kode_hari, jp_mulai, jp_selesai FROM jadwal WHERE jp_selesai IN (SELECT MAX(jp_selesai) FROM jadwal);</pre>

kode_dosen	kode_mk	kode_hari	jp_mulai	jp_selesai
a bc Filter...	a bc Filter...	a bc Filter...	a bc Filter...	a bc Filter...
D001	02001	003	10	12
D001	02001	005	10	12
D002	02037	001	7	12
D002	02025	004	7	12
D004	02032	001	10	12
D004	02034	002	7	12
D005	02016	001	10	12
D006	02037	004	7	12
D006	02037	005	7	12
D007	02011	004	8	12
D008	02012	005	9	12
D008	02012	005	9	12
D009	02005	003	7	12
D011	02019	004	8	12
D015	02005	002	7	12
D015	02005	003	7	12
D015	02033	004	7	12

```

MariaDB [jadwal_perkuliahan]> SELECT kode_dosen, kode_mk, kode_hari, jp_mulai, jp_selesai
-> FROM jadwal
-> WHERE jp_selesai IN (SELECT MAX(jp_selesai) FROM jadwal);
+-----+-----+-----+-----+-----+
| kode_dosen | kode_mk | kode_hari | jp_mulai | jp_selesai |
+-----+-----+-----+-----+-----+
| D001       | 02001   | 003       | 10        | 12         |
| D001       | 02001   | 005       | 10        | 12         |
| D002       | 02037   | 001       | 7         | 12         |
| D002       | 02025   | 004       | 7         | 12         |
| D004       | 02032   | 001       | 10        | 12         |
| D004       | 02034   | 002       | 7         | 12         |
| D005       | 02016   | 001       | 10        | 12         |
| D006       | 02037   | 004       | 7         | 12         |
| D006       | 02037   | 005       | 7         | 12         |
| D007       | 02011   | 004       | 8         | 12         |
| D008       | 02012   | 005       | 9         | 12         |
| D008       | 02012   | 005       | 9         | 12         |
| D009       | 02005   | 003       | 7         | 12         |
| D011       | 02019   | 004       | 8         | 12         |
| D015       | 02005   | 002       | 7         | 12         |
| D015       | 02005   | 003       | 7         | 12         |
| D015       | 02033   | 004       | 7         | 12         |
| D016       | 02033   | 005       | 7         | 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**.

```
SELECT kode_dosen, kode_mk, kode_hari, jp_mulai,jp_selesai  
FROM jadwal  
WHERE jp_selesai < ALL  
(SELECT jp_selesai FROM jadwal WHERE jp_selesai=6);
```

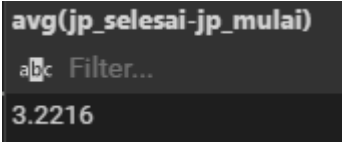
kode_dosen	kode_mk	kode_hari	jp_mulai	jp_selesai
abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...
D001	02001	004	1	3
D002	02036	003	2	4
D003	02012	002	1	4
D003	02012	002	1	4
D003	02017	003	2	5
D003	02017	003	2	5
D003	02017	005	2	5
D004	02028	001	1	3
D005	02016	001	1	3
D005	02016	002	1	3
D005	02016	004	1	3
D005	02016	005	1	3
D006	02010	002	1	3
D007	02038	003	3	5
D007	02038	003	3	5
D008	02012	001	2	5

Subquery mengembalikan lebih dari 1 baris, artinya subquery Anda mengembalikan lebih dari satu nilai untuk jp_selesai. Untuk menggunakan operator perbandingan seperti <, >, <=, atau >=, subquery harus mengembalikan tepat satu nilai.

```
MariaDB [jadwal_perkuliahan]> SELECT kode_dosen, kode_mk, kode_hari, jp_mulai, jp_selesai
-> FROM jadwal
-> WHERE jp_selesai < (SELECT jp_selesai FROM jadwal WHERE jp_selesai = 6);
ERROR 1242 (21000): Subquery returns more than 1 row
MariaDB [jadwal_perkuliahan]> SELECT kode_dosen, kode_mk, kode_hari, jp_mulai, jp_selesai
-> FROM jadwal
-> WHERE jp_selesai < 6;
```

kode_dosen	kode_mk	kode_hari	jp_mulai	jp_selesai
D001	02001	004	1	3
D002	02036	003	2	4
D003	02012	002	1	4
D003	02012	002	1	4
D003	02017	003	2	5
D003	02017	003	2	5
D003	02017	005	2	5
D004	02028	001	1	3
D005	02016	001	1	3
D005	02016	002	1	3
D005	02016	004	1	3
D005	02016	005	1	3
D006	02010	002	1	3
D007	02038	003	3	5
D007	02038	003	3	5
D008	02012	001	2	5
D008	02036	003	2	4

Praktikum - Bagian 4: Percobaan FUNGSI AGREGASI

Langkah	Keterangan
1	<p>Untuk menghitung rata-rata, digunakan tambahan statement AVG. Eksekusi SQL berikut untuk menampilkan rata-rata durasi perkuliahan dalam satu jadwal perkuliahan sesuai kolom <code>jp_mulai</code>, <code>jp_selesai</code> dari tabel jadwal.</p>
	<pre>SELECT AVG(jp_selesai-jp_mulai) FROM jadwal;</pre>
	 <pre>avg(jp_selesai-jp_mulai) 3.2216</pre> <pre>MariaDB [jadwal_perkuliahan]> SELECT AVG(jp_selesai - jp_mulai) AS rata_rata_dura si FROM jadwal; +-----+ rata_rata_durasi +-----+ 3.2216 +-----+ 1 row in set (0.001 sec)</pre>
2	<p>Untuk menghitung nilai tertinggi, digunakan tambahan statement MAX. Eksekusi SQL berikut untuk menampilkan dimulainya jam perkuliahan yang paling siang sesuai kolom <code>jp_mulai</code> dari tabel jadwal.</p>
	<pre>SELECT MAX (jp_mulai) FROM jadwal;</pre>

	<div> <div>MAX (jp_mulai)</div> <div>abc Filter...</div> <div>10</div> </div> <pre> MariaDB [jadwal_perkuliahan]> SELECT MAX(jp_mulai) AS jam_mulai_tertinggi FROM jadwal; +-----+ jam_mulai_tertinggi +-----+ 10 +-----+ 1 row in set (0.001 sec) MariaDB [jadwal_perkuliahan]> </pre>
3	<p>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.</p>
	<pre>SELECT MIN (jp_selesai) FROM jadwal;</pre>
	<div> <div>MIN (jp_selesai)</div> <div>abc Filter...</div> <div>3</div> </div> <pre> MariaDB [jadwal_perkuliahan]> SELECT MIN(jp_selesai) AS jam_selesai_terendah FROM jadwal; +-----+ jam_selesai_terendah +-----+ 3 +-----+ 1 row in set (0.001 sec) </pre>
4	<p>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.</p>
	<pre>SELECT SUM(jp_selesai-jp_mulai) FROM jadwal WHERE kode_dosen='D001';</pre>

sum(jp_selesai-jp_mulai)

abc Filter...

14

```
MariaDB [jadwal_perkuliahan]> SELECT kode_dosen, SUM(jp_selesai - jp_mulai) AS total_jam_ajar
-> FROM jadwal
-> WHERE kode_dosen = 'D001'
-> GROUP BY kode_dosen;
+-----+-----+
| kode_dosen | total_jam_ajar |
+-----+-----+
| D001      | 14             |
+-----+-----+
1 row in set (0.003 sec)
```

Coba tampilkan jadwal dosen dengan kode 'D001' dan pastikan apakah benar total jam ajar dosen tersebut sesuai dengan perintah sum dengan nilai 14 pada gambar di atas.

```
SELECT * FROM jadwal WHERE kode_dosen='D001';
```

kode_jadwal	kode_kelas	kode_dosen	kode_mk	kode_ruang	kode_hari	jp_mulai	jp_selesai
abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...
1	2021020101	D001	02001	0504	001	7	9
2	2021010103	D001	02001	0506	002	9	11
3	2021010105	D001	02001	0806	003	10	12
4	2021010102	D001	02001	0506	004	1	3
5	2021010106	D001	02001	0806	004	4	6
6	2021010101	D001	02001	0506	004	7	9
7	2021010104	D001	02001	0506	005	10	12

```
MariaDB [jadwal_perkuliahan]> SELECT * FROM jadwal WHERE kode_dosen='D001';
```

kode_jadwal	kode_kelas	kode_dosen	kode_mk	kode_ruang	kode_hari	jp_mulai	jp_selesai
1	2021020101	D001	02001	0504	001	7	9
2	2021010103	D001	02001	0506	002	9	11
3	2021010105	D001	02001	0806	003	10	12
4	2021010102	D001	02001	0506	004	1	3
5	2021010106	D001	02001	0806	004	4	6
6	2021010101	D001	02001	0506	004	7	9
7	2021010104	D001	02001	0506	005	10	12

7 rows in set (0.001 sec)

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**.

```
SELECT COUNT (kode_ruang) FROM ruang;
```

COUNT (kode_ruang)

abc Filter...

36

	<pre> MariaDB [jadwal_perkuliahan]> SELECT COUNT(*) AS total_baris -> FROM ruang; +-----+ total_baris +-----+ 36 +-----+ 1 row in set (0.001 sec) </pre>	
6	Lanjutkan ke bagian <u>Tugas!</u>	

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!

```
MariaDB [jadwal_perkuliahan]> SELECT *
-> FROM jadwal
-> WHERE kode_hari = (SELECT kode_hari FROM hari WHERE nama_hari = 'selasa')
-> AND jp_mulai = '10:30';
```

kode_jadwal	kode_kelas	kode_dosen	kode_mk	kode_ruang	kode_hari	jp_mulai	jp_selesai
141	2021020403	D032	02009	0502	002	10	12
161	2021010104	D037	02016	0504	002	10	12
315	2021020109	D074	02038	0805	002	10	12
368	2021010203	D085	02006	0806	002	10	12

4 rows in set, 1 warning (0.001 sec)

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!

```
MariaDB [jadwal_perkuliahan]> SELECT COUNT(DISTINCT kode_dosen) AS jumlah_dosen
-> FROM jadwal
-> WHERE kode_hari = (SELECT kode_hari FROM hari WHERE nama_hari = 'selasa');
```

jumlah_dosen
51

1 row in set (0.001 sec)

```
MariaDB [jadwal_perkuliahan]> SELECT DISTINCT kode_dosen
-> FROM jadwal
-> WHERE kode_hari = (SELECT kode_hari FROM hari WHERE nama_hari = 'selasa');
```

kode_dosen
D001
D003
D004
D005
D006
D007
D009
D010
D011
D012
D013
D015
D021
D025
D027
D028
D030
D031
D032
D034
D035

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.

```
MariaDB [jadwal_perkuliahan]> SELECT kode_dosen
-> FROM jadwal
-> WHERE kode_hari = (SELECT kode_hari FROM hari WHERE nama_hari = 'selasa')
-> INTERSECT
-> SELECT kode_dosen
-> FROM dosen
-> WHERE nama_dosen LIKE 'M%';
```

kode_dosen
D046
D047
D048
D049
D052
D054

6 rows in set (0.001 sec)

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

```
MariaDB [jadwal_perkuliahan]> SELECT *, kode_ruang
-> FROM jadwal
-> WHERE kode_hari = (SELECT kode_hari FROM hari WHERE nama_hari = 'Tentukan
Hari')
-> ORDER BY kode_ruang ASC;
Empty set (0.002 sec)
```

-- Selamat Mengerjakan --

Daftar Pustaka

- Dwi Puspitasari, S.Kom, “**Buku Ajar Dasar Basis Data**”, *Program Studi Manajemen Informatika Politeknik Negeri Malang*, 2012.
- <http://www.tutorialride.com>
- <https://www.w3schools.com/>