

## Modul 2

# SQL

Oleh: Akhsin Nurlayli, S.Pd., M.Eng.

---



## Agenda

Pada modul ini kita akan mempelajari:

- Pengenalan fungsi dari komponen-komponen SQL
- Penggunaan komponen-komponen SQL dan cara pemakaiannya
- DDL (*Data Definition Language*)
- DML (*Data Manipulation Language*)
- DCL (*Data Control Language*)

## Daftar Isi

<b>Agenda .....</b>	<b>1</b>
<b>Daftar Isi .....</b>	<b>2</b>
<b>Dasar Teori .....</b>	<b>3</b>
A. SQL.....	3
B. DDL .....	3
C. DML .....	4
D. DCL.....	4
<b>Latihan.....</b>	<b>5</b>
A. INSERTING .....	5
B. UPDATE.....	6
C. DELETE .....	6
D. SELECTING .....	7
<b>Tugas Praktikum .....</b>	<b>9</b>

## Dasar Teori

### A. SQL

Kepanjangan dari SQL adalah *Stuctured Query Language*, sebuah Bahasa yang digunakan untuk *request* informasi dari bais data. Bahasa ini sudah disediakan oleh DBMS. Bentuk konstruksi dari SQL adalah kombinasi *relational-algebra* dan *relational-calculus*.

### B. DDL

DDL (*Data Definition Language*) merupakan kumpulan perintah SQL yang digunakan untuk membuat, mengubah dan menghapus struktur dan definisi metadata dari objek-objek database.

#### 1. Creating Table

Digunakan untuk membuat tabel baru.

```
CREATE TABLE <nama_tabel>
(<nama_kolom_1> <tiipe_data>(<panjang_data>)
[UNIQUE] [NOT NULL] [PRIMARY KEY] [DEFAULT <nilai_default>]
[referential_constraint_definition] [INDEX],
<nama_kolom_2> <tiipe_data>(<panjang_data>)
[UNIQUE] [NOT NULL] [PRIMARY KEY] [DEFAULT <nilai_default>]
[referential_constraint_definition] [INDEX], ....);
```

#### 2. Altering

- Mengubah nama kolom

```
ALTER TABLE <nama_tabel> CHANGE
<nama_kolom_lama> <nama_kolom_baru> <tiipe_data>(<panjang_data>)
[FIRST | AFTER <nama_kolom>];
```

- Menambah dan menghapus kolom

```
ALTER TABLE <nama_tabel>
ADD <nama_kolom_baru> <tiipe_data>(<panjang_data>) [FIRST | AFTER
<nama_kolom>];
```

- Mengubah nama tabel

```
ALTER TABLE <nama_tabel> RENAME TO <nama_tabel_baru>;
```

- Menghapus *primary key*

```
ALTER TABLE <nama_tabel> DROP PRIMARY KEY;
```

- Menambah dan menghapus batasan (*constraint*)

```
ALTER TABLE <nama_tabel> ADD CONSTRAINT <nama_constraint> UNIQUE
<nama_kolom>;
```

### 3. Dropping

Untuk menghapus tabel.

```
DROP TABLE <nama_table>;
```

## C. DML

*Data Manipulation Language* atau yang biasa dikenal dengan DML adalah kumpulan perintah SQL yang digunakan untuk proses pengolahan isi data di dalam table seperti memasukkan, merubah dan menghapus isi data, dan tidak terkait dengan perubahan struktur dan definisi tipe data dari objek database.

### - INSERT

Untuk menambahkan data.

```
INSERT INTO <nama_tabel> [(<nama_kolom_1>, <nama_kolom_2>, ...)] VALUES  
(<nilai_kolom_1>, <nilai_kolom_2>, ...);
```

### - UPDATE

Untuk memodifikasi data.

```
UPDATE <nama_tabel>  
SET <nama_kolom_1 = nilai_kolom_1>,  
<nama_kolom_2 = nilai_kolom_2>, ...,  
<nama_kolom_N = nilai_kolom_N> [WHERE <kondisi>];
```

### - DELETE

Untuk menghapus data.

```
DELETE FROM <nama_tabel>  
WHERE <kondisi>;
```

### - SELECT

Untuk pengambilan data atau menampilkan data tertentu.

```
SELECT <nama_kolom_1, nama_kolom_2, ..., nama_kolom_N> FROM <nama_tabel>  
WHERE <kondisi>;
```

## D. DCL

DCL adalah sub bahasa SQL yang berfungsi untuk melakukan pengontrolan data dan server databasenya, seperti manipulasi user dan hak akses (*priviledges*). Yang termasuk perintah dalam DCL ada dua, yaitu **GRANT** dan **REVOKE**.

GRANT digunakan untuk memberikan hak akses oleh admin ke salah satu user atau pengguna. Hak akses tersebut bisa berupa hak membuat (CREATE), mengambil data (SELECT), menghapus data (DELETE), mengubah data (UPDATE), dan hak khusus lainnya yang berhubungan dengan sistem database. REVOKE digunakan untuk mencabut hak akses yang telah diberikan kepada user. Dalam ini merupakan kebalikan dari perintah GRANT. Sehingga untuk administrator *database*, perintah-perintahnya adalah: **CREATE DATABASE, CREATE USER, DROP USER, GRANT PRIVILEGE, REVOKE PRIVILEGE**

## Latihan

### A. INSERTING

Coba lakukan penambahan data pada *database* mahasiswa, tabel mahasiswa.

```
C:\ Command Prompt - mysql -u root
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use mahasiswa
Database changed
MariaDB [mahasiswa]> show tables
-> ;
+-----+
| Tables_in_mahasiswa |
+-----+
| mahasiswa            |
| matakuliah           |
+-----+
2 rows in set (0.001 sec)

MariaDB [mahasiswa]> desc mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| no         | int(12)       | NO   |     | NULL    |       |
| nim        | char(10)      | NO   | PRI | NULL    |       |
| nama       | varchar(60)   | NO   |     | NULL    |       |
| tanggal_lahir | varchar(50)   | YES  |     | NULL    |       |
| jenis_kelamin | char(1)       | NO   |     | NULL    |       |
| alamat     | varchar(120)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.034 sec)

MariaDB [mahasiswa]>
```

Selain menuliskan perintah sql pada CMD, bisa dengan menuliskan perintahnya pada *file* dengan ekstensi .sql. Misalnya ketikkan di sublime, notepad++, atau semacamnya, simpan *file* dengan ekstensi .sql.

C:\tes\insert1.sql

File Edit Selection Find View Goto Tools Project Preferences Help

```
insert1.sql
1 INSERT INTO mahasiswa (no, nim, nama, tanggal_lahir, jenis_kelamin, alamat)
2 VALUES('1','1105','Ardhito','2 November 1999','Laki-laki','Giwangan');
3
```

Untuk memanggil *file* .sql yang telah dibuat, silakan mengetikkan `source <tempat menyimpan file/namafilename.sql>`. Sehingga data berhasil disimpan. Lakukan hal serupa jika ingin mengetikkan perintah pada *file* .sql agar tersimpan perintah yang pernah digunakan.

```

MariaDB [mahasiswa]> source c:\tes\insert1.sql
Query OK, 1 row affected, 1 warning (0.004 sec)

MariaDB [mahasiswa]> select*from mahasiswa;
+-----+-----+-----+-----+-----+-----+
| no | nim | nama | tanggal_lahir | jenis_kelamin | alamat |
+-----+-----+-----+-----+-----+-----+
| 1 | 1105 | Ardhito | 2 November 1999 | L | Giwangan |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

MariaDB [mahasiswa]>

```

## B. UPDATE

Mengubah isi data pada kolom tertentu pada *field* tertentu. Misalnya mengubah nama mahasiswa pada nomor 1.

```

1 UPDATE mahasiswa
2 SET nama='Pramono'
3 WHERE no='1';
4

```

Setelah memanggil *file* .sql, coba tampilkan tabel untuk mengetahui data yang sudah berhasil ditambahkan.

```

MariaDB [mahasiswa]> source c:\tes\update1.sql
Query OK, 1 row affected (0.004 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [mahasiswa]> select*from mahasiswa;
+-----+-----+-----+-----+-----+-----+
| no | nim | nama | tanggal_lahir | jenis_kelamin | alamat |
+-----+-----+-----+-----+-----+-----+
| 1 | 1105 | Pramono | 2 November 1999 | L | Giwangan |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

MariaDB [mahasiswa]>

```

## C. DELETE

Menghapus data tertentu. Misalnya menghapus data mahasiswa yang memiliki no. 1.

```

1 DELETE FROM mahasiswa
2 WHERE no='1';

```

Setelah memanggil *file* .sql, coba tampilkan tabel untuk mengetahui data yang sudah berhasil dihapus.

```
MariaDB [mahasiswa]> source c:\tes\delete1.sql
Query OK, 1 row affected (0.004 sec)

MariaDB [mahasiswa]> select*from mahasiswa;
Empty set (0.001 sec)

MariaDB [mahasiswa]>
```

#### D. SELECTING

Coba lakukan penambahan data, bisa lebih banyak dari yang dicontohkan di bawah ini.

```
insert1.sql  insert2.sql  update1.sql  delete1.sql
1 INSERT INTO mahasiswa (no, nim, nama, tanggal_lahir, jenis_kelamin, alamat)
2 VALUES('1','1105','Ardhito','2 November 1999','L','Giwangan'),
3 ('2','1106','Pramono','10 Oktober 1998','L','Ngemplak'),
4 ('3','1107','Pamungkas','8 September 1999','L','Gejayan'),
5 ('4','1109','Julia','19 Juni 1998','P','Gamping');
```

Setelah memanggil *file* .sql, coba tampilkan tabel untuk mengetahui data yang sudah berhasil ditambahkan.

```
MariaDB [mahasiswa]> select*from mahasiswa;
Empty set (0.000 sec)

MariaDB [mahasiswa]> source c:\tes\insert2.sql
Query OK, 4 rows affected (0.004 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [mahasiswa]> select*from mahasiswa;
+----+-----+-----+-----+-----+-----+
| no | nim  | nama    | tanggal_lahir | jenis_kelamin | alamat    |
+----+-----+-----+-----+-----+-----+
| 1  | 1105 | Ardhito | 2 November 1999 | L             | Giwangan  |
| 2  | 1106 | Pramono | 10 Oktober 1998 | L             | Ngemplak  |
| 3  | 1107 | Pamungkas | 8 September 1999 | L             | Gejayan   |
| 4  | 1109 | Julia   | 19 Juni 1998    | P             | Gamping   |
+----+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [mahasiswa]>
```

Selain berfungsi untuk menampilkan semua data pada sebuah tabel, fungsi SELECT dapat digunakan dalam berbagai kriteria, antara lain:

1. Menampilkan semua data pada tabel

```
SELECT * FROM MAHASISWA;
```

Untuk yang belum tahu, arti dari \* (asterik) adalah ALL yang artinya menampilkan semua kolom.

2. Menampilkan data pada kriteria tertentu

Misalnya ingin menampilkan data mahasiswa yang memiliki berjenis kelamin perempuan saja, maka:

```
MariaDB [mahasiswa]> SELECT * FROM mahasiswa WHERE jenis_kelamin='P';
+----+-----+-----+-----+-----+-----+
| no | nim | nama | tanggal_lahir | jenis_kelamin | alamat |
+----+-----+-----+-----+-----+-----+
| 4 | 1109 | Julia | 19 Juni 1998 | P | Gamping |
+----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

### 3. Menampilkan data dengan operator LIKE

Pada tabel **matakuliah** yang sudah dibuat saat mengerjakan Tugas Praktikum pada Modul 1, tambahkan data (lebih banyak data lebih baik) hingga memunculkan data seperti berikut ini:

```
MariaDB [mahasiswa]> SELECT * FROM MATAKULIAH;
+-----+-----+-----+-----+
| kode_matakuliah | nama_matakuliah | sks | jp |
+-----+-----+-----+-----+
| 12345 | Basis Data | 2 | 4 |
| 12346 | Praktikum Basis Data | 2 | 4 |
| 12348 | Aljabar Linier | 2 | 4 |
+-----+-----+-----+-----+
3 rows in set (0.001 sec)
```

Kemudian ketikkan seperti di bawah ini:

```
MariaDB [mahasiswa]> SELECT * FROM matakuliah WHERE nama_matakuliah LIKE "Basis%";
+-----+-----+-----+-----+
| kode_matakuliah | nama_matakuliah | sks | jp |
+-----+-----+-----+-----+
| 12345 | Basis Data | 2 | 4 |
+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [mahasiswa]> SELECT * FROM matakuliah WHERE nama_matakuliah LIKE "%Basis%";
+-----+-----+-----+-----+
| kode_matakuliah | nama_matakuliah | sks | jp |
+-----+-----+-----+-----+
| 12345 | Basis Data | 2 | 4 |
| 12346 | Praktikum Basis Data | 2 | 4 |
+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

**Apa kegunaan LIKE dan tanda % pada kata Basis? Coba Jelaskan!**

### 4. Menampilkan data dengan pengurutan

Dengan menggunakan perintah **ORDER BY**, kita dapat mengurutkan data. **Coba jelaskan bagaimana cara menggunakan perintah pengurutan menggunakan ORDER BY! Dan berikan contoh penggunaan pada tabel yang telah Anda miliki!**



## Tugas Praktikum

1. Buat tabel baru dengan nama **tabel\_mahasiswa\_2**! Screenshot perintah pembuatan tabel baru dengan menggunakan *file* .sql. Dengan *field* sebagai berikut:

Nama <i>Field</i>	Tipe Data	Keterangan
Nim	Varchar(50)	Primary Key
Nama	Varchar(50)	Nama Mahasiswa
Jenis_kelamin	Varchar(50)	Jenis Kelamin Mahasiswa
Tempat_lahir	Varchar(50)	Tempat Lahir Mahasiswa
Tanggal_lahir	Varchar(50)	Tanggal Lahir Mahasiswa
Alamat	Varchar(50)	Alamat Mahasiswa

Kemudian tambahkan data pada tabel tersebut dengan data sebagai berikut:

nim	nama	jenis_kelamin	tempat_lahir	tanggal_lahir	alamat
140533601613	Cintya	Perempuan	Batam	1998-09-07	Jalan Apel
140533606464	Lugas	Laki-Laki	Batu	1995-04-19	Jalan Mangga
160533608100	Dera	laki-laki	Surabaya	1997-05-17	Jalan Melon
160533608101	Budi	Laki-Laki	Medan	1998-10-29	Jalan Apel
160533608135	Fahmi	Laki-laki	Jombang	1999-06-01	Jalan Sirsat
160533608158	Dona	Perempuan	Bandung	1998-05-16	Jalan Jeruk
160533608189	Erni	Perempuan	Lampung	1997-04-10	Jalan Anggur
160533608203	Dewi	Perempuan	Jogjakarta	1998-04-18	Jalan Markisa

2. Jika diketahui NIM mahasiswa adalah 130533608145, maka:

- 2 digit NIM pertama adalah kode tahun angkatan
- 2 digit NIM berikutnya adalah kode fakultas
- 1 digit NIM berikutnya adalah kode jurusan
- 1 digit NIM berikutnya adalah kode prodi
- 1 digit NIM berikutnya adalah kode jenjang
- 5 digit NIM terakhir adalah nomor urut mahasiswa

Tuliskan pernyataan SQL untuk mendapatkan data mahasiswa dengan **kode tahun angkatan = 16**, kemudian urutkan secara ascending!

Sehingga akan menampilkan data sebagai berikut:

nim	nama	jenis_kelamin	tempat_lahir	tanggal_lahir	alamat
160533608100	Dera	laki-laki	Surabaya	1997-05-17	Jalan Melon
160533608101	Budi	Laki-Laki	Medan	1998-10-29	Jalan Apel
160533608135	Fahmi	Laki-laki	Jombang	1999-06-01	Jalan Sirsat
160533608158	Dona	Perempuan	Bandung	1998-05-16	Jalan Jeruk
160533608189	Erni	Perempuan	Lampung	1997-04-10	Jalan Anggur
160533608203	Dewi	Perempuan	Jogjakarta	1998-04-18	Jalan Markisa

3. Tuliskan pernyataan SQL untuk mendapatkan data mahasiswa dengan kode fakultas = 05 **DAN** jenis kelamin = laki-laki. Urutan berdasarkan nim secara descending!

nim	nama	jenis_kelamin	tempat_lahir	tanggal_lahir	alamat
140533606464	Lugas	Laki-Laki	Batu	1995-04-19	Jalan Mangga
160533608100	Dera	laki-laki	Surabaya	1997-05-17	Jalan Melon
160533608101	Budi	Laki-Laki	Medan	1998-10-29	Jalan Apel
160533608135	Fahmi	Laki-laki	Jombang	1999-06-01	Jalan Sirsat

4. Tuliskan pernyataan SQL untuk mendapatkan data nim, nama, jenis kelamin, dan alamat pada tabel mahasiswa yang memiliki nim dengan kode tahun angkatan = 14 dan alamat <> "jalan mangga". Apa kegunaan "<>"?

nim	nama	jenis_kelamin	alamat
140533601613	Cintya	Perempuan	Jalan Apel

5. Analisislah *query* yang terdapat pada soal nomor 2, 3, dan 4!