

## MODUL 3 – Kelompok Perintah SQL

### 1.1. CAPAIAN PEMBELAJARAN

1. Mampu menjelaskan DDL dan DML
2. Mampu menggunakan perintah *Data Manipulation Language* (DDL)

### 1.2. ALAT DAN BAHAN

1. Seperangkat komputer lengkap/Laptop dengan koneksi internet
2. Sistem Operasi Windows/Mac/Linux
3. Aplikasi Paket Web server XAMPP
4. Aplikasi Kantor (Microsoft Office/Libre Office/WPS Office/etc)

### 1.3. DASAR TEORI

Data Manipulation Language (DML) adalah komponen kunci dari bahasa SQL (Structured Query Language) yang digunakan untuk memanipulasi data dalam basis data relasional. DML memberikan kemampuan kepada pengguna untuk melakukan operasi dasar seperti menyisipkan, memperbarui, menghapus, dan mengambil data dari tabel. Perintah INSERT memungkinkan pengguna untuk menambahkan baris baru ke dalam tabel dengan menentukan nilai untuk setiap kolom. Operasi ini berguna untuk memasukkan data baru ke dalam basis data. Sebagai contoh, perintah `INSERT INTO nama_tabel (kolom1, kolom2) VALUES (nilai1, nilai2);` akan menyisipkan data baru ke dalam tabel sesuai dengan nilai yang ditentukan.

Selanjutnya, perintah UPDATE digunakan untuk memperbarui nilai dari satu atau beberapa kolom dalam satu atau beberapa baris dalam sebuah tabel. Dengan menggunakan perintah ini, pengguna dapat mengubah nilai yang sudah ada menjadi nilai baru sesuai kebutuhan. Misalnya, perintah `UPDATE nama_tabel SET kolom1 = nilai1 WHERE kondisi;` akan mengubah nilai kolom1 dalam tabel nama\_tabel di mana kondisi yang ditentukan terpenuhi.

Perintah DELETE memungkinkan pengguna untuk menghapus satu atau beberapa baris dari sebuah tabel berdasarkan kondisi yang ditentukan. Dengan menggunakan DELETE, pengguna dapat menghapus data yang tidak diinginkan atau tidak relevan dari basis data. Sebagai contoh, perintah `DELETE FROM nama_tabel WHERE kondisi;` akan menghapus baris-baris dari tabel nama\_tabel di mana kondisi yang ditentukan terpenuhi.

Meskipun perintah SELECT secara teknis bukan termasuk dalam DML, namun juga

merupakan bagian penting dari manipulasi data dalam SQL. Perintah SELECT digunakan untuk mengambil atau menampilkan data dari satu atau beberapa tabel dalam basis data. Ini memungkinkan pengguna untuk menentukan kolom-kolom tertentu yang ingin diambil, serta menerapkan berbagai klausa seperti WHERE, ORDER BY, dan GROUP BY untuk memfilter dan mengurutkan data sesuai kebutuhan.

Dengan demikian, DML merupakan bagian yang sangat penting dari SQL karena memberikan kemampuan kepada pengguna untuk melakukan operasi dasar yang diperlukan untuk memanipulasi data dalam sebuah basis data relasional. Dengan menggunakan perintah-perintah DML yang sesuai, pengguna dapat dengan mudah menyisipkan, memperbarui, menghapus, dan mengambil data, yang merupakan bagian penting dari manajemen data dalam sistem basis data.

## **1.4. PRAKTIKUM**

### **1.4.1. Perintah INSERT**

Perintah SQL INSERT INTO digunakan untuk menambahkan atau menyisipkan data baru ke dalam sebuah tabel dalam sebuah basis data relasional. Ini adalah salah satu perintah yang paling sering digunakan dalam SQL dan memberikan kemampuan untuk menambahkan data ke dalam tabel secara programatik.

Perintah INSERT INTO memungkinkan Kita untuk secara dinamis menambahkan data baru ke dalam tabel sesuai dengan kebutuhan aplikasi atau skenario bisnis Kita. Ini adalah operasi fundamental dalam manajemen data dan digunakan dalam banyak aplikasi untuk memperbarui basis data dengan data baru. Penting untuk memastikan bahwa nilai-nilai yang dimasukkan sesuai dengan tipe data dan batasan yang diberlakukan pada kolom dalam tabel untuk menghindari kesalahan atau kegagalan dalam operasi penambahan data.

Pernyataan INSERT INTO dapat ditulis dengan dua cara:

1. Tentukan nama kolom dan nilai yang akan disisipkan:

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

2. Jika Anda menambahkan nilai untuk semua kolom tabel, Anda tidak perlu menentukan nama kolom dalam kueri SQL. Namun, pastikan urutan nilainya sama dengan urutan kolom pada tabel. Di sini, sintaks INSERT INTO adalah sebagai berikut:

```
INSERT INTO table_name VALUES (value1, value2, value3, ...);
```

Contoh penggunaan perintah INSERT INTO pada tabel anggota adalah sebagai berikut :

Field	Type	Null	Key	Default	Extra
ID_ANGGOTA	varchar(10)	NO	PRI	NULL	
ID_ADMIN	varchar(10)	YES	MUL	NULL	
FULL_NAME	varchar(128)	YES		NULL	
TMP_LAHIR	varchar(90)	YES		NULL	
TGL_LAHIR	varchar(20)	YES		NULL	
ALAMAT	mediumtext	YES		NULL	
GENDER	enum('l','p')	YES		NULL	
TELP	varchar(20)	YES		NULL	
FOTO	varchar(100)	YES		NULL	
D_CREATED	date	YES		NULL	

*Gambar 1. Tabel Anggota*

Perintah untuk memasukkan semua atribut di tabel anggota pada Gambar 1. Tabel Anggotasebagai berikut :

```
MariaDB [perpus]> INSERT INTO anggota VALUES ('AG00000001', 'AD001', 'Imam Adi Nata', 'Bantul', '1988/01/01', 'mBantul DIY', 'l', '081215529989', 'imam.jpg', '2024/03/05');
Query OK, 1 row affected (0.085 sec)
MariaDB [perpus]> |
```

*Gambar 2. Memasukkan data pada tabel anggota*

Perintah SQL untuk memasukkan data berhasil jika respon yang tampil setelah dieksekusi adalah “Query OK, 1 row affected”. Contoh selanjutnya adalah memasukkan data pada kolom secara spesifik adalah sebagai berikut :

```
MariaDB [perpus]> INSERT INTO anggota
-> ('ID_ANGGOTA', 'FULL_NAME', 'GENDER')
-> VALUES ('AG00000002', 'Brad pitt', 'l');
Query OK, 1 row affected (0.044 sec)
```

*Gambar 3. Memasukkan data secara spesifik*

Perintah SQL pada Gambar 3 digunakan untuk memasukkan data anggota hanya pada kolom ID\_ANGGOTA, FULL\_NAME, dan GENDER saja, sehingga atribut lainnya dibiarkan kosong.

#### 1.4.2. Perintah SELECT

Query SELECT merupakan salah satu perintah utama dalam bahasa SQL (Structured Query Language) yang digunakan untuk mengambil atau menampilkan data dari satu atau beberapa tabel dalam sebuah basis data. Ini adalah perintah yang paling sering digunakan dalam SQL dan memberikan kemampuan untuk mengambil informasi yang diperlukan dari basis data. Berikut adalah penjelasan lengkap tentang Query SELECT:

```
SELECT [kolom1, kolom2, ...] FROM nama_tabel WHERE kondisi [ORDER BY kolom] [LIMIT jumlah];
```

Penjelasan:

- **SELECT:** Ini adalah kata kunci yang menunjukkan bahwa Anda ingin mengambil data dari tabel tertentu. Anda dapat menentukan kolom-kolom tertentu yang ingin Anda ambil dengan menggunakan nama kolom, atau menggunakan wildcard (\*) untuk mengambil semua kolom.
- **kolom1, kolom2, ...:** Ini adalah daftar kolom yang ingin Anda ambil dari tabel. Anda dapat menentukan satu atau beberapa kolom, atau menggunakan wildcard (\*) untuk mengambil semua kolom.
- **FROM:** Ini adalah kata kunci yang menunjukkan dari mana Anda ingin mengambil data. Anda harus menentukan nama tabel dari mana Anda ingin mengambil data.
- **nama\_tabel:** Ini adalah nama tabel dari mana Anda ingin mengambil data.
- **WHERE:** Ini adalah klausa opsional yang memungkinkan Anda untuk memfilter data berdasarkan kondisi tertentu. Hanya baris-baris yang memenuhi kondisi yang akan diambil.
- **kondisi:** Ini adalah ekspresi logis yang menentukan kriteria untuk memfilter baris yang akan diambil. Misalnya, WHERE kolom = nilai.
- **ORDER BY:** Ini adalah klausa opsional yang memungkinkan Anda untuk mengurutkan hasil berdasarkan kolom tertentu dalam urutan menaik atau menurun.
- **kolom:** Ini adalah nama kolom yang ingin Anda gunakan untuk mengurutkan hasil. Misalnya, ORDER BY kolom ASC untuk urutan menaik atau ORDER BY kolom DESC untuk urutan menurun.
- **LIMIT:** Ini adalah klausa opsional yang membatasi jumlah baris yang akan diambil dari hasil. Ini berguna terutama ketika Anda hanya ingin mengambil sejumlah kecil baris dari tabel yang besar.

- **jumlah:** Ini adalah jumlah maksimum baris yang akan diambil dari hasil. Misalnya, LIMIT 10 akan membatasi hasil menjadi 10 baris.

Data yang dikembalikan disimpan dalam tabel hasil, yang disebut *record-set*. Berikut adalah contoh penggunaan perintah SELECT pada tabel anggota :

### 1. SELECT semua kolom

Perintah ini digunakan untuk menampilkan data pada semua kolom dalam basis data.

```
MariaDB [perpus]> SELECT * FROM anggota;
```

ID_ANGGOTA	ID_ADMIN	FULL_NAME	TMP_LAHIR	TGL_LAHIR	ALAMAT	GENDER	TELP	FOTO	D_CREATED
AG00000001	AD001	Imam Adi Nata	Bantul	1988/01/01	mSantul DIY	l	881215529989	imam.jpg	2020-03-05
AG00000002	NULL	Brad Pitt	NULL	NULL	NULL	l	881215529989	NULL	2020-03-05
AGT001	AD001	Multazam Arroihan G	Bojonegara	02/20/2000	Jl. Danau Merinci V G604, Sawojajar, Malang	l	880733573529	20_Multazam_A_3.JPG	2017-06-16
AGT002	AD001	Galbie Elinour	Malang	06/16/2017	Jl. Kawi no. 89, Malang Kota	p	889778678994	12_GALBIE_ELINOUR1.JPG	2017-06-16
AGT003	AD001	Alfan Cahyo Wicaksono	Malang	10/13/1998	Jl. Siliak Kemah 101	l	887669827654	2_Alfan_Cahyo_1.JPG	2017-06-16
AGT004	AD001	Sasna Salsabila Hapsari	Madiun	03/08/1999	Perum. Aspol 043, Kota Madiun	p	8818897546788	30_Sasna_S_1.JPG	2017-06-16
AGT005	AD001	Riski Ananda Widiya P	Tulungagung	02/19/2000	Perum. Grand Emerald ZU3, Tulungagung	p	893889765643	31_Riski_Ananda.JPG	2017-06-16
AGT006	AD001	Anggyanisa Mutia Putri	Madiun	03/15/2000	Gg. Pattimura 89	p	883389755644	2_Anggyanisa_M_1.JPG	2017-06-16
AGT007	AD001	Prasetya Ananta W. S.	Malang	06/16/1999	Sulfat Indah 89	l	88267599432	28_Prasetya_Ananta.JPG	2017-06-16
AGT008	AD001	Irvina Firadila	Lumajang	08/12/1999	Ds. Sambungrejo RT. 4 RW. 7 Kec. Melasasih, Lumajang	p	8876654897765	14_Irvina_F_1.JPG	2017-06-16
AGT009	AD001	Ridhwan Rofianto	Surabaya	04/03/2003	Murjini cilo	l	889776598334	29_RIDHWAN_ROFIANTO.JPG	2017-06-16
AGT010	AD001	Muhammad Iqbal Samudra	Banyuwangi	04/05/2000	Ds. Sumoharjo, RT. 5 RW. 9	l	88677564987	18_Muham_Iqbal.JPG	2017-06-16
AGT011	AD001	Nabila Hanggana Raras	Sidoarjo	03/08/2000	Perum. Sugih Asri F32, Candi, Sidoarjo	p	884878635267	28_Nabila_Hanggana_Raras.JPG	2017-06-16
AGT012	AD001	Erika Prafitasari Rotinsulu	Sidoarjo	03/06/2000	Perum. Bumi Citra Fajar A76, Sekardangan, Sidoarjo	p	88567698256	9_Erika_P_1.JPG	2017-06-16
AGT013	AD001	Nico Ardian Nugroho	Malang	04/02/2000	Jl. Sironggo Ajoh Kav. 4, Malang	l	889765689645	24_Nico_Ardian.JPG	2017-06-16
AGT014	AD001	Ad Reana Vidya	Malang	06/29/2000	Perum. Sirewa Taman AG65, Kepanjen	p	887989764839	2_Ad_Reana.JPG	2017-06-29
AGT015	AD001	Vejryn Shaviero	Jember	06/10/2000	Jl. Utama Kota gg. 4, Jember	l	885763689495	34_Vejryn_SHAVIERO.JPG	2017-06-29
AGT016	AD001	Ridjal Fathoni Rosyidi	Tuban	07/09/1998	Jl. Alun alun kota	l	88766754345	19_Muham_Ridjal.JPG	2017-07-09

18 rows in set (0.040 sec)

Gambar 4. Perintah SELECT pada semua kolom

### 2. SELECT spesifik kolom

Perintah ini digunakan untuk menampilkan data pada kolom yang spesifik.

```
MariaDB [perpus]> SELECT FULL_NAME, GENDER FROM anggota;
```

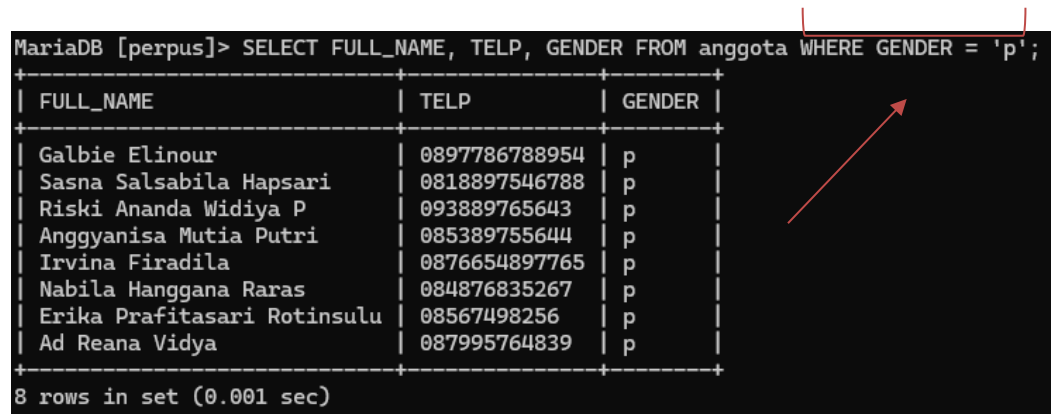
FULL_NAME	GENDER
Imam Adi Nata	l
Brad pitt	l
Multazam Arroihan G	l
Galbie Elinour	p
Alfan Cahyo Wicaksono	l
Sasna Salsabila Hapsari	p
Riski Ananda Widiya P	p
Anggyanisa Mutia Putri	p
Prasetya Ananta W. S.	l
Irvina Firadila	p
Ridhwan Rofianto	l
Muhamad Iqbal Samudra	l
Nabila Hanggana Raras	p
Erika Prafitasari Rotinsulu	p
Nico Ardian Nugroho	l
Ad Reana Vidya	p
Vejryn Shaviero	l
Ridjal Fathoni Rosyidi	l

18 rows in set (0.001 sec)

Gambar 5. Perintah SELECT dengan kolom yang spesifik

### 3. SELECT dengan kondisi

SELECT dengan kondisi digunakan untuk mencari data-data secara spesifik, misal kita ingin melihat data anggota yang memiliki jenis kelamin perempuan saja maka kita gunakan kolom sebagai kondisi pada perintah SELECT. Letak kondisi pada perintah SELECT berada pada bagian belakang perintah setelah nama tabel ditambahkan beberapa kalimat perintah dengan dimulai “WHERE” lalu dilanjutkan kondisi yang diinginkan. Penggunaan SELECT dengan kondisi dapat dilihat pada contoh berikut :



```
MariaDB [perpus]> SELECT FULL_NAME, TELP, GENDER FROM anggota WHERE GENDER = 'p';
```

FULL_NAME	TELP	GENDER
Galbie Elinour	0897786788954	p
Sasna Salsabila Hapsari	0818897546788	p
Riski Ananda Widiya P	093889765643	p
Anggyanisa Mutia Putri	085389755644	p
Irvina Firadila	0876654897765	p
Nabila Hanggana Raras	084876835267	p
Erika Prafitasari Rotinsulu	08567498256	p
Ad Reana Vidya	087995764839	p

8 rows in set (0.001 sec)

*Gambar 6. SELECT dengan kondisi*

### 1.4.3. Perintah UPDATE

Perintah SQL UPDATE adalah salah satu perintah yang penting dalam bahasa SQL yang digunakan untuk memperbarui data yang ada dalam tabel sebuah basis data relasional. Ini memungkinkan pengguna untuk mengubah nilai dari satu atau beberapa kolom dalam satu atau lebih baris sesuai dengan kriteria yang ditentukan. Perintah UPDATE memberikan fleksibilitas dalam mengelola data yang ada, baik untuk memperbaiki kesalahan, mengupdate informasi, atau mengubah status dari data dalam basis data.

Sintaksis dasar dari perintah UPDATE melibatkan dua kata kunci utama, yaitu UPDATE dan SET. Kata kunci UPDATE menunjukkan bahwa perintah yang akan dijalankan adalah untuk memperbarui data dalam tabel, sementara kata kunci SET digunakan untuk menetapkan nilai baru untuk kolom-kolom yang akan diperbarui. Klausula WHERE adalah opsional, namun sangat penting karena memungkinkan untuk membatasi baris-baris mana yang akan diperbarui berdasarkan kriteria tertentu.

Proses menggunakan perintah UPDATE dimulai dengan menentukan tabel yang akan diperbarui, diikuti dengan menetapkan nilai baru untuk kolom-kolom yang ingin diperbarui dengan menggunakan klausa SET. Kondisi yang akan membatasi baris-baris yang akan diperbarui kemudian ditentukan menggunakan klausa WHERE. Ini memungkinkan untuk memperbarui data secara selektif berdasarkan kriteria yang spesifik, seperti nama karyawan,

nomor identitas, atau status tertentu.

Penggunaan perintah UPDATE harus dilakukan dengan hati-hati, terutama ketika menentukan klausa WHERE, karena kesalahan dalam penulisan kondisi dapat mengakibatkan perubahan yang tidak diinginkan pada data yang ada. Oleh karena itu, sebaiknya selalu melakukan uji coba terlebih dahulu pada data yang tidak kritis sebelum menjalankan perintah UPDATE pada data yang penting. Dengan pemahaman yang baik tentang sintaks dan fungsi perintah UPDATE, pengguna dapat memanfaatkannya secara efektif untuk memelihara dan memperbarui data dalam basis data mereka sesuai dengan kebutuhan.

Contoh perintah update untuk mengubah alamat pada anggota dengan ID\_ANGGOTA = 'AG00000001' adalah sebagai berikut :

```
MariaDB [perpus]> UPDATE `anggota` SET `ALAMAT` = 'Bantul DIY' WHERE ID_ANGGOTA = 'AG00000001';
Query OK, 1 row affected (0.046 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [perpus]> SELECT FULL_NAME, ALAMAT FROM anggota;
```

FULL_NAME	ALAMAT
Imam Adi Nata	Bantul DIY
Brad pitt	NULL
Multazam Arroihan G	Jl. Danau Kerinci V G6D4, Sawojajar, Malang
Galbie Elinour	Jl. Kawi no. 89, Malang Kota
Alfan Cahyo Wicaksono	Jl. Silak Kesam 101
Sasna Salsabila Hapsari	Perum. Aspol 443, Kota Madiun
Riski Ananda Widiya P	Perum. Grand Emerald ZU3, Tulungagung
Anggyanisa Mutia Putri	Gg. Pattimura 89
Prasetya Ananta W. S.	Sulfat Indah 89
Irvina Firadila	Ds. Sambungrejo RT. 4 RW. 7 Kec. Welasasih, Lumajang
Ridhwan Rofianto	Mburine cito
Muhamad Iqbal Samudra	Ds. Sumoharjo, RT. 5 RW. 9
Nabila Hanggana Raras	Perum. Sugih Asri F32, Candi, Sidoarjo
Erika Prafitasari Rotinsulu	Perum. Bumi Citra Fajar A76, Sekardangan, Sidoarjo
Nico Ardian Nugroho	Jl. Sironggo Ajoh Kav. 4, Malang
Ad Reana Vidya	Perum. Siswa Taman AG65, Kepanjen
Vejryn Shaviero	Jl. Utama Kota gg. 6, Jember
Ridjal Fathoni Rosyidi	Jl. Alun alun kota

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

*Gambar 7. Perintah UPDATE pada tabel anggota*

#### 1.4.4. Perintah DELETE

Perintah DELETE pada MySQL digunakan untuk menghapus satu atau lebih baris data dari sebuah tabel dalam sebuah basis data relasional. Perintah DELETE merupakan salah satu perintah kritis dalam SQL yang memungkinkan pengguna untuk menghapus data yang tidak diinginkan atau tidak relevan dari basis data.

Sintaksis dasar dari perintah DELETE melibatkan kata kunci utama DELETE FROM, diikuti dengan nama tabel yang akan dihapus baris-datanya. Pengguna dapat juga menentukan kondisi dengan menggunakan klausa WHERE untuk membatasi baris-baris mana yang akan dihapus. Tanpa klausa WHERE, perintah DELETE akan menghapus semua baris dari tabel, sehingga perlu dilakukan dengan hati-hati.

Perintah DELETE secara umum digunakan dalam skenario ketika data yang ada sudah tidak lagi relevan atau dibutuhkan dalam basis data, atau ketika data tersebut tidak diinginkan lagi. Contohnya, jika ada entri ganda dalam sebuah tabel, perintah DELETE dapat digunakan untuk menghapus satu dari entri tersebut. Demikian juga, jika ada data yang sudah usang atau tidak relevan, perintah DELETE dapat membantu membersihkan tabel dari data tersebut.

Namun, perlu diingat bahwa perintah DELETE bersifat permanen, artinya data yang dihapus tidak dapat dikembalikan dengan mudah. Oleh karena itu, sebelum menjalankan perintah DELETE, sangat disarankan untuk membuat cadangan (backup) data yang akan dihapus terlebih dahulu. Hal ini akan membantu untuk menghindari kehilangan data yang tidak disengaja atau memungkinkan untuk memulihkan data jika terjadi kesalahan.

Selain itu, penggunaan perintah DELETE juga harus dilakukan dengan hati-hati, terutama ketika menggunakan klausa WHERE. Kondisi yang tidak tepat dapat menyebabkan penghapusan data yang tidak diinginkan atau bahkan menghapus semua data dalam sebuah tabel. Oleh karena itu, selalu penting untuk memastikan bahwa klausa WHERE dibuat dengan cermat dan telah diuji sebelum menjalankan perintah DELETE dalam produksi.

Dengan memahami dengan baik fungsi, sintaks, dan potensi risiko dari perintah DELETE, pengguna dapat menggunakan alat ini secara efektif untuk membersihkan, memelihara, dan mengelola data dalam basis data MySQL mereka sesuai dengan kebutuhan aplikasi atau skenario bisnis.

Berikut adalah contoh penggunaan perintah DELETE pada tabel anggota. Kita akan menghapus data anggota dengan ID\_ANGGOTA 'AG00000002' sehingga perintah yang kita gunakan seperti pada Gambar 8.



```

MariaDB [perpus]> DELETE FROM `anggota` WHERE `ID_ANGGOTA` = 'AG00000002';
Query OK, 1 row affected (0.045 sec)

MariaDB [perpus]> SELECT ID_ANGGOTA, FULL_NAME FROM `anggota`;
+-----+-----+
| ID_ANGGOTA | FULL_NAME |
+-----+-----+
| AG00000001 | Imam Adi Nata |
| AGT001 | Multazam Arroihan G |
| AGT002 | Galbie Elinour |
| AGT003 | Alfian Cahyo Wicaksono |
| AGT004 | Sasna Salsabila Hapsari |
| AGT005 | Riski Ananda Widiya P |
| AGT006 | Anggyanisa Mutia Putri |
| AGT007 | Prasetya Ananta W. S. |
| AGT008 | Irvina Firadila |
| AGT009 | Ridhwan Rofianto |
| AGT010 | Muhamad Iqbal Samudra |
| AGT011 | Nabila Hanggana Raras |
| AGT012 | Erika Prafitasari Rotinsulu |
| AGT013 | Nico Ardian Nugroho |
| AGT014 | Ad Reana Vidya |
| AGT015 | Vejryn Shaviero |
| AGT016 | Ridjal Fathoni Rosyidi |
+-----+-----+
17 rows in set (0.001 sec)

```

Gambar 8. Perintah DELETE pada tabel anggota.

## 1.5. TUGAS MODUL 4

### 1.5.1. Soal

1. Buatlah tabel dengan struktur sebagai berikut! (Poin :20)

Nama	Type Data	Keterangan
NPM*	Char(5)	Primary Key, Not Null
Nama	Varchar(25)	Not Null
Tempat Lahir	Varchar(30)	Not Null
Tanggal Lahir	Date	Not Null
Jenis Kelamin	Enum('L','P')	Not Null
No_Hp	Varchar(25)	Not Null

2. Buat Perintah SQL DML untuk mengisi data pada tabel tersebut (minimal 5 data) (Poin : 20)
3. Buat perintah SQL DML untuk mengubah nama\_depan pada data ke 3 (poin : 20)
4. Buat perintah SQL DML untuk menghapus data ke 4. (point : 10)
5. Tampilkan semua data dan kolom pada tabel tersebut! (poin : 10)
6. Tampilkan data NPM, Nama dan No\_HP dengan jenis kelamin Laki-Laki. (poin : 20)

### 1.5.2. Petunjuk Pengerjaan

#### a) File Query:

- File basis data disimpan di dalam file backup MySQL.
- Simpan file dengan nama "**TugasModul4\_DBMS\_NPM.sql**". Untuk NPM diisi sesuai milik Anda.

#### b) Laporan:

- Buatlah laporan akhir berdasarkan diagram yang Anda buat.
- Laporan dibuat sesuai format dan **diketik**.
- Masukkan langkah-langkah pengerjaan tugas ke dalam laporan dalam bentuk screenshot dan penjelasan.
- **Cantumkan tanda tangan** Anda di setiap halaman dokumen laporan.
- Jika di dalam laporan ada gambar atau *screenshot* yang ingin ditampilkan, bisa ditempelkan pada halaman(menyesuaikan)
- Laporan disimpan dalam bentuk pdf
- Penamaan *file* pdf : " **LaporanModul4\_DBMS\_NPM.pdf**"

c) Pengumpulan:

- *File* yang dikumpulkan yaitu:
  - Query: **TugasModul4\_DBMS\_NPM.sql**.
  - Laporan : **TugasModul4\_DBMS\_NPM.pdf**
- Kedua *file* di atas disatukan di dalam *folder* dan dikompres dalam bentuk .rar/zip
- *File* diunggah ke Elita dengan nama " **TugasModul4\_DBMS\_NPM.rar** "
- **Batas Pengumpulan:** Sebelum Pertemuan Praktik Ke 5.