

LATIHAN MODUL 2
LAPORAN PRAKTIKUM BASIS DATA
Data Definition Language (DDL)



Disusun oleh:

Nama : Stefen Tjung
NIM : 121140057
Kelas : Basis Data RA

PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN

2023

BAB 1

TEORI DASAR

A. Data Definition Language (DDL)

Data Definition Language atau DDL adalah sebuah perintah bahasa pemrograman pada SQL yang digunakan untuk membuat dan memodifikasi struktur dari suatu objek pada database, kemudian digambarkan desain secara menyeluruh berdasarkan database. MySQL itu sendiri adalah DBMS atau Database management system yang menggunakan bahasa SQL sebagai penghubung antara software dengan database server. DDL digunakan untuk membuat skema, tabel, indeks, dan lain sebagainya.

B. Perintah Dasar DDL

1. Create

- Create database => Untuk membuat database yang baru
- Create Table => Untuk membuat tabel yang baru
- Create Procedure => Untuk membuat prosedur yang baru
- Create index => Untuk membuat index yang baru
- Create function => Untuk membuat fungsi yang baru
- Create trigger => Untuk membuat reaksi tertentu atau trigger pada data base

2. Alter

Alter merupakan perintah dalam DDL yang digunakan untuk mengubah stuktur pada tabel yaitu Alter. Perintah ini digunakan untuk memodifikasi bentuk kolom, menambah dan juga mengganti tabel yang sudah ada

- Menambah Kolom Tabel
Berikut perintah Alter yang digunakan untuk menambah kolom table.
`ALTER TABLE nama_table ADD nama_field tipe_data`
- Modifikas Kolom Tabel
Berikut perintah Alter yang digunakan untuk memodifikasi kolom table.
`ALTER TABLE nama_table MODIFY nama_field tipe_data`
- Menghapus Kolom Tabel
Berikut perintah alter untuk menghapus kolom table.
`ALTER TABLE nama_table DROP nama_field`

3. Drop

Drop merupakan perintah dalam DDL yang digunakan untuk membuat perintah menghapus objek dalam database.

- Menghapus Database
Berikut perintah pada drop yang digunakan untuk menghapus database. `DROP DATABASE nama_database`
- Menghapus tabel
Berikut perintah pada drop yang digunakan untuk menghapus tabel dalam database. `DROP TABLE nama_table`

BAB 2

PEMBAHASAN DAN ANALISIS

1. Membuat Database dengan nama “Bank_Mino”

Sintaks yang digunakan untuk membuat database yang baru adalah “CREATE DATABASE nama_database”. Sehingga, untuk membuat database yang baru sesuai dengan nama database di atas ialah dengan cara mengetikkan perintah “CREATE DATABASE Bank_Mino” pada command prompt atau cmd. Perintah dan hasil tersebut dapat dilihat pada gambar sebagai berikut. Kemudian, ketikkan perintah SHOW DATABASES; untuk melihat list database yang sudah dibuat.

```
MariaDB [(none)]> create database Bank_Mino;  
Query OK, 1 row affected (0.004 sec)
```

```
MariaDB [(none)]> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| bank_mino |  
| information_schema |  
| mysql |  
| performance_schema |  
| phpmyadmin |  
| test |  
| universitas |  
+-----+  
7 rows in set (0.158 sec)
```

2. Membuat tabel mahasiswa

Membuat tabel dengan nama rekening pada database Bank_Mino. Pada Tabel tersebut berisi atribut no_rekening, kode_cabang, pin, dan saldo. Atribut no_rekening dijadikan sebagai *primary key*. Masing-masing atribut tersebut juga terdapat tipe datanya.

Untuk memilih database yang ingin dimanipulasi ialah dengan cara mengetikkan perintah “USE Bank_Mino”. Terdapat output “Database changed” yang menyatakan bahwa database tersebut telah berubah atau sedang diakses.

Selanjutnya, ketikkan perintah “CREATE table rekening” sesuai sintaks berikut yang diikuti dengan atribut serta tipe data dan juga *primary key* nya. Hal itu dapat dilihat pada gambar sebagai berikut.

```
MariaDB [(none)]> use Bank_Mino  
Database changed
```

```
MariaDB [Bank_Mino]> CREATE TABLE rekening (  
-> no_rekening int(11),  
-> kode_cabang char(5),  
-> pin char(6),  
-> saldo int(11),  
-> PRIMARY KEY (no_rekening) );  
Query OK, 0 rows affected (1.049 sec)
```

```
MariaDB [Bank_Mino]> DESC rekening;
```

Field	Type	Null	Key	Default	Extra
no_rekening	int(11)	NO	PRI	NULL	
kode_cabang	char(5)	YES		NULL	
pin	char(6)	YES		NULL	
saldo	int(11)	YES		NULL	

```
4 rows in set (0.070 sec)
```

3. Mengubah Struktur Tabel

Berikut cara dan sintaks aturan perubahan struktur pada tabel rekening a.

Tambahkan kolom “nama” setelah kolom kode_cabang.

Pada tabel rekening, tambahkan atribut nama yang dimana atribut tersebut bertipe int. Kolom atribut tersebut diletakkan setelah kolom kode_cabang. Sintaks yang digunakan adalah “ALTER TABLE rekening ADD nama varchar (20) after kode_cabang;”. Sehingga,. Hal itu dapat dilihat pada gambar sebagai berikut.

```
MariaDB [Bank_Mino]> ALTER TABLE rekening ADD nama varchar(20) AFTER kode_cabang;
Query OK, 0 rows affected (0.160 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [Bank_Mino]> DESC rekening;
```

Field	Type	Null	Key	Default	Extra
no_rekening	int(11)	NO	PRI	NULL	
kode_cabang	char(5)	YES		NULL	
nama	varchar(20)	YES		NULL	
pin	char(6)	YES		NULL	
saldo	int(11)	YES		NULL	

```
5 rows in set (0.064 sec)
```

b. Tambahkan kolom “keterangan” pada posisi akhir kolom

Pada tabel mahasiswa, tambahkan atribut keterangan yang dimana atribut tersebut bertipe varchar. Kolom tersebut diletakkan pada akhir kolom. Sintaks yang digunakan untuk menambahkan kolom adalah “ALTER TABLE rekening ADD keterangan varchar(50);”. Hal itu dapat dilihat pada gambar sebagai berikut.

```
MariaDB [Bank_Mino]> ALTER TABLE rekening ADD keterangan varchar (50);
Query OK, 0 rows affected (0.211 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [Bank_Mino]> DESC rekening;
```

Field	Type	Null	Key	Default	Extra
no_rekening	int(11)	NO	PRI	NULL	
kode_cabang	char(5)	YES		NULL	
nama	varchar(20)	YES		NULL	
pin	char(6)	YES		NULL	
saldo	int(11)	YES		NULL	
keterangan	varchar(50)	YES		NULL	

6 rows in set (0.024 sec)

- c. Ubah kolom nama menjadi “nama_lengkap”

Pada tabel mahasiswa, ubah atribut nama menjadi nama_lengkap yang dimana atribut tersebut bertipe varchar. Sintaks yang digunakan adalah ALTER TABLE rekening CHANGE nama nama_lengkap varchar (20);. Hal itu dapat dilihat pada gambar sebagai berikut.

```
MariaDB [Bank_Mino]> ALTER TABLE rekening CHANGE nama nama_lengkap varchar(20);
Query OK, 0 rows affected (0.423 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [Bank_Mino]> DESC rekening;
```

Field	Type	Null	Key	Default	Extra
no_rekening	int(11)	NO	PRI	NULL	
kode_cabang	char(5)	YES		NULL	
nama_lengkap	varchar(20)	YES		NULL	
pin	char(6)	YES		NULL	
saldo	int(11)	YES		NULL	
keterangan	varchar(50)	YES		NULL	

6 rows in set (0.099 sec)

- d. Hapus kolom keterangan

Pada tabel mahasiswa, hapus atribut keterangan. Sintaks yang digunakan adalah “ALTER TABLE rekening DROP keterangan;”. Hal itu dapat dilihat pada gambar sebagai berikut.

```
MariaDB [Bank_Mino]> ALTER TABLE rekening DROP keterangan;
Query OK, 0 rows affected (0.153 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [Bank_Mino]> DESC rekening;
```

Field	Type	Null	Key	Default	Extra
no_rekening	int(11)	NO	PRI	NULL	
kode_cabang	char(5)	YES		NULL	
nama_lengkap	varchar(20)	YES		NULL	
pin	char(6)	YES		NULL	
saldo	int(11)	YES		NULL	

```
5 rows in set (0.074 sec)
```

4. Buatlah tabel dengan nama “transaksi” dengan struktur tabel berikut.

Membuat tabel dengan nama transaksi pada database Bank_Mino. Pada Tabel tersebut berisi atribut no_transaksi, no_rekening, jenis_transaksi, dan jumlah. Atribut no_transaksi dijadikan sebagai *primary key*. Sedangkan Attribute no_rekening dijadikan sebagai Foreign_Key. Masing-masing atribut tersebut juga terdapat tipe datanya.

Selanjutnya, ketikkan perintah “CREATE table transaksi” sesuai sintaks berikut yang diikuti dengan atribut serta tipe data dan juga primary keynya. Hal itu dapat dilihat pada gambar sebagai berikut.

```
MariaDB [Bank_Mino]> CREATE TABLE transaksi (
  -> no_transaksi int(11),
  -> no_rekening int(11),
  -> jenis_transaksi char(10),
  -> jumlah int(11),
  -> PRIMARY KEY (no_transaksi),
  -> FOREIGN KEY (no_rekening) REFERENCES rekening (no_rekening) );
Query OK, 0 rows affected (0.618 sec)
```

```
MariaDB [Bank_Mino]> DESC transaksi;
```

Field	Type	Null	Key	Default	Extra
no_transaksi	int(11)	NO	PRI	NULL	
no_rekening	int(11)	YES	MUL	NULL	
jenis_transaksi	char(10)	YES		NULL	
jumlah	int(11)	YES		NULL	

```
4 rows in set (0.039 sec)
```

Perintah ON UPDATE RESTRICT itu berfungsi sebagai apabila terdapat penghapusan data pada suatu tabel, maka hal itu tidak akan diperbolehkan jika pada suatu tabel lainnya masih terdapat relasi datanya.

Perintah ON UPDATE CASCADE itu berfungsi sebagai apabila terdapat penghapusan data pada suatu tabel, maka akan secara otomatis akan dapat menghapus data yang sesuai dalam tabel lainnya.

5. Ubah struktur tabel “transaksi”

- a. Tambahkan kolom “tanggal” setelah kolom jenis_transaksi.

Pada table transaksi, tambahkan atribut tanggal yang dimana atribut tersebut bertipe varchar. Kolom atribut tersebut diletakkan setelah kolom jenis_transaksi. Sintaks yang digunakan untuk menambahkan kolom adalah “ALTER TABLE transaksi ADD tanggal varchar(8) AFTER jenis_transaksi”. Hal itu dapat dilihat pada gambar sebagai berikut.

```
MariaDB [Bank_Mino]> ALTER TABLE transaksi ADD tanggal varchar(8) AFTER jenis_transaksi;
Query OK, 0 rows affected (0.142 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [Bank_Mino]> DESC transaksi;
+-----+-----+-----+-----+-----+-----+
| Field          | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| no_transaksi   | int(11)   | NO   | PRI | NULL    |       |
| no_rekening    | int(11)   | YES  | MUL | NULL    |       |
| jenis_transaksi | char(10)  | YES  |     | NULL    |       |
| tanggal        | varchar(8)| YES  |     | NULL    |       |
| jumlah         | int(11)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.087 sec)
```

- b. Ubah kolom “tanggal” menjadi kolom “tanggal_transaksi”.

Pada table tanggal, ubah atribut dosen menjadi tanggal_transaksi yang dimana atribut tersebut bertipe varchar. Sintaks yang digunakan adalah ALTER TABLE transaksi CHANGE tanggal tanggal_transaksi varchar (8). Hal itu dapat dilihat pada gambar sebagai berikut.

```
MariaDB [Bank_Mino]> ALTER TABLE transaksi CHANGE tanggal tanggal_transaksi varchar(8);
Query OK, 0 rows affected (0.131 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [Bank_Mino]> DESC transaksi;
```

Field	Type	Null	Key	Default	Extra
no_transaksi	int(11)	NO	PRI	NULL	
no_rekening	int(11)	YES	MUL	NULL	
jenis_transaksi	char(10)	YES		NULL	
tanggal_transaksi	varchar(8)	YES		NULL	
jumlah	int(11)	YES		NULL	

5 rows in set (0.038 sec)

BAB III

KESIMPULAN

1. Terdapat penerapan dan cara-cara untuk membuat database yang baru pada MySQL. Sintaks yang digunakan ialah CREATE DATABASE nama_database.
2. Terdapat penerapan dan cara-cara untuk membuat tabel beserta atribut dan tipe datanya masing-masing. Sintaks yang digunakan untuk membuat tabel ialah CREATE TABLE nama_table kemudian didalamnya terdapat database yang berisi atribut dan tipe datanya serta atribut yang dijadikan sebagai *primary key*.
3. Terdapat penerapan dan cara-cara untuk mengubah struktur tabel pada database. Sintaks yang digunakan ialah ALTER TABLE table_name CHANGE old_name_column new_name_new_column new-data_type.
4. Terdapat penerapan dan cara-cara untuk menambahkan kolom struktur tabel pada databse. Sintaks yang digunakan ialah ALTER TABLE nama_table add new_column newdata_type.
5. Terdapat penerapan dan cara-cara untuk menghubungkan suatu tabel dengan tabel lainnya dengan melalui *foreign key*. Sintaks yang digunakan ialah "ALTER TABLE table_name ADD FOREIGN KEY(column_name_of_foreign_key) kemudian diikuti dengan REFERENCES table_name(column_name_as_references) ON UPDATE RESTRICT, ON UPDATE CASCADE;".
6. Perintah ON UPDATE RESTRICT itu berfungsi sebagai apabila terdapat penghapusan data pada suatu tabel, maka hal itu tidak akan diperbolehkan jika pada suatu tabel lainnya masih terdapat relasi datanya.
7. Perintah ON UPDATE CASCADE itu berfungsi sebagai apabila terdapat penghapusan data pada suatu tabel, maka akan secara otomatis akan dapat menghapus data yang sesuai dalam tabel lainnya.

DAFTAR PUSTAKA

<https://ngodingdata.com/memahami-perintah-ddl-mysql/> <https://mfikri.com/artikel/Data-definition-language-pada-MySQL.html>

Modul Praktikum Basis Data Pertemuan 2