

-Modul ini diperuntukan bagi Mata Kuliah Pemograman dan Basis Data Program Studi S1 Agribisnis-

Materi : Data Manipulation Language (DML) dan Data Definition Language (DDL)

1. Tujuan :

Setelah mempelajari modul ini, kalian bisa:

- Memahami konsep dasar DML dan DDL dalam SQL.
- Mampu membuat dan mengelola data menggunakan perintah SQL.
- Mampu membuat struktur tabel relasional sederhana dan mengimplementasikan Primary Key (PK) dan Foreign Key (FK).
- Mampu membaca dan membuat ERD (Entity Relationship Diagram) di MYSQL.

2. Dasar Teori

2.1. Pengertian

DDL (Data Definition Language) adalah bagian dari SQL yang digunakan untuk mendefinisikan struktur dari database, seperti membuat, mengubah, atau menghapus tabel dan database itu sendiri. Operasi DDL mencakup : Create, Alter.

Noted : Perintah DDL bersifat permanen dan langsung mengubah struktur database.

DML (Data Manipulation Language) adalah bagian dari SQL (Structured Query Language) yang digunakan untuk mengelola data dalam tabel database. Operasi DML mencakup : Insert, select, Update, Delete.

Noted : Perintah DML digunakan dalam proses operasional sehari-hari pada sistem informasi.

Perintah SQL	Fungsi
CREATE DATABASE	Membuat database baru
USE	Memilih database yang akan digunakan
CREATE TABLE	Membuat tabel baru dalam database
INSERT INTO	Menambahkan data ke dalam tabel
SELECT	Menampilkan data dari tabel
UPDATE	Mengubah data dalam tabel
DELETE	Menghapus data dari tabel
ALTER	Mengubah struktur tabel

2.2. Perbedaan DML dan DDL

Perbedaan	DDL	DML
Fokus	Struktur database	Isi data dalam tabel
Contoh perintah	CREATE, ALTER.	INSERT, SELECT, UPDATE
Waktu penggunaan	Awal/rekayasa database	Saat operasional sehari-hari
Efek	Permanen, langsung tersimpan	Bisa dibatalkan jika pakai transaksi

Perbedaan	DDL	DML
Bisa digunakan JOIN?	✗ Tidak	✓ Ya

3. STUDI KASUS : Sistem Informasi akademik mahasiswa.

TABEL MAHASISWA

NIM	Nama_Mhs	Alamat_Mhs
101	Andi	Bandung

TABEL MATA KULIAH

Kode Matkul	Nama Matkul
BD01	Basis Data
PIK02	PIK

TABEL DOSEN

ID_Dosen	Nama_Dosen	Status_Dosen
D01	Budi	Tetap

TABEL NILAI

NIM	Kode_Matkul	Nilai	ID_Dosen
101	BD01	A	D01
101	PIK02	B	D01

4. PRAKTIKUM

1. CREATE DATABASE

Membuat database baru untuk menyimpan semua tabel dan data.

```
CREATE DATABASE db_akademik;
```

Keterangan :

Anda membuat database baru dengan nama db_akademik. Wajib dilakukan pertama kali.

2. USE

Memilih database yang akan digunakan dalam sesi kerja SQL.

```
USE db_akademik;
```

Keterangan :

Anda memfokuskan semua query setelahnya ke database db_akademik.

3. CREATE TABLE

Membuat struktur tabel baru untuk menyimpan data.

Membuat tabel mahasiswa

```
CREATE TABLE mahasiswa (  
  NIM INT (10) PRIMARY KEY,  
  Nama_Mhs VARCHAR(50),  
  Alamat_Mhs VARCHAR(100)  
);
```

Keterangan : Kolom NIM dijadikan Primary Key karena unik untuk tiap mahasiswa.

Membuat Tabel mata kuliah

```
CREATE TABLE mata_kuliah (  
  Kode_Matkul VARCHAR(10) PRIMARY KEY,  
  Nama_Matkul VARCHAR(50)  
);
```

Keterangan : Kolom Kode_Matkul dijadikan Primary Key karena unik untuk tiap matkul.

Membuat Tabel Dosen

```
CREATE TABLE Dosen (  
  ID_Dosen VARCHAR(10) PRIMARY KEY,  
  Nama_Dosen VARCHAR(50),  
  Status_Dosen ENUM('Tetap', 'Honorar')  
);
```

Keterangan : Kolom ID_Dosen dijadikan Primary Key karena unik untuk tiap dosen . kolom Status_Dosen hanya boleh berisi 'Tetap' atau 'Honorar'. Kalau masukkan nilai lain, akan error.

#Membuat tabel nilai

```
CREATE TABLE nilai (  
  NIM INT(10),  
  Kode_Matkul VARCHAR(10),  
  Nilai VARCHAR(2),  
  ID_Dosen VARCHAR(10),  
  FOREIGN KEY (NIM) REFERENCES mahasiswa(NIM),  
  FOREIGN KEY (Kode_Matkul) REFERENCES mata_kuliah(Kode_Matkul),
```

```
FOREIGN KEY (ID_Dosen) REFERENCES dosen(ID_Dosen)
);
```

Keterangan : Tabel nilai menggunakan foreign key untuk menghubungkan 3 tabel utama.

4. INSERT INTO

Menambahkan data baru ke dalam tabel.

```
-- Tabel Mahasiswa
INSERT INTO mahasiswa (NIM, Nama_Mhs, Alamat_Mhs) VALUES
('101', 'Andi', 'Bandung');

-- Tabel Mata Kuliah
INSERT INTO mata_kuliah (Kode_Matkul, Nama_Matkul) VALUES
('BD01', 'Basis Data'),
('PIK02', 'PIK');

-- Tabel Dosen
INSERT INTO dosen (ID_Dosen, Nama_Dosen, Status_Dosen) VALUES
('D01', 'Budi', 'Tetap');

-- Tabel Nilai
INSERT INTO nilai (NIM, Kode_Matkul, Nilai, ID_Dosen) VALUES
('101', 'BD01', 'A', 'D01'),
('101', 'PIK02', 'B', 'D01');
```

5. SELECT

Menampilkan data dari tabel.

```
SELECT * FROM mahasiswa;
```

Keterangan : Menampilkan seluruh isi tabel mahasiswa.

Gabungkan data dari beberapa tabel (JOIN):

```
SELECT nilai.NIM, mahasiswa>Nama_Mhs, mata_kuliah>Nama_Matkul,
nilai.Nilai, dosen>Nama_Dosen
FROM nilai
JOIN mahasiswa ON nilai.NIM = mahasiswa.NIM
JOIN mata_kuliah ON nilai.Kode_Matkul = mata_kuliah.Kode_Matkul
JOIN dosen ON nilai.ID_Dosen = dosen.ID_Dosen;
```

Keterangan : melihat semua data nilai lengkap dengan nama mahasiswa, matkul, dan dosennya.

OUTPUT :

NIM	Nama_Mhs	Nama_Matkul	Nilai	Nama_Dosen
101	Andi	Basis Data	A	Budi
101	Andi	PIK	B	Budi

6. DELETE

Menghapus data dari tabel.

```
DELETE FROM nilai
WHERE NIM = '101' AND Kode_Matkul = 'BD01';
```

Keterangan : menghapus nilai Andi untuk matkul Basis Data.

Noted : selalu gunakan WHERE untuk menghindari penghapusan semua data!

7. UPDATE

Mengubah nilai di suatu kolom pada baris tertentu.

```
UPDATE mahasiswa
SET Alamat_Mhs = 'Tasikmalaya'
WHERE NIM = '101';
```

Keterangan : alamat mahasiswa dengan NIM 101 diubah jadi Tasikmalaya.

8. ALTER TABLE

Mengubah struktur tabel, seperti menambahkan kolom baru.

```
ALTER TABLE mahasiswa
ADD No_HP VARCHAR(15);
```

Keterangan : menambahkan kolom No_HP ke tabel mahasiswa.

9. Membuat ERD di SQL

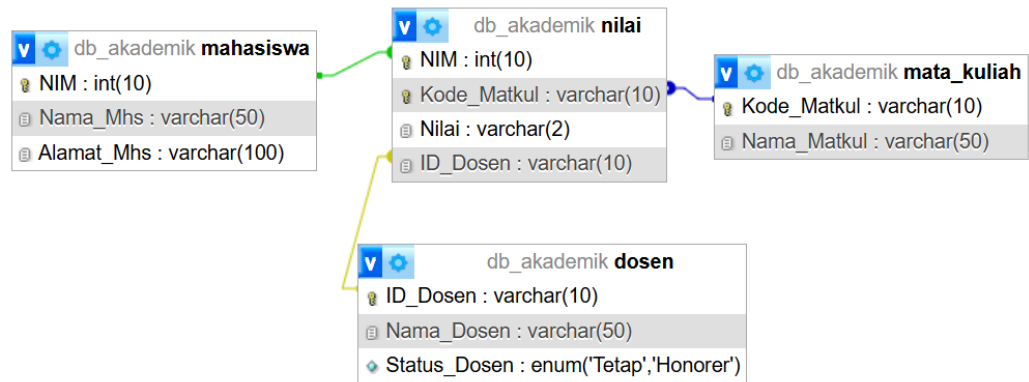
1. Buat Foreign Key Lewat Relation View

- Klik nama tabel yang ingin diberi foreign key, misalnya nilai.
- Klik tab Structure > scroll ke bawah > klik Relation view.
- Pada kolom NIM, set foreign key ke mahasiswa.NIM.
- Pada kolom Kode_Matkul, set FK ke mata_kuliah.Kode_Matkul.
- Pada kolom ID_Dosen, set FK ke dosen.ID_Dosen.
- Klik Save

2. Buka Tab Designer

- Setelah relasi dibuat, kembali ke halaman database.
- Klik tab Designer (paling atas atau di bagian More).
- phpMyAdmin akan menampilkan diagram otomatis:
- Tabel-tabel akan muncul sebagai kotak
- Relasi antar tabel akan digambarkan dengan garis (FK → PK)

Hasil ERD :



Keterangan :

Relasi	Tipe Relasi	Keterangan
mahasiswa → nilai	1:M (One to Many)	Satu mahasiswa bisa punya banyak nilai
mata_kuliah → nilai	1:M (One to Many)	Satu matkul bisa diambil banyak mahasiswa
dosen → nilai	1:M (One to Many)	Satu dosen bisa memberikan banyak nilai