

**LAPORAN RESMI**  
**MODUL I**  
**DDL & DML**  
**SISTEM MANAJEMEN BASIS DATA**



<b>NAMA</b>	<b>: ALIYUL RIDHO</b>
<b>N.R.P</b>	<b>: 230441100135</b>
<b>DOSEN</b>	<b>: FITRI DAMAYANTI, S.Kom., M.Kom.</b>
<b>ASISTEN</b>	<b>: ABDUL JABBAR RAMADHANI</b>
<b>TGL PRAKTIKUM</b>	<b>: 22 Maret 2025</b>

**Disetujui : 11 April 2025**  
**Asisten**

**Abdul Jabbar Ramadhani**  
**21.04.411.00062**



**LABORATORIUM TEKNOLOGI INFORMASI**  
**PRODI SISTEM INFORMASI**  
**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS TRUNOJOYO MADURA**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Basis data adalah komponen penting dari sistem informasi yang digunakan untuk menyimpan, mengelola, dan memproses data secara efektif. Faktanya, manajemen basis data dibuat oleh *SQL (Structured Query Language)*, termasuk sejumlah grup perintah, seperti *Data Definition Language (DDL)* dan *Data Manipulation Language (DML)*.

DDL digunakan untuk menentukan struktur basis data, seperti pembuatan, perubahan, dan penghapusan tabel atau skema. Sementara itu, DML berfungsi untuk memanipulasi data yang disimpan dalam basis data, seperti operasi penambahan, pembaruan, penghapusan, dan pengambilan data.

Dalam praktikum ini, implementasi perintah DDL dan DML telah diterapkan dalam pengelolaan basis data. Tujuan dari praktikum ini adalah untuk memahami konsep dan aplikasi dari kedua jenis perintah dalam Sistem Manajemen Basis Data (SMBD). Dengan adanya pengalaman ini, mahasiswa akan dapat merancang serta mengelola basis data dengan baik, serta memahami bagaimana setiap perintah SQL memengaruhi struktur dan isi dari basis data.

Selain itu, praktikum ini juga memberikan pemahaman mendalam tentang bagaimana transaksi basis data dilakukan secara efisien dan aman, termasuk aspek integritas data serta optimasi dalam pengelolaan informasi. Pemahaman yang diperoleh dari praktikum ini akan menjadi landasan penting bagi mahasiswa dalam menghadapi implementasi basis data dalam dunia nyata, baik dalam pengembangan perangkat lunak maupun analisis data di berbagai sektor industri.

### **1.2 Tujuan**

- Mampu memahami konsep dasar dari DDL dan DML
- Mampu melakukan atau mengimplementasikan proses DDL
- Mampu melakukan atau mengimplementasikan proses DML

## BAB II

### DASAR TEORI

#### 2.1 DDL (*Data Definition Language*)

##### 2.1.1 Pengertian DDL

*Data Definition Language* (DDL) adalah bagian dari SQL yang digunakan untuk mendefinisikan dan mengelola struktur dari suatu basis data. Perintah DDL digunakan untuk membuat, mengubah, dan menghapus objek dalam basis data seperti tabel, skema, indeks, dan relasi antar tabel. DDL bekerja pada tingkat struktural basis data dan tidak berhubungan langsung dengan manipulasi data.

Perintah DDL bersifat permanen, artinya perubahan yang dilakukan akan langsung tersimpan dalam sistem basis data tanpa perlu melakukan commit. Oleh karena itu, perintah DDL harus digunakan dengan hati-hati untuk menghindari perubahan yang tidak diinginkan.

##### 2.1.2 Penggunaan DDL

DDL digunakan dengan menuliskan perintah SQL dalam suatu query, lalu menjalankannya pada Sistem Manajemen Basis Data (DBMS) seperti MySQL, PostgreSQL, SQL Server, atau Oracle. Penggunaan DDL biasanya dilakukan dalam tahap awal perancangan basis data atau saat diperlukan perubahan struktural.

##### 2.1.3 *Query* Yang Biasa Digunakan

###### 1) *CREATE*

Biasa digunakan untuk membuat objek basis data seperti tabel, skema atau lainnya. Contoh penggunaannya :

```
CREATE TABLE Mahasiswa (  
    NIM VARCHAR(10) PRIMARY KEY,  
    Nama VARCHAR(50),  
    Jurusan VARCHAR(30),  
    Angkatan INT  
);
```

## 2) *ALTER*

Digunakan untuk mengubah struktur tabel, seperti menambah atau menghapus kolom. Contoh penggunaannya :

```
ALTER TABLE Mahasiswa ADD Email VARCHAR(50);
```

## 3) *DROP*

Digunakan untuk menghapus objek dalam basis data, seperti tabel atau skema. Contoh penggunaannya :

```
DROP TABLE Mahasiswa;
```

## 4) *TRUNCATE*

Digunakan untuk menghapus data dalam tabel tanpa menghapus struktur yang ada di dalamnya. Contoh penggunaannya :

```
TRUNCATE TABLE Mahasiswa;
```

## 5) *RENAME*

Digunakan untuk mengubah nama suatu tabel atau kolom. Contoh penggunaannya :

```
ALTER TABLE Mahasiswa RENAME TO Data_Mahasiswa;
```

## 2.2 *DML ( Data Manipulation Language )*

### 2.2.1 Pengertian DML ( Data Manipulation Language )

*Data Manipulation Language* (DML) adalah bagian dari SQL yang digunakan untuk memanipulasi data yang tersimpan dalam basis data. Berbeda dengan *Data Definition Language* (DDL) yang mengatur struktur basis data, DML berfokus pada pengolahan data di dalam tabel, seperti menambahkan, memperbarui, menghapus, dan mengambil data.

Perintah DML biasanya digunakan dalam aplikasi untuk berinteraksi dengan basis data, baik dalam transaksi sederhana maupun kompleks. Perintah DML bersifat tidak permanen sampai perintah COMMIT dijalankan, sehingga perubahan masih bisa dibatalkan menggunakan ROLLBACK sebelum disimpan secara permanen.

### 2.2.2 Penggunaan DML

DML digunakan dengan menuliskan perintah SQL dalam suatu query dan menjalankannya pada Sistem Manajemen Basis Data (DBMS) seperti MySQL, PostgreSQL, SQL Server, atau Oracle. Perintah DML sering digunakan dalam operasi harian untuk mengelola data yang tersimpan dalam tabel.

### 2.2.3 Query Yang Biasa Digunakan

a) *INSERT*

Digunakan untuk memasukkan data baru ke dalam tabel. Contoh penggunaanya :

```
sql INSERT INTO Mahasiswa (NIM, Nama, Jurusan, Angkatan) VALUES ('220101', 'Budi Santoso', 'Informatika', 2022);
```

b) *SELECT*

Digunakan untuk mengambil data dari suatu tabel. Contoh penggunaanya :

```
sql SELECT * FROM Mahasiswa WHERE Jurusan = 'Informatika';
```

c) *UPDATE*

Digunakan untuk memperbarui data yang ada dalam sebuah tabel di dalam data base. Contoh penggunaannya :

```
sql UPDATE Mahasiswa SET Nama = 'Andi Wijaya' WHERE NIM = '220101';
```

d) *DELETE*

Digunakan untuk menghapus sebuah data dari tabel yang didasari oleh kondisi tertentu. Contoh penggunaanya :

```
sql DELETE FROM Mahasiswa WHERE NIM = '220101';
```

BAB III  
TUGAS PENDAHULUAN

3.1 Soal

1. Jelaskan apa yang dimaksud dengan DDL (Data Definition Language) dan DML (Data Manipulation Language) dalam data base!
2. Berikan contoh perintah utama dari DDL (minimal 5) dan DML (minimal 5) dan jelaskan fungsi dari perintah tersebut!
3. Jelaskan perbedaan antara Count(), Sum(), Avg(), min(), dan max() dalam SQL!
4. Apa perbedaan antara perintah CREATE TABLE dan CREATE DATABASE dalam SQL?
5. Jelaskan apa yang dimaksud dengan primary key dan foreign key dalam pembuatan table!
6. Buatlah query SQL untuk membuat tabel dengan isi kolom (id, nama, prodi, angkatan) dengan primary key pada kolom id!

3.2 Jawaban

1. DDL digunakan untuk mendefinisikan dan mengelola struktur database, seperti tabel dan indeks. contoh: CREATE, ALTER, DROP. Sedangkan DML digunakan untuk mengelola data dalam tabel, seperti menambah, mengubah, dan menghapus data contoh: insert, update, delete.
2. DDL, contohnya :
  - CREATE TABLE = untuk membuat tabel baru
  - ALTER TABLE : Mengubah struktur tabel
  - DROP TABLE : menghapus tabel
  - CREATE INDEX : membuat indeks untuk mempercepat pencarian
  - TRUNCATE TABLE = menghapus semua data dalam tabel tanpa menghapus struktur yang sudah ada.



DML, Contohnya :

- SELECT : mengambil data dari tabel
- INSERT : menambahkan data ke tabel
- UPDATE : mengubah data dalam tabel
- DELETE : menghapus data dari tabel
- MERGE : menggabungkan data dari dua tabel

3. Perbedaan fungsi agregat dalam SQL, antara lain :

- a. COUNT() : menghitung jumlah baris
- b. SUM() : menjumlahkan nilai dalam kolom
- c. AVG() : menghitung nilai rerata dalam kolom
- d. MIN() : menghitung nilai terkecil dalam kolom
- e. MAX() : menghitung nilai maksimal dalam kolom

4. CREATE TABLE digunakan untuk membuat tabel baru yang digunakan sebagai identitas baru dalam database, Sedangkan CREATE DATABASE digunakan untuk membuat database baru yang akan berisi tabel.

5. Primary key sendiri merupakan kolom unik dalam tabel yang digunakan sebagai identitas utama setiap baris data (tidak boleh ada duplikat dan harus selalu memiliki nilai). Sedangkan untuk foreign key sendiri adalah kolom yang menghubungkan tabel satu dengan tabel lain, biasanya merujuk ke primary key di tabel lain untuk menjaga hubungan data.

6. CREATE TABLE mahasiswa (  
id INT PRIMARY KEY,  
nama VARCHAR (100),  
prodi VARCHAR (50),  
angkatan INT  
);

## **BAB IV**

### **IMPLEMENTASI**

#### **4.1 Tugas Praktikum**

##### **4.1.1 Soal 1**

a) Buatlah database akademik program studi dengan ketentuan berikut:

• TABEL UTAMA :

- Mahasiswa
- Dosen
- Mata Kuliah
- KRS

Aturlah relasi dari setiap tabel se kreatif mungkin. Diperbolehkan menambah tabel.

- b) Isi data pada tabel masing – masing 10 data untuk table Mahasiswa, Dosen dan
- c) Matakuliah
- d) Tampilkan keseluruhan data pada setiap tabel.
- e) Tambahkan data pada tabel KRS sebanyak 5 data.
- f) Lakukan perubahan pada salah satu nama tabel
- g) Hapus database tersebut.

##### **4.1.2 Soal 2**

Tentukan salah satu topik dan buatlah database sesuai topik yang sudah dipilih dengan ketentuan sebagai berikut:

1. Topik tidak boleh ada yang sama
2. Disarankan untuk topik bisa di list di grup mengantisipasi ada judul yang sama
3. Database yang dibuat harus terdapat tabel Master dan tabel Transaksi
4. Buatlah tabel pada database tersebut minimal 5 tabel
5. Aturlah relasi antar tabel se kreatif mungkin

Noted: Topik yang telah dipilih akan dipakai terus selama praktikum berlangsung.



## 4.2 Source Code

### 4.2.1 Tugas Praktikum Soal No. 1

- a) Membuat dan menggunakan database

```
CREATE DATABASE akademik;  
USE akademik;
```

- b) Membuat tabel yang dibutuhkan

```
CREATE TABLE jurusan (  
    id_jurusan INT PRIMARY KEY AUTO_INCREMENT,  
    nama_jurusan VARCHAR(100) NOT NULL  
);  
CREATE TABLE mahasiswa (  
    nim VARCHAR(12) PRIMARY KEY,  
    nama VARCHAR(100) NOT NULL,  
    tanggal_lahir DATE,  
    alamat VARCHAR(255),  
    id_jurusan INT,  
    FOREIGN KEY (id_jurusan) REFERENCES jurusan(id_jurusan) ON  
DELETE SET NULL  
);
```

- c) Mengisi data pada tabel yang diinginkan

```
INSERT INTO mahasiswa (nim, nama, tanggal_lahir, alamat, id_jurusan)  
VALUES  
('230441100001', 'Naruto Uzumaki', '2002-10-10', 'Konohagakure', 1),  
('230441100002', 'Sasuke Uchiha', '2002-07-23', 'Konohagakure', 1),  
('230441100003', 'Hinata Hyuga', '2002-12-27', 'Konohagakure', 2),  
('230441100004', 'Mikasa Ackerman', '2001-02-10', 'Shiganshina', 3),  
('230441100005', 'Eren Yeager', '2001-03-30', 'Shiganshina', 3),  
('230441100006', 'Gojo Satoru', '2000-12-07', 'Tokyo', 2),  
('230441100007', 'Itachi Uchiha', '1995-06-09', 'Konohagakure', 1),
```

```
('230441100008', 'Levi Ackerman', '1990-12-25', 'Underground City', 3),  
( '230441100009', 'Guts', '1985-04-05', 'Midland', 3),  
( '230441100010', 'Luffy D. Monkey', '2002-05-05', 'East Blue', 2);
```

d) Menampilkan tabel

```
SELECT * FROM mahasiswa;  
SELECT * FROM dosen;  
SELECT * FROM mata_kuliah;  
SELECT * FROM krs;  
SELECT * FROM detail_krs;
```

e) Menambahkan data ke tabel KRS

```
INSERT INTO krs (nim, semester, tahun_akademik) VALUES  
( '230441100001', 1, '2024/2025'),  
( '230441100002', 1, '2024/2025'),  
( '230441100003', 2, '2023/2024'),  
( '230441100004', 3, '2022/2023'),  
( '230441100005', 4, '2021/2022');
```

f) Merubah nama tabel

```
RENAME TABLE krs TO tbl_krs;
```

g) Menghapus database

```
DROP DATABASE akademik;
```

#### 4.2.2 Tugas Praktikum Soal No. 2

a) Membuat dan menggunakan database

```
CREATE DATABASE lure_store;  
USE lure_store;
```

b) Membuat table yang diperlukan

```
CREATE TABLE customers (  
    customer_id INT PRIMARY KEY AUTO_INCREMENT,  
    NAME VARCHAR(100) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    phone VARCHAR(15),  
    address VARCHAR(255),  
    created_at TIMESTAMP DEFAULT  
CURRENT_TIMESTAMP  
);  
  
CREATE TABLE products (  
    id_produk INT PRIMARY KEY AUTO_INCREMENT,  
    NAME VARCHAR(100) NOT NULL,  
    kategori VARCHAR(50) NOT NULL,  
    harga DECIMAL(10,2) NOT NULL,  
    stok INT NOT NULL DEFAULT 0,  
    deskripsi TEXT,  
    created_at TIMESTAMP DEFAULT  
CURRENT_TIMESTAMP  
);  
  
CREATE TABLE orders (  
    order_id INT PRIMARY KEY AUTO_INCREMENT,  
    customer_id INT NOT NULL,  
    order_date TIMESTAMP DEFAULT  
CURRENT_TIMESTAMP,  
    total_price DECIMAL(10,2) NOT NULL,  
    STATUS ENUM('Pending', 'Shipped', 'Completed', 'Canceled')  
DEFAULT 'Pending',  
    FOREIGN KEY (customer_id) REFERENCES  
customers(customer_id) ON DELETE CASCADE  
);
```

- c) Mengatur relasi antar tabel

```

FOREIGN KEY (order_id) REFERENCES orders(order_id) ON
DELETE CASCADE,

FOREIGN KEY (id_produk) REFERENCES products(id_produk)
ON DELETE CASCADE

```

### 4.3 Hasil

#### 4.3.1 Tugas Praktikum Soal No. 1

- a. Tabel mahasiswa

<input type="checkbox"/> nim	nama	tanggal_lahir	alamat	id_jurusan
<input type="checkbox"/> 230441100001	Naruto Uzumaki	2002-10-10	Konohagakure	1
<input type="checkbox"/> 230441100002	Sasuke Uchiha	2002-07-23	Konohagakure	1
<input type="checkbox"/> 230441100003	Hinata Hyuga	2002-12-27	Konohagakure	2
<input type="checkbox"/> 230441100004	Mikasa Ackerman	2001-02-10	Shiganshina	3
<input type="checkbox"/> 230441100005	Eren Yeager	2001-03-30	Shiganshina	3
<input type="checkbox"/> 230441100006	Gojo Satoru	2000-12-07	Tokyo	2
<input type="checkbox"/> 230441100007	Itachi Uchiha	1995-06-09	Konohagakure	1
<input type="checkbox"/> 230441100008	Levi Ackerman	1990-12-25	Underground City	3
<input type="checkbox"/> 230441100009	Guts	1985-04-05	Midland	3
<input type="checkbox"/> 230441100010	Luffy D. Monkey	2002-05-05	East Blue	2

- b. Tabel dosen

<input type="checkbox"/> nip	nama	email	telepon	id_jurusan
<input type="checkbox"/> 100100100001	Kakashi Hatake	kakashi@example.com	081234567890	1
<input type="checkbox"/> 100100100002	Jiraiya	jiraiya@example.com	082345678901	2
<input type="checkbox"/> 100100100003	Orochimaru	orochimaru@example.com	083456789012	3
<input type="checkbox"/> 100100100004	All Might	allmight@example.com	084567890123	1
<input type="checkbox"/> 100100100005	Aizawa Shouta	aizawa@example.com	085678901234	2
<input type="checkbox"/> 100100100006	Meruem	meruem@example.com	086789012345	3
<input type="checkbox"/> 100100100007	Silvers Rayleigh	rayleigh@example.com	087890123456	1
<input type="checkbox"/> 100100100008	Gojou Satoru	gojou@example.com	088901234567	2
<input type="checkbox"/> 100100100009	Genryusai Yamamoto	yamamoto@example.com	089012345678	3
<input type="checkbox"/> 100100100010	Zeno Zoldyck	zeno@example.com	080123456789	1

- c. Tabel mata\_kuliah

<input type="checkbox"/> kode_mk	nama_mk	sks	nip
<input type="checkbox"/> MK001	Jutsu Dasar	3	100100100001
<input type="checkbox"/> MK002	Kenjutsu Lanjutan	2	100100100002
<input type="checkbox"/> MK003	Teknik Kuno Seiryu	3	100100100003
<input type="checkbox"/> MK004	Analisis Strategi Peperangan	4	100100100004
<input type="checkbox"/> MK005	Quirk dan Evolusi	3	100100100005
<input type="checkbox"/> MK006	Teori Kutukan dan Penghapusan	3	100100100006
<input type="checkbox"/> MK007	Haki dan Buah Iblis	3	100100100007
<input type="checkbox"/> MK008	Bankai Mastery	3	100100100008
<input type="checkbox"/> MK009	Teknik Pembasmian Titan	3	100100100009
<input type="checkbox"/> MK010	Pengkodean Cyborg dan AI	4	100100100010

d. Tabel KRS

id_krs	nim	semester	tahun_akademik
1	230441100001	1	2024/2025
2	230441100002	1	2024/2025
3	230441100003	2	2023/2024
4	230441100004	3	2022/2023
5	230441100005	4	2021/2022

e. Tabel detail\_krs

<input type="checkbox"/>	id_detail_krs	id_krs	kode_mk
<input type="checkbox"/>	1	1	MK001
<input type="checkbox"/>	2	1	MK002
<input type="checkbox"/>	3	2	MK001

### 4.3.2 Tugas Praktikum Soal No. 2

1) Tabel customers

customer_id	name	email	phone	address	created_at
1	Sasuke Uchiha	sasuke@konoha.com	08123456789	Desa Konoha, Rumah Uchiha	2025-03-31 19:03:12
2	Megumin	megumin@explosion.com	08234567890	Belakang Guild Adventurer, Axel Town	2025-03-31 19:03:12
3	Levi Ackerman	levi@scoutinglegion.com	08345678901	Dinding Maria, Kamar Kapten Levi	2025-03-31 19:03:12
4	Gintoki Sakata	gintoki@yorozuya.com	08456789012	Habukicho, Edo, Jepang	2025-03-31 19:03:12
5	Aqua	aqua@goddess.com	08567890123	Dimensi Dewa, Tempat Reinkarnasi	2025-03-31 19:03:12
6	Zoro Roronoa	zoro@strawhats.com	08678901234	??? (Dia sendiri juga tidak tahu)	2025-03-31 19:03:12
7	Senku Ishigami	senku@100billionpercent.com	08789012345	Stone World, Jepang Masa Batu	2025-03-31 19:03:12
8	Anya Forger	anya@elegant.com	08890123456	Rumah Keluarga Forger, Nota Berlin	2025-03-31 19:03:12
9	Kaguya Shinomiya	kaguya@shuchiin.com	08901234567	Akademi Shuchiin, Ruang OSIS	2025-03-31 19:03:12
10	Tanjiro Kamado	tanjiro@demoncorp.com	08012345678	Gunung Sagiri, Jepang	2025-03-31 19:03:12

2) Tabel produk

id_produk	name	kategori	harga	stok	deskripsi	created_at
1	Sharingan Bait	Fishing Lures	99.99	10	Menarik perhatian ikan dengan Genjutsu!	39B 2025-03-31 19:03:07
2	Explosion Bait	Fishing Lures	77.77	20	Bait ini hanya bisa digunakan sekali. EXPLOSION!	48B 2025-03-31 19:03:07
3	Ackerman Hook	Hooks	129.99	15	Secepat refleksi Levi!	21B 2025-03-31 19:03:07
4	Zura Rod	Fishing Rods	88.88	25	Bukan Katsura, tapi pancing Zura!	33B 2025-03-31 19:03:07
5	Divine Water Bait	Fishing Lures	55.55	30	Diberkahi oleh Aqua, tapi jangan percaya airnya.	48B 2025-03-31 19:03:07
6	Lost Navigator Compass	Accessories	10.10	50	Dijamin bikin kamu kehilangan arah seperti Zoro.	48B 2025-03-31 19:03:07
7	100 Billion Percent Rod	Fishing Rods	150.00	5	Rod ini dibuat dengan ilmu pengetahuan!	39B 2025-03-31 19:03:07
8	Elegant Hook	Hooks	49.49	35	Bait ini memiliki ELEGANCE, cocok untuk Anya.	45B 2025-03-31 19:03:07
9	Love Is War Net	Nets	120.00	12	Strategi cerdas menangkap ikan, tapi hati-hati jangan jat...	66B 2025-03-31 19:03:07
10	Sun Breathing Line	Fishing Line	200.00	8	Matahari terbit! Jaring yang tak tertandingi.	45B 2025-03-31 19:03:07

3) Tabel orders

order_id	customer_id	order_date	total_price	status
1	1	2025-03-31 19:03:45	99.99	Pending
2	2	2025-03-31 19:03:45	77.77	Shipped
3	3	2025-03-31 19:03:45	129.99	Compl...
4	4	2025-03-31 19:03:45	88.88	Canceled
5	5	2025-03-31 19:03:45	55.55	Pending
6	6	2025-03-31 19:03:45	10.10	Compl...
7	7	2025-03-31 19:03:45	150.00	Pending
8	8	2025-03-31 19:03:45	49.49	Shipped
9	9	2025-03-31 19:03:45	120.00	Pending
10	10	2025-03-31 19:03:45	200.00	Compl...



4) Tabel ordered\_item

order_item_id	order_id	id_produk	banyak_item	price
1	1	1	1	99.99
2	2	2	1	77.77
3	3	3	1	129.99
4	4	4	1	88.88
5	5	5	1	55.55
6	6	6	1	10.10
7	7	7	1	150.00
8	8	8	1	49.49
9	9	9	1	120.00
10	10	10	1	200.00

5) Tabel supplier

id_pemasok	nama_pemasok	phone	email	address
1	Uchiha Fishing Corp	0811223344	uchiha@fishing.com	Konoha, Jepang
2	Explosion Supply Co.	0822334455	megumin@boom.com	Axel Town, Fantasy ...
3	Ackerman Gear Works	0833445566	levi@gears.com	Wall Maria, Paradis...
4	Yorozuya Fishing Tools	0844556677	gintoki@yorozuya.com	Edo, Jepang
5	Aqua Holy Water Inc.	0855667788	aqua@water.com	Dimensi Dewa
6	Lost Navigator Supplies	0866778899	zoro@whereami.com	Somewhere, IDK
7	Ishigami Science Lab	0877889900	senku@science.com	Stone World, Jepang
8	Elegant Gear	0888990011	anya@elegant.com	Berlint, Westalis
9	Shuchiin Fishing Tech	0899001122	kaguya@shuchiin.com	Tokyo, Jepang
10	Demon Slayer Corp	0800112233	tanjiro@demonlayer.com	Jepang, Era Taisho

6) Tabel product\_supplier

	id_produk	id_pemasok
<input type="checkbox"/>	1	1
<input type="checkbox"/>	2	2
<input type="checkbox"/>	3	3
<input type="checkbox"/>	4	4
<input type="checkbox"/>	5	5
<input type="checkbox"/>	6	6
<input type="checkbox"/>	7	7
<input type="checkbox"/>	8	8
<input type="checkbox"/>	9	9
<input type="checkbox"/>	10	10

## 4.4 Penjelasan

### 4.4.1 Tugas Praktikum Soal No. 1

Pada praktikum ini, saya membuat sebuah database bernama akademik yang digunakan untuk menyimpan data terkait sistem akademik, seperti jurusan, mahasiswa, dosen, mata kuliah, serta Kartu Rencana Studi (KRS). Saya memulai dengan membuat tabel jurusan, yang memiliki kolom id\_jurusan sebagai primary key dan nama\_jurusan untuk menyimpan nama jurusan. Selanjutnya, saya membuat tabel mahasiswa dengan nim sebagai primary key, serta kolom id\_jurusan sebagai foreign key yang menghubungkan mahasiswa dengan jurusan mereka. Jika jurusan

dihapus, maka id\_jurusan pada tabel mahasiswa akan menjadi NULL sesuai dengan aturan ON DELETE SET NULL.

Selain itu, saya membuat tabel dosen dengan nip sebagai primary key, serta atribut tambahan seperti email dan telepon. Sama seperti tabel mahasiswa, tabel dosen juga memiliki foreign key id\_jurusan yang menghubungkan dosen dengan jurusan tertentu. Untuk menyimpan data mata kuliah, saya membuat tabel mata\_kuliah dengan kode\_mk sebagai primary key, serta kolom nip sebagai foreign key yang menghubungkan mata kuliah dengan dosen pengampunya. Jika dosen dihapus, maka kolom nip dalam tabel mata kuliah akan menjadi NULL.

Setelah itu, saya membuat tabel KRS (Kartu Rencana Studi) yang mencatat mata kuliah yang diambil mahasiswa dalam satu semester tertentu. Tabel ini memiliki foreign key nim yang terhubung ke tabel mahasiswa dengan aturan ON DELETE CASCADE, sehingga jika seorang mahasiswa dihapus, maka seluruh KRS miliknya juga ikut terhapus. Selain itu, saya juga membuat tabel detail\_krs untuk mencatat mata kuliah yang diambil dalam setiap KRS. Tabel ini memiliki foreign key id\_krs yang terhubung ke tabel krs dan foreign key kode\_mk yang terhubung ke tabel mata\_kuliah.

Setelah struktur tabel selesai dibuat, saya melakukan insert data untuk mengisi tabel dengan informasi terkait jurusan, mahasiswa, dosen, mata kuliah, dan KRS. Data mahasiswa dan dosen yang saya gunakan berasal dari karakter fiksi dalam berbagai anime, yang saya tempatkan dalam jurusan yang sesuai dengan kemampuan mereka. Saya juga menggunakan beberapa perintah tambahan, seperti `RENAME TABLE krs TO tbl_krs` untuk mengubah nama tabel, serta `DROP DATABASE akademik` untuk menghapus seluruh database beserta isinya.

Dengan perancangan ini, saya dapat mengelola data akademik secara efisien, memastikan keterhubungan antar tabel melalui foreign key, serta menerapkan aturan ON DELETE CASCADE dan ON DELETE SET NULL agar data tetap konsisten saat terjadi penghapusan pada tabel yang saling terhubung.

#### **4.4.2 Tugas Praktikum Soal No. 2**

Dalam praktikum ini, saya merancang sebuah database bernama `lure_store` yang digunakan untuk mengelola sistem penjualan alat pancing. Database ini terdiri dari beberapa tabel utama, yaitu `customers` (pelanggan), `products` (produk), `orders`

(pesanan), `order_items` (detail pesanan), `suppliers` (pemasok), dan `product_suppliers` (hubungan produk dengan pemasok).

Saya memulai dengan membuat tabel `customers`, yang menyimpan informasi pelanggan seperti `customer_id` sebagai primary key, `name`, `email` (yang bersifat unik), `phone`, `address`, serta `created_at` yang otomatis mencatat waktu pendaftaran pelanggan menggunakan `CURRENT_TIMESTAMP`. Kemudian, saya membuat tabel `products` untuk menyimpan daftar produk dengan `id_produk` sebagai primary key, serta informasi seperti nama, kategori, harga, stok, deskripsi, dan waktu pembuatan produk.

Selanjutnya, saya merancang tabel `orders` untuk mencatat pesanan pelanggan. Tabel ini memiliki foreign key `customer_id` yang terhubung ke tabel `customers`, dengan aturan `ON DELETE CASCADE` sehingga jika pelanggan dihapus, semua pesanan yang terkait juga akan ikut terhapus. Tabel ini juga memiliki kolom status yang menggunakan tipe data `ENUM` dengan pilihan `Pending`, `Shipped`, `Completed`, dan `Canceled` untuk mencatat status pesanan.

Saya juga membuat tabel `order_items` untuk mencatat detail pesanan, di mana setiap produk dalam pesanan akan dicatat dengan foreign key `order_id` yang terhubung ke tabel `orders` dan `id_produk` yang terhubung ke tabel `products`, keduanya dengan aturan `ON DELETE CASCADE`. Selain itu, saya menambahkan kolom `banyak_item` dengan `CHECK (banyak_item > 0)` untuk memastikan jumlah pesanan selalu lebih dari nol.

Untuk mengelola pemasok, saya membuat tabel `suppliers`, yang menyimpan informasi pemasok seperti `id_pemasok`, `nama_pemasok`, `phone`, `email`, dan `address`. Kemudian, saya membuat tabel `product_suppliers` sebagai tabel relasi antara produk dan pemasok, di mana primary key dari tabel ini adalah kombinasi dari `id_produk` dan `id_pemasok`, dengan keduanya sebagai foreign key yang terhubung ke tabel `products` dan `suppliers` dengan aturan `ON DELETE CASCADE`. Setelah struktur database selesai dibuat, saya menambahkan beberapa data contoh ke dalam setiap tabel, termasuk data pelanggan, produk, pesanan, dan pemasok. Data pelanggan dan produk menggunakan nama karakter fiksi dari berbagai anime, sementara produk dibuat dengan nama yang sesuai dengan karakter tersebut.

Terakhir, saya menjalankan beberapa perintah `SELECT` untuk menampilkan data dari tabel `customers`, `products`, `orders`, `order_items`, `suppliers`, dan `product_suppliers` guna memastikan bahwa semua data telah masuk dengan benar. Dengan perancangan ini, database `lure_store` dapat digunakan untuk mengelola sistem penjualan alat pancing secara efisien, dengan hubungan antar tabel yang jelas dan aturan referensial yang memastikan integritas data tetap terjaga.

## **BAB V**

### **PENUTUP**

#### **5.1 Analisa**

Berdasarkan hasil praktikum yang telah saya lakukan, saya dapat menyimpulkan bahwa Data Definition Language (DDL) dan Data Manipulation Language (DML) memiliki peran yang sangat penting dalam pengelolaan basis data. DDL digunakan untuk mendefinisikan struktur basis data, seperti pembuatan tabel dan pengaturan skema, sedangkan DML berfungsi untuk memanipulasi data dalam tabel, seperti menambahkan, mengubah, dan menghapus data. Dengan memahami kedua konsep ini, saya dapat lebih mudah dalam merancang dan mengelola basis data secara efisien.

Selama praktikum, saya menemukan bahwa perintah DDL bersifat permanen, artinya perubahan yang dilakukan langsung tersimpan dalam sistem dan tidak dapat dibatalkan. Oleh karena itu, dalam implementasi nyata, saya harus lebih berhati-hati dalam menggunakan perintah ini untuk menghindari perubahan yang tidak diinginkan. Sementara itu, perintah DML lebih fleksibel karena memungkinkan saya untuk membatalkan perubahan sebelum dikonfirmasi dengan COMMIT. Hal ini membuat DML lebih cocok untuk operasi harian yang bersifat dinamis, sedangkan DDL lebih sering digunakan dalam tahap awal perancangan basis data.

Selain itu, saya juga menyadari bahwa penggunaan perintah DDL dan DML memiliki dampak yang berbeda terhadap performa sistem. Perintah DDL dapat mempengaruhi keseluruhan struktur basis data, sedangkan perintah DML hanya mempengaruhi data di dalamnya. Oleh karena itu, saya perlu memahami strategi yang tepat dalam penggunaan kedua jenis perintah ini agar sistem basis data tetap berjalan dengan optimal tanpa mengganggu proses lainnya.

Dengan mengikuti praktikum ini, saya merasa lebih memahami bagaimana cara mengelola basis data dengan baik, mulai dari tahap perancangan hingga manipulasi data. Saya yakin bahwa keterampilan ini akan sangat berguna di masa depan, terutama dalam bidang pengembangan perangkat lunak, analisis data, dan manajemen sistem informasi. Oleh karena itu, pengalaman ini menjadi langkah awal yang penting bagi saya dalam mendalami dunia basis data dan sistem informasi secara lebih mendalam.



## 5.2 Kesimpulan

Berdasarkan hasil praktikum yang telah saya lakukan, saya dapat menyimpulkan bahwa Data Definition Language (DDL) dan Data Manipulation Language (DML) merupakan dua elemen penting dalam pengelolaan basis data. DDL digunakan untuk mendefinisikan struktur basis data, sementara DML berfungsi untuk memanipulasi data yang ada. Pemahaman terhadap kedua konsep ini sangat penting dalam merancang dan mengelola sistem basis data secara efisien. Adapun beberapa poin utama yang dapat saya simpulkan dari praktikum ini adalah sebagai berikut:

- 1) DDL digunakan untuk membuat, mengubah, dan menghapus struktur basis data seperti tabel dan skema, dengan perintah utama seperti *CREATE*, *ALTER*, *DROP*, dan *TRUNCATE*.
- 2) DML digunakan untuk memanipulasi data dalam tabel dengan perintah seperti *INSERT*, *SELECT*, *UPDATE*, dan *DELETE*.
- 3) Penggunaan DDL dan DML harus disesuaikan dengan kebutuhan, di mana DDL lebih sering digunakan pada tahap awal perancangan basis data, sedangkan DML digunakan dalam operasi harian yang bersifat dinamis.
- 4) Pemahaman dan penerapan yang baik terhadap DDL dan DML akan membantu dalam mengelola basis data secara lebih efektif, meningkatkan efisiensi sistem, serta mendukung pengolahan data dalam berbagai aplikasi dan sistem informasi.