

MODUL PRAKTIKUM BASIS DATA

Pertemuan 5: Data Manipulation Language (DML)

1. Capaian Pembelajaran Mata Kuliah (CPMK)

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. Memahami konsep dasar Data Manipulation Language (DML).
2. Menggunakan perintah DML untuk mengelola data dalam database.
3. Melakukan operasi **INSERT**, **UPDATE**, **DELETE**, dan **SELECT** pada tabel.
4. Mengelola relasi sederhana antar tabel menggunakan query DML.

2. Tujuan Praktikum

Mahasiswa mampu:

- Menambahkan data ke dalam tabel (INSERT).
- Menampilkan data dari tabel (SELECT).
- Mengubah data dalam tabel (UPDATE).
- Menghapus data dari tabel (DELETE).
- Menggabungkan data dari beberapa tabel sederhana.

3. Dasar Teori

3.1 Pengertian DML

Data Manipulation Language (DML) adalah kumpulan perintah SQL yang digunakan untuk **mengelola isi data** dalam database.

3.2 Jenis Perintah DML

1. SELECT

Digunakan untuk mengambil data dari tabel.

```
SELECT * FROM mahasiswa;
```

2. INSERT

Digunakan untuk menambahkan data baru ke tabel.

```
INSERT INTO mahasiswa (nim, nama, id_fakultas)
VALUES ('22001', 'Andi', 'F01');
```

3. UPDATE

Digunakan untuk mengubah data yang sudah ada.

```
UPDATE mahasiswa
SET nama = 'Andi Saputra'
WHERE nim = '22001';
```

4. DELETE

Digunakan untuk menghapus data dari tabel.

```
DELETE FROM mahasiswa
WHERE nim = '22001';
```

3.3 Relasi Antar Tabel (JOIN)

Relasi antara tabel dapat digunakan dalam query:

```
SELECT m.nama, f.nama_fakultas
FROM mahasiswa m
JOIN fakultas f ON m.id_fakultas = f.id_fakultas;
```

4. Percobaan Praktikum

4.1 Persiapan Tabel

Buka pgAdmin

- Jalankan aplikasi pgAdmin.
- Pilih database akademik_db yang sudah dibuat praktikum sebelumnya
- Lalu klik kanan pada database pilih QueryTool

Jalankan perintah berikut terlebih dahulu untuk membuat tabel fakultas dan tabel mahasiswa:

```
CREATE TABLE fakultas (
    id_fakultas VARCHAR(5) PRIMARY KEY,
    nama_fakultas VARCHAR(50)
);
```

```
CREATE TABLE mahasiswa (
  nim VARCHAR(10) PRIMARY KEY,
  nama VARCHAR(50),
  id_fakultas VARCHAR(5),
  FOREIGN KEY (id_fakultas) REFERENCES fakultas(id_fakultas)
);
```

```
1 CREATE TABLE fakultas (
2   id_fakultas VARCHAR(5) PRIMARY KEY,
3   nama_fakultas VARCHAR(50)
4 );
5 CREATE TABLE mahasiswa (
6   nim VARCHAR(10) PRIMARY KEY,
7   nama VARCHAR(50),
8   id_fakultas VARCHAR(5),
9   FOREIGN KEY (id_fakultas) REFERENCES fakultas(id_fakultas)
10 );
11
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 93 msec.

➤ Melihat isi tabel apa saja yang sudah dibuat:

```
1 SELECT table_name FROM information_schema.tables WHERE table_schema='public';
```

Data Output Messages Notifications

Showing rows: 1 to 2 Page No: 1

table_name
1 fakultas
2 mahasiswa

4.2 Percobaan INSERT

➤ Tambahkan data ke tabel **fakultas**:

```
INSERT INTO fakultas VALUES ('F01', 'Teknik');
INSERT INTO fakultas VALUES ('F02', 'Ekonomi');
INSERT INTO fakultas VALUES ('F03', 'Ilmu Komputer');
```

```

1  INSERT INTO fakultas VALUES ('F01', 'Teknik');
2  INSERT INTO fakultas VALUES ('F02', 'Ekonomi');
3  INSERT INTO fakultas VALUES ('F03', 'Ilmu Komputer');
4

```

Data Output [Messages](#) Notifications

INSERT 0 1

Query returned successfully in 100 msec.

➤ Tambahkan data ke tabel **mahasiswa**:

```

INSERT INTO mahasiswa VALUES ('22001', 'Budi', 'F01');
INSERT INTO mahasiswa VALUES ('22002', 'Siti', 'F02');
INSERT INTO mahasiswa VALUES ('22003', 'Andi', 'F03');

```

```

1  INSERT INTO mahasiswa VALUES ('22001', 'Budi', 'F01');
2  INSERT INTO mahasiswa VALUES ('22002', 'Siti', 'F02');
3  INSERT INTO mahasiswa VALUES ('22003', 'Andi', 'F03');
4

```

Data Output [Messages](#) Notifications

INSERT 0 1

Query returned successfully in 144 msec.

4.3 Percobaan SELECT

➤ Menampilkan semua data mahasiswa dan fakultas:

```
SELECT * FROM mahasiswa;
```

```

1  SELECT * FROM mahasiswa;
2

```

Data Output [Messages](#) Notifications

	nim [PK] character varying (10)	nama character varying (50)	id_fakultas character varying (5)
1	22001	Budi	F01
2	22002	Siti	F02
3	22003	Andi	F03

```
SELECT * FROM fakultas;
```

```
1 SELECT * FROM fakultas;
2
```

Data Output Messages Notifications

	id_fakultas [PK] character varying (5)	nama_fakultas character varying (50)
1	F01	Teknik
2	F02	Ekonomi
3	F03	Ilmu Komputer

- Menampilkan data tertentu (data nama dari tabel mahasiswa):

```
SELECT nama FROM mahasiswa;
```

```
1 SELECT nama FROM mahasiswa;
2
```

Data Output Messages Notifications

	nama character varying (50)
1	Budi
2	Siti
3	Andi

- Menampilkan dengan kondisi (menampilkan data mahasiswa yang memiliki ide_fakultas = F01:

```
SELECT * FROM mahasiswa
WHERE id_fakultas = 'F01';
```

```

1 SELECT * FROM mahasiswa
2 WHERE id_fakultas = 'F01';
3
4

```

Data Output Messages Notifications

Showing rows: 1

	nim [PK] character varying (10)	nama character varying (50)	id_fakultas character varying (5)
1	22001	Budi	F01

4.4 Percobaan UPDATE

Data awal:

	nim [PK] character varying (10)	nama character varying (50)	id_fakultas character varying (5)
1	22001	Budi	F01
2	22002	Siti	F02
3	22003	Andi	F03

➤ Mengubah nama mahasiswa:

```

UPDATE mahasiswa
SET nama = 'Budi Santoso'
WHERE nim = '22001';

```

```

1 UPDATE mahasiswa
2 SET nama = 'Budi Santoso'
3 WHERE nim = '22001';
4

```

Data Output Messages Notifications

UPDATE 1

Query returned successfully in 105 msec.

➤ Datanya berubah menjadi:

```

1 SELECT *FROM mahasiswa;
2
3

```

Data Output Messages Notifications

Showing rows: 1 to 3

	nim [PK] character varying (10)	nama character varying (50)	id_fakultas character varying (5)
1	22002	Siti	F02
2	22003	Andi	F03
3	22001	Budi Santoso	F01

4.5 Percobaan DELETE

- Menghapus data mahasiswa(menghapus data mahasiswa yang memiliki nim=22003):

```

DELETE FROM mahasiswa
WHERE nim = '22003';

```

```

1 DELETE FROM mahasiswa
2 WHERE nim = '22003';
3
4

```

Data Output Messages Notifications

DELETE 1

Query returned successfully in 142 msec.

```

1 SELECT *FROM mahasiswa;
2
3
4

```

Data Output Messages Notifications

Showing rows: 1 to 2

	nim [PK] character varying (10)	nama character varying (50)	id_fakultas character varying (5)
1	22002	Siti	F02
2	22001	Budi Santoso	F01

4.6 Percobaan JOIN

- Menggabungkan data (menampilkan data dari tabel yang berbeda):

```
SELECT m.nim, m.nama, f.nama_fakultas
FROM mahasiswa m
JOIN fakultas f ON m.id_fakultas = f.id_fakultas;
```

```
1 SELECT m.nim, m.nama, f.nama_fakultas
2 FROM mahasiswa m
3 JOIN fakultas f ON m.id_fakultas = f.id_fakultas;
4
```

Data Output Messages Notifications

Showing rows: 1

	nim character varying (10)	nama character varying (50)	nama_fakultas character varying (50)
1	22002	Siti	Ekonomi
2	22001	Budi Santoso	Teknik

5. Latihan Praktikum

➤ Untuk soal latihan 1 – latihan 6 sesuai tabel di percobaan

Latihan 1 (INSERT)

Tambahkan data berikut:

- Fakultas:
 - F04 → Hukum
- Mahasiswa:
 - 22004 → Rina → F04
 - 22005 → Dodi → F01

Latihan 2 (SELECT)

1. Tampilkan semua data mahasiswa.
2. Tampilkan nama mahasiswa dari fakultas Teknik.
3. Tampilkan mahasiswa yang NIM-nya diawali “22”.

Latihan 3 (UPDATE)

1. Ubah nama “Siti” menjadi “Siti Aminah”.
2. Ubah fakultas Dodi menjadi Ilmu Komputer.

Latihan 4 (DELETE)

1. Hapus data mahasiswa dengan NIM 22005.
2. Hapus semua mahasiswa dari fakultas Ekonomi.

Latihan 5 (JOIN)

1. Tampilkan daftar mahasiswa beserta nama fakultasnya.
2. Tampilkan hanya mahasiswa dari fakultas Ilmu Komputer.

Latihan 6

Buat query untuk:

1. Menghitung jumlah mahasiswa per fakultas.
2. Menampilkan fakultas yang memiliki lebih dari 1 mahasiswa.

Latihan 7

Studi Kasus: Sistem Penjualan Toko Buku

Sebuah toko buku mencatat data pelanggan dan transaksi pembelian buku. Setiap pelanggan bisa melakukan lebih dari satu transaksi.

✓ Tabel pelanggan

Kolom	Keterangan
id_pelanggan	ID pelanggan
nama_pelanggan	Nama pelanggan
kota	Kota asal

✓ Tabel transaksi

Kolom	Keterangan
id_transaksi	ID transaksi
id_pelanggan	ID pelanggan
nama_buku	Nama buku
jumlah	Jumlah buku
total_harga	Total harga

- a. Tambahkan data pelanggan berikut:

- ID 1 – Andi – Surabaya
 - ID 2 – Budi – Malang
 - ID 3 – Citra – Surabaya
- b. Tambahkan data transaksi:
- Transaksi 101: Andi membeli buku Basis Data, jumlah 2, total harga 120000
 - Transaksi 102: Budi membeli buku Algoritma, jumlah 1, total harga 75000
 - Transaksi 103: Andi membeli buku Pemrograman Web, jumlah 1, total harga 90000
- c. Tampilkan seluruh data pelanggan
- d. Tampilkan seluruh data transaksi
- e. Tampilkan nama pelanggan, nama buku, dan total harga (gunakan JOIN)
- f. Tampilkan transaksi yang dilakukan oleh pelanggan asal Surabaya
- g. Tampilkan pelanggan yang membeli buku Basis Data
- h. Kota pelanggan bernama Budi awalnya salah input. Ubah kotanya menjadi Blitar
- i. Harga buku Pemrograman Web berubah. Update total harga menjadi 95000
- j. Terdapat transaksi yang dibatalkan dengan ID transaksi 102. Hapus transaksi tersebut.

7. Kesimpulan

Pada praktikum ini mahasiswa telah mempelajari:

- Cara menambahkan, mengubah, menghapus, dan menampilkan data.
- Penggunaan query DML dalam pengelolaan database.
- Penggabungan data dari beberapa tabel menggunakan JOIN.