

LATIHAN MODUL 3
LAPORAN PRAKTIKUM BASIS
DATA Data Manipulation Language
(DML)



Disusun oleh:

Nama: Antonius Munthe
NIM: 121140032
Kelas: Basis Data RA

PROGRAM STUDI TEKNIK
INFORMATIKA INSTITUT TEKNOLOGI
SUMATERA LAMPUNG SELATAN

2022

BAB 1

TEORI DASAR

A. Data Manipulation Language (DML)

Data Manipulation Language (DML) adalah bahasa pemrograman khusus yang digunakan untuk memanipulasi atau mengelola data dalam database. DML digunakan untuk menambah, mengubah, menghapus, atau memperbarui data yang tersimpan dalam sebuah database. Dalam konteks sistem manajemen basis data (DBMS), DML memungkinkan pengguna untuk mengambil atau memanipulasi data yang disimpan dalam database. Dalam DML, perintah SQL (Structured Query Language) digunakan untuk memanipulasi data. SQL adalah bahasa pemrograman yang digunakan untuk mengelola basis data relasional.

Contoh perintah DML dalam SQL adalah SELECT, INSERT, UPDATE, DELETE, dan sebagainya. Perintah-perintah ini memungkinkan pengguna untuk mengambil data dari tabel, menambahkan data baru ke dalam tabel, mengubah data yang sudah ada dalam tabel, atau menghapus data dari tabel. Dengan menggunakan DML, pengguna dapat mengatur data dalam database mereka, melakukan tindakan pengelolaan data, dan memastikan bahwa data yang disimpan dalam database tetap akurat dan terkini.

B. Perintah Dasar DML

- SELECT : Perintah ini digunakan untuk mengambil baris dari tabel
- UPDATE : Perintah ini memodifikasi data dari satu atau lebih record.
- INSERT : Perintah ini menambahkan satu atau lebih record ke tabel database.
- DELETE : Perintah ini menghapus satu atau lebih record dari tabel sesuai dengan kondisi yang ditentukan.

Ada juga beberapa fungsi lain seperti:

>> Aggregate

- A. SUM digunakan untuk menghitung total nominal data
- B. COUNT digunakan untuk menghitung jumlah kemunculan data
- C. AVG digunakan untuk menghitung rata-rata sekelompok data.
- D. MAX dan MIN untuk mendapatkan nilai maksimum atau minimum dari sekelompok data

>> Comparison Operator

- A. AS digunakan untuk menampilkan tabel atau kolom menggunakan alias.
- B. AND digunakan untuk semua kondisi AND digunakan untuk semua kondisi yang dipisahkan oleh AND bernilai TRUE
- C. OR digunakan untuk menampilkan data yang memenuhi semua syarat atau satu persatu syarat
- D. BETWEEN digunakan untuk menentukan lebar data yang ingin ditampilkan. Data tersebut ditampilkan dari data yang rendah sampai data yang tinggi
- E. IN digunakan untuk pencocokan pada suatu daftar nilai
- F. LIKE digunakan untuk pencarian sebuah data dengan diwakili karakter tertentu
- G. LIMIT digunakan untuk membatasi data yang ingin ditampilkan.

BAB 2

PEMBAHASAN DAN ANALISIS

1. Masukkan data berikut ke dalam tabel "rekening".

no_rekening	kode_cabang	pin	saldo
120023	JGJ01	837242	500000
120087	JGJ02	112233	2150000
140932	JOJ02	024422	1200000
220983	SMG12	732312	5422000
332102	JKT48	228883	400500

```
MariaDB [bank_mino]> insert into rekening (no_rekening,kode_cabang,pin,saldo) values(120087, "JGJ02"
, 112233, 2150000);
Query OK, 1 row affected (0.004 sec)
```

```
MariaDB [bank_mino]> insert into rekening (no_rekening,kode_cabang,pin,saldo) values(140932, "JOJ01"
, 024422, 1200000);
Query OK, 1 row affected (0.003 sec)
```

```
MariaDB [bank_mino]> insert into rekening (no_rekening,kode_cabang,pin,saldo) values(220983, "SMG12"
, 732312, 5422000);
Query OK, 1 row affected (0.003 sec)
```

```
MariaDB [bank_mino]> insert into rekening (no_rekening,kode_cabang,pin,saldo) values(220983, "SMG12"
, 732312, 5422000);
insert into rekening (no_rekening,kode_cabang,pin,saldo) values(332102, "JKT48"
, 228883, 400500);
Query OK, 1 row affected (0.004 sec)
```

```
MariaDB [bank_mino]> select * from rekening;
+-----+-----+-----+-----+
| no_rekening | kode_cabang | pin   | saldo |
+-----+-----+-----+-----+
| 120023      | JGJ01       | 837242 | 500000 |
| 120087      | JGJ02       | 112233 | 2150000 |
| 140932      | JOJ01       | 024422 | 1200000 |
| 220983      | SMG12       | 732312 | 5422000 |
| 332102      | JKT48       | 228883 | 400500 |
+-----+-----+-----+-----+
```

2. Masukkan data berikut ke dalam tabel "transaksi".

no_transaksi	no_rekening	jenis_transaksi	tgl_transaksi	jumlah
1	220983	DEBET	2014-06-12 12:00:45	500000
2	220983	DEBET	2014-06-13 08:20:33	2150000
3	140932	ATM	2014-06-14	100000

Siswa mengisi data di bawah ini. Latihan praktikum dikerjakan secara individu

			10:56:52	
4	140932	ATM	2014-06-14 15:16:23	100000
5	120087	DEBET	2014-06-15 15:03:33	1250000
6	220983	ATM	2014-06-15 15:32:15	150000
7	220983	ATM	2014-06-15 15:40:12	100000
8	220983	DEBET	2014-06-16 08:23:23	2500000

```

MariaDB [bank_mino]> INSERT INTO transaksi (no_transaksi, no_rekening, jenis_transaksi, tgl_transaksi, jumlah) VALUES (5, 120087, "DEBET", "2014-06-15 15:03:33", 1250000);
Query OK, 1 row affected (0.008 sec)

MariaDB [bank_mino]> INSERT INTO transaksi (no_transaksi, no_rekening, jenis_transaksi, tgl_transaksi, jumlah) VALUES (6, 220983, "ATM", "2014-06-15 15:32:15", 150000);
Query OK, 1 row affected (0.007 sec)

MariaDB [bank_mino]> INSERT INTO transaksi (no_transaksi, no_rekening, jenis_transaksi, tgl_transaksi, jumlah) VALUES (7, 220983, "ATM", "2014-06-15 15:40:12", 100000);
Query OK, 1 row affected (0.008 sec)

MariaDB [bank_mino]> INSERT INTO transaksi (no_transaksi, no_rekening, jenis_transaksi, tgl_transaksi, jumlah) VALUES (8, 220983, "DEBET", "2014-06-16 08:23:23", 2500000);
Query OK, 1 row affected (0.008 sec)

MariaDB [bank_mino]> select * from transaksi;
+-----+-----+-----+-----+-----+
| no_transaksi | no_rekening | jenis_transaksi | tgl_transaksi | jumlah |
+-----+-----+-----+-----+-----+
| 1 | 220983 | DEBET | 2014-06-12 12:00:45 | 500000 |
| 2 | 220983 | DEBET | 2014-06-13 08:20:33 | 2150000 |
| 3 | 140932 | ATM | 2014-06-14 10:56:52 | 100000 |
| 4 | 140932 | ATM | 2014-06-14 15:16:23 | 100000 |
| 5 | 120087 | DEBET | 2014-06-15 15:03:33 | 1250000 |
| 6 | 220983 | ATM | 2014-06-15 15:32:15 | 150000 |
| 7 | 220983 | ATM | 2014-06-15 15:40:12 | 100000 |
| 8 | 220983 | DEBET | 2014-06-16 08:23:23 | 2500000 |
+-----+-----+-----+-----+-----+

```

3. Ubah nilai pin menjadi “450817” dari rekening dengan no rekening “220983”

```

MariaDB [bank_mino]> update rekening set pin = 450817 where no_rekening = 220983;
Query OK, 0 rows affected (0.036 sec)

```

4. Ubah tanggal_transaksi menjadi 2014-06-17 20:34:22 dari no transaksi adalah “6”.

```

MariaDB [bank_mino]> update transaksi set tgl_transaksi = "2014-06-17 20:34:22" where
-> no_transaksi = 6;
Query OK, 0 rows affected (0.010 sec)
Rows matched: 1 Changed: 0 Warnings: 0

```

5. Tampilkan semua data dari tabel transaksi.

```

MariaDB [bank_mino]> select * from transaksi;
+-----+-----+-----+-----+-----+
| no_transaksi | no_rekening | jenis_transaksi | tgl_transaksi | jumlah |
+-----+-----+-----+-----+-----+
| 1 | 220983 | DEBET | 2014-06-12 12:00:45 | 500000 |
| 2 | 220983 | DEBET | 2014-06-13 08:20:33 | 2150000 |
| 3 | 140932 | ATM | 2014-06-14 10:56:52 | 100000 |
| 4 | 140932 | ATM | 2014-06-14 15:16:23 | 100000 |
| 5 | 120087 | DEBET | 2014-06-15 15:03:33 | 1250000 |
| 6 | 220983 | ATM | 2014-06-17 20:34:22 | 150000 |
| 7 | 220983 | ATM | 2014-06-15 15:40:12 | 100000 |
| 8 | 220983 | DEBET | 2014-06-16 08:23:23 | 2500000 |
+-----+-----+-----+-----+-----+
8 rows in set (0.000 sec)

```

6. Tampilkan 5 transaksi terakhir data dari tabel transaksi

```

MariaDB [bank_mino]> select * from transaksi order by no_transaksi
-> desc limit 5;
+-----+-----+-----+-----+-----+
| no_transaksi | no_rekening | jenis_transaksi | tgl_transaksi | jumlah |
+-----+-----+-----+-----+-----+
| 8 | 220983 | DEBET | 2014-06-16 08:23:23 | 2500000 |
| 7 | 220983 | ATM | 2014-06-15 15:40:12 | 100000 |
| 6 | 220983 | ATM | 2014-06-17 20:34:22 | 150000 |
| 5 | 120087 | DEBET | 2014-06-15 15:03:33 | 1250000 |
| 4 | 140932 | ATM | 2014-06-14 15:16:23 | 100000 |
+-----+-----+-----+-----+-----+
5 rows in set (0.011 sec)

```

7. Tampilkan data transaksi pada tanggal 2014-06-15.

```
MariaDB [bank_mino]> select * from transaksi where tgl_transaksi like "2014-06-15%";
+-----+-----+-----+-----+-----+
| no_transaksi | no_rekening | jenis_transaksi | tgl_transaksi | jumlah |
+-----+-----+-----+-----+-----+
|          5 |      120087 | DEBET          | 2014-06-15 15:03:33 | 1250000 |
|          7 |      220983 | ATM            | 2014-06-15 15:40:12 | 100000 |
+-----+-----+-----+-----+-----+
2 rows in set (0.007 sec)
```

8. Tampilkan data transaksi pada tanggal 2014-06-14 dan 2014-06-15.

```
MariaDB [bank_mino]> select * from
-> transaksi where tgl_transaksi between " 2014-06-14 00:00:00" and "2014-06-15 23:59:59";
+-----+-----+-----+-----+-----+
| no_transaksi | no_rekening | jenis_transaksi | tgl_transaksi | jumlah |
+-----+-----+-----+-----+-----+
|          3 |      140932 | ATM            | 2014-06-14 10:56:52 | 100000 |
|          4 |      140932 | ATM            | 2014-06-14 15:16:23 | 100000 |
|          5 |      120087 | DEBET          | 2014-06-15 15:03:33 | 1250000 |
|          7 |      220983 | ATM            | 2014-06-15 15:40:12 | 100000 |
+-----+-----+-----+-----+-----+
4 rows in set (0.009 sec)
```

9. Tampilkan transaksi terakhir dari pada tanggal 2014-06-15.

```
MariaDB [bank_mino]> select * from transaksi where tgl_transaksi like "2014-06-15%" order by tgl_transaksi desc limit 1;
+-----+-----+-----+-----+-----+
| no_transaksi | no_rekening | jenis_transaksi | tgl_transaksi | jumlah |
+-----+-----+-----+-----+-----+
|          7 |      220983 | ATM            | 2014-06-15 15:40:12 | 100000 |
+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

10. Tampilkan seluruh data pada tabel rekening yang berasal dari Jogja.

```
MariaDB [bank_mino]> select * from rekening where kode_cabang like "JGJ%";
+-----+-----+-----+-----+
| no_rekening | kode_cabang | pin    | saldo |
+-----+-----+-----+-----+
|      120023 | JGJ01      | 837242 | 500000 |
|      120087 | JGJ02      | 112233 | 2150000 |
+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

11. Hapus data pada rekening yang memiliki no rekening "120087".

```
MariaDB [bank_mino]> delete from rekening where no_rekening = "120087";
Query OK, 1 row affected (0.012 sec)
```

12. Tampilkan kode no_rekening dan kode_cabang dari nasabah yang memiliki saldo antara 500000 dan 3000000.

```
MariaDB [bank_mino]> select no_rekening, kode_cabang from rekening where saldo between 500000 and 3000000;
+-----+-----+
| no_rekening | kode_cabang |
+-----+-----+
|      120023 | JGJ01      |
|      140932 | JOJ01      |
+-----+-----+
2 rows in set (0.001 sec)
```

13. Tampilkan total transaksi yang terjadi pada tanggal 15 Juni 2014.

```
MariaDB [bank_mino]> select sum(jumlah) as total_transaksi from transaksi where tgl_transaksi like "2014-06-15%";
+-----+
| total_transaksi |
+-----+
|          1350000 |
+-----+
1 row in set (0.012 sec)
```

14. Tampilkan seluruh dana transaksi yang pernah dilakukan oleh no_rekening "220983".

```
MariaDB [bank_mino]> select jumlah from transaksi where no_rekening = "220983";
```

jumlah
500000
2150000
150000
100000
2500000

```
5 rows in set (0.009 sec)
```

15. Tampilkan rata-rata dana transaksi yang terjadi antara tanggal 13 Juni 2014 s.d 15 Juni 2014.

```
MariaDB [bank_mino]> select avg(jumlah) as ratarata_dana from transaksi where tgl_transaksi between "2014-06-13 00:00:00" and "2014-06-15 23:59:59";
```

ratarata_dana
740000.0000

```
1 row in set (0.001 sec)
```

16. Tampilkan banyaknya transaksi yang dilakukan oleh masing-masing pemilik rekening.

```
MariaDB [bank_mino]> select count(no_rekening), no_rekening as banyak_transaksi from transaksi group by no_rekening;
```

count(no_rekening)	banyak_transaksi
1	120087
2	140932
5	220983

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

17. Tampilkan seluruh transaksi pada bulan Juni 2014.

```
MariaDB [bank_mino]> select * from transaksi where tgl_transaksi like "2014-06%";
```

no_transaksi	no_rekening	jenis_transaksi	tgl_transaksi	jumlah
1	220983	DEBET	2014-06-12 12:00:45	500000
2	220983	DEBET	2014-06-13 08:20:33	2150000
3	140932	ATM	2014-06-14 10:56:52	100000
4	140932	ATM	2014-06-14 15:16:23	100000
5	120087	DEBET	2014-06-15 15:03:33	1250000
6	220983	ATM	2014-06-17 20:34:22	150000
7	220983	ATM	2014-06-15 15:40:12	100000
8	220983	DEBET	2014-06-16 08:23:23	2500000

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

18. Tampilkan seluruh transaksi yang dilakukan oleh masing-masing pemilik rekening.

```
MariaDB [bank_mino]> select * from transaksi where no_rekening = 220983; dan ' select * from transaksi where no_rekening = 140932;
```

no_transaksi	no_rekening	jenis_transaksi	tgl_transaksi	jumlah
1	220983	DEBET	2014-06-12 12:00:45	500000
2	220983	DEBET	2014-06-13 08:20:33	2150000
6	220983	ATM	2014-06-17 20:34:22	150000
7	220983	ATM	2014-06-15 15:40:12	100000
8	220983	DEBET	2014-06-16 08:23:23	2500000

```
5 rows in set (0.008 sec)
```

BAB III

KESIMPULAN

Berdasarkan modul DML, dapat disimpulkan bahwa Data Manipulation Language (DML) merupakan bahasa pemrograman khusus yang digunakan untuk memanipulasi atau mengelola data dalam database. DML digunakan untuk menambah, mengubah, menghapus, atau memperbarui data yang tersimpan dalam sebuah database.

Dalam konteks sistem manajemen basis data (DBMS), DML memungkinkan pengguna untuk mengambil atau memanipulasi data yang disimpan dalam database. Dalam DML, perintah SQL (Structured Query Language) digunakan untuk memanipulasi data. SQL adalah bahasa pemrograman yang digunakan untuk mengelola basis data relasional.

Beberapa perintah dasar DML yang biasanya digunakan dalam SQL untuk memanipulasi data dalam sebuah database antara lain SELECT, INSERT, UPDATE, DELETE. Dengan menggunakan DML, pengguna dapat mengatur data dalam database mereka, melakukan tindakan pengelolaan data, dan memastikan bahwa data yang disimpan dalam database tetap akurat dan terkini. Oleh karena itu, DML merupakan salah satu aspek penting dalam manajemen basis data yang perlu dipahami oleh para pengguna basis data.

DAFTAR PUSTAKA

- https://www.canva.com/design/DAFepCIPKIQ/4soHTjoFBpzYrpifrDZZPA/edit?utm_content=DAFepCIPKIQ&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton
- Modul Praktikum Basis Data Pertemuan 3