

LAPORAN PRAKTIKUM BASIS DATA

Disusun Untuk Memenuhi Tugas Mata Kuliah Praktikum Basis Data

Dosen Pengampu : Ridwan Setiawan, S.T., M.Kom.



Disusun oleh :

Siti Saadah Rahmawati

2206154

Informatika-E

JURUSAN ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI GARUT
2024

1. Pada gambar dibawah adalah perintah untuk membuat tabel transaksi dengan struktur tabel seperti pada gambar dibawah.

```
MariaDB [onlineshopku_e_sitisaadahr]> create table transaksi(
-> id_trans varchar(3) not null primary key,
-> nama_pelanggan varchar(15),
-> total_harga int(11));
Query OK, 0 rows affected (0.086 sec)
```

2. Kemudian, pada gambar dibawah adalah perintah untuk menampilkan tabel beserta struktur tabel transaksi yang telah dibuat.

```
MariaDB [onlineshopku_e_sitisaadahr]> desc transaksi;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_trans | varchar(3) | NO | PRI | NULL | |
| nama_pelanggan | varchar(15) | YES | | NULL | |
| total_harga | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.071 sec)
```

3. Selanjutnya, pada gambar dibawah merupakan perintah untuk menambahkan data pada tabel transaksi.

```
MariaDB [onlineshopku_e_sitisaadahr]> insert into transaksi(id_trans, nama_pelanggan, total_harga) values
-> ('001', 'Budi', '20000'),
-> ('002', 'Jajang', '5000'),
-> ('003', 'Jajang', '10000'),
-> ('004', 'Udin', '25000'),
-> ('005', 'Budi', '10000'),
-> ('006', 'Budi', '14000'),
-> ('007', 'Asep', '30000');
Query OK, 7 rows affected (0.108 sec)
Records: 7 Duplicates: 0 Warnings: 0

MariaDB [onlineshopku_e_sitisaadahr]> select * from transaksi;
+-----+-----+-----+
| id_trans | nama_pelanggan | total_harga |
+-----+-----+-----+
| 001 | Budi | 20000 |
| 002 | Jajang | 5000 |
| 003 | Jajang | 10000 |
| 004 | Udin | 25000 |
| 005 | Budi | 10000 |
| 006 | Budi | 14000 |
| 007 | Asep | 30000 |
+-----+-----+-----+
7 rows in set (0.101 sec)
```

4. Kemudian, pada gambar dibawah adalah fungsi aggregate yang digunakan untuk mengambil data tunggal hasil dari perhitungan data yang tersimpan dalam suatu kolom.

```
MariaDB [onlineshopku_e_sitisaadahr]> select count(nama_pelanggan) as jumlah_baris from transaksi;
+-----+
| jumlah_baris |
+-----+
| 7 |
+-----+
1 row in set (0.008 sec)
```

5. Selanjutnya, pada gambar dibawah adalah penggunaan perintah GROUP BY yang hanya bisa digunakan dalam Query yang mengandung paling sedikit satu fungsi Aggregate. Jika tidak ada fungsi aggregate tidak perlu menggunakan GROUP BY. Gambar dibawah Menampilkan data pelanggan dan berapa banyak transaksi pembelian barang yang telah dilakukannya.

```
MariaDB [onlineshopku_e_sitisaadahr]> select
-> nama_pelanggan,
-> count(nama_pelanggan) 'banyak_pembelian'
-> from transaksi group by nama_pelanggan;
```

nama_pelanggan	banyak_pembelian
Asep	1
Budi	3
Jajang	2
Udin	1

4 rows in set (0.005 sec)

6. Kemudian, pada gambar dibawah perintah GROUP BY ditambah dengan perintah ORDER BY yang digunakan untuk mengurutkan hasil query. ORDER BY tidak dapat digunakan pada query yang hanya mengandung fungsi aggregate atau tanpa ada GROUP BY. Dan group by harus ditulis sebelum order by seperti pada gambar dibawah.

```
MariaDB [onlineshopku_e_sitisaadahr]> select
-> nama_pelanggan,
-> count(nama_pelanggan) 'banyak_pembelian'
-> from transaksi
-> group by nama_pelanggan
-> order by banyak_pembelian desc;
```

nama_pelanggan	banyak_pembelian
Budi	3
Jajang	2
Udin	1
Asep	1

4 rows in set (0.022 sec)

7. Selanjutnya, pada gambar dibawah terdapat fungsi having yang mempunyai fungsi dan sintak yang sama dengan WHERE.

```

MariaDB [onlineshopku_e_sitisaadahr]> select
-> nama_pelanggan,
-> count(nama_pelanggan) 'banyak_pembelian'
-> from transaksi
-> group by nama_pelanggan
-> having banyak_pembelian >=2
-> order by banyak_pembelian desc;
+-----+-----+
| nama_pelanggan | banyak_pembelian |
+-----+-----+
| Budi           | 3                |
| Jajang         | 2                |
+-----+-----+
2 rows in set (0.002 sec)

```

8. Pada gambar dibawah adalah perintah untuk membuat tabel produk dan menampilkan hasil tabel produk yang telah dibuat. Tabel ini dibuat untuk menyimpan informasi produk yang memiliki field kode_produk, nama_produk dan harga.

```

MariaDB [onlineshopku_e_sitisaadahr]> create table produk(
-> kode_produk varchar(6) not null primary key,
-> nama_produk varchar(100) not null,
-> harga int(11) not null);
Query OK, 0 rows affected (0.093 sec)

MariaDB [onlineshopku_e_sitisaadahr]> describe produk;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| kode_produk | varchar(6) | NO   | PRI | NULL    |       |
| nama_produk | varchar(100) | NO   |     | NULL    |       |
| harga       | int(11)    | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.048 sec)

```

9. Pada gambar dibawah merupakan perintah untuk membuat tabel log harga produk dan menampilkan hasil tabel yang telah dibuat. Tabel ini dibuat untuk menyimpan informasi perubahan harga produk, informasi yang akan kita simpan adalah kode_produk, harga_lama, harga_baru dan waktu_perubahan.

```

MariaDB [onlineshopku_e_sitisaadahr]> create table log_harga_produk(
-> log_id int(11) not null primary key auto_increment,
-> kode_produk varchar(8) not null,
-> harga_lama int(11) not null,
-> harga_baru int(11) not null,
-> waktu_perubahan datetime not null);
Query OK, 0 rows affected (0.039 sec)

MariaDB [onlineshopku_e_sitisaadahr]> describe log_harga_produk;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| log_id     | int(11)   | NO   | PRI | NULL    | auto_increment |
| kode_produk | varchar(8) | NO   |     | NULL    |              |
| harga_lama  | int(11)   | NO   |     | NULL    |              |
| harga_baru  | int(11)   | NO   |     | NULL    |              |
| waktu_perubahan | datetime | NO   |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.056 sec)

```

10. Kemudian, pada gambar dibawah adalah perintah untuk membuat sebuah trigger untuk mencatat perubahan harga produk ketika sebuah record produk di update. Jadi karena trigger ini merupakan kode prosedural yang otomatis dijalankan untuk menanggapi perubahan tertentu, maka untuk mencatat perubahan harga produk menggunakan trigger.

```
MariaDB [onlineshopku_e_sitisaadahr]> delimiter $$
MariaDB [onlineshopku_e_sitisaadahr]> create trigger before_produk_update
-> before update on produk
-> for each row
-> begin
-> insert into log_harga_produk
-> set kode_produk = old.kode_produk,
-> harga_baru=new.harga,
-> harga_lama=old.harga,
-> waktu_perubahan=now();
-> end$$
Query OK, 0 rows affected (0.048 sec)
```

11. Pada gambar dibawah adalah perintah untuk melakukan insert beberapa data untuk melakukan uji coba dari trigger yang telah dibuat sebelumnya.

```
MariaDB [onlineshopku_e_sitisaadahr]> insert into produk values ('BR001', 'SEMINGGU JAGO CODEIGNITER', '120000');
Query OK, 1 row affected (0.038 sec)

MariaDB [onlineshopku_e_sitisaadahr]> insert into produk values ('BR002', 'SEMINGGU JAGO PHP MYSQL', '80000');
Query OK, 1 row affected (0.022 sec)
```

12. Berikut adalah hasil dari perintah insert diatas :

```
MariaDB [onlineshopku_e_sitisaadahr]> select * from produk;
+-----+-----+-----+
| kode_produk | nama_produk                | harga |
+-----+-----+-----+
| BR001      | SEMINGGU JAGO CODEIGNITER | 120000 |
| BR002      | SEMINGGU JAGO PHP MYSQL   | 80000  |
+-----+-----+-----+
2 rows in set (0.001 sec)
```

13. Kemudian, pada gambar dibawah adalah perintah untuk melakukan update data produk.

```
MariaDB [onlineshopku_e_sitisaadahr]> update produk set harga=9000 where kode_produk='BR001';
Query OK, 1 row affected (0.026 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

14. Pada gambar dibawah, terdapat tampilan dari tabel produk yang sudah di update. Setelah melakukan proses update pada tabel log_harga_produk terdapat informasi kapan perubahan data produk dilakukan.

```

MariaDB [onlineshopku_e_sitisaadahr]> select * from produk;
+-----+-----+-----+
| kode_produk | nama_produk          | harga |
+-----+-----+-----+
| BR001      | SEMINGGU JAGO CODEIGNITER | 9000  |
| BR002      | SEMINGGU JAGO PHP MYSQL  | 80000 |
+-----+-----+-----+
2 rows in set (0.001 sec)

MariaDB [onlineshopku_e_sitisaadahr]> select * from log_harga_produk;
+-----+-----+-----+-----+-----+
| log_id | kode_produk | harga_lama | harga_baru | waktu_perubahan |
+-----+-----+-----+-----+-----+
| 1      | BR001      | 120000    | 9000       | 2024-05-27 09:20:29 |
+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

```

15. Pada gambar dibawah terdapat perintah SHOW TRIGGERS untuk menampilkan list trigger pada database.

```

MariaDB [onlineshopku_e_sitisaadahr]> show triggers;
+-----+-----+-----+-----+-----+-----+-----+
| Trigger          | Event | Table | Statement                               | sql_mode          | Definer          | chara |
+-----+-----+-----+-----+-----+-----+-----+
| before_produk_update | UPDATE | produk | begin
insert into log_harga_produk
set kode_produk = old.kode_produk,
harga_baru=new.harga,
harga_lama=old.harga,
waktu_perubahan=now();
end | BEFORE | 2024-05-27 09:08:35.73 | NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGINE_SUBSTITUTION | root@localhost | utf8mb4 | utf8mb4_general_ci |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.099 sec)

```

16. Terakhir, jika kita ingin menghapus trigger yang sudah kita buat caranya dapat mengetikan perintah seperti pada gambar dibawah.

```

MariaDB [onlineshopku_e_sitisaadahr]> drop trigger before_produk_update;

```