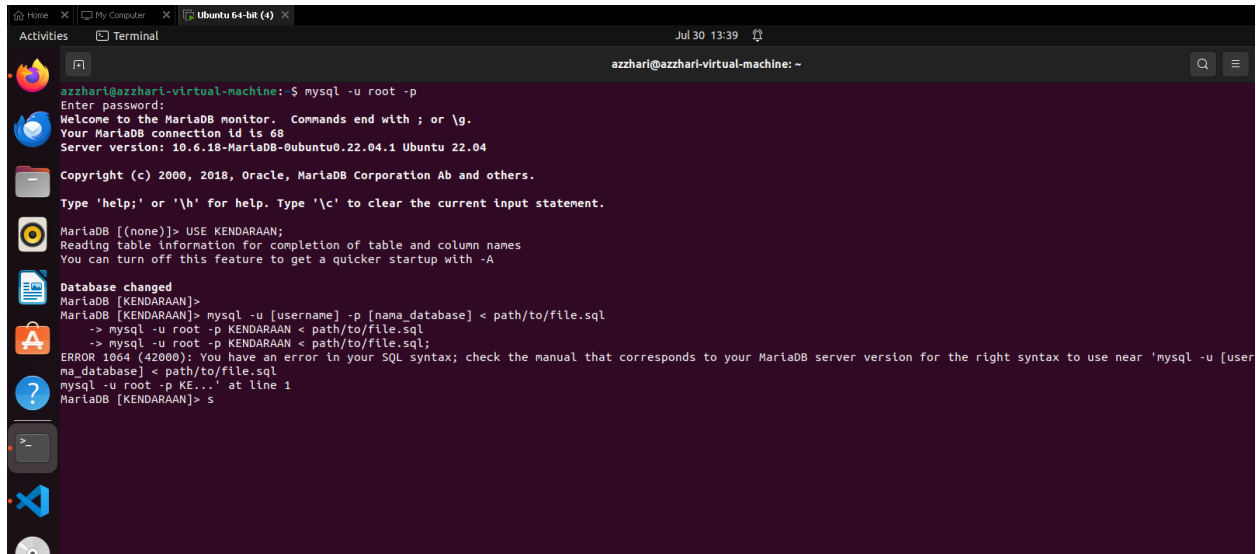


Teknis Database

Pada topik teknis database kali ini, database menggunakan mariadb, source code disimpan kedalam file dengan extension .sql dan untuk soal nomor 1 simpan bersama dengan DOCUMENTATION.md atau DOCUMENTATION.pd



```
azhahri@azhahri-virtual-machine: ~  
$ mysql -u root -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 68  
Server version: 10.6.18-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB [(none)]> USE KENDARAAN;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
Database changed  
MariaDB [KENDARAAN]>  
MariaDB [KENDARAAN]> mysql -u [username] -p [nama_database] < path/to/file.sql  
-> mysql -u root -p KENDARAAN < path/to/file.sql  
-> mysql -u root -p KENDARAAN < path/to/file.sql;  
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'mysql -u [user ma_database] < path/to/file.sql' at line 1  
mysql -u root -p KE...'  
MariaDB [KENDARAAN]> s
```

1. Buatlah langkah-langkah untuk melakukan backup, restore database mariadb pada aplikasi yang anda buat saat ini

JAWABAN:

BACKUP DATABASE

```
mysqldump -u [root] -p kendaraan > kendaraan_backup.sql;
```

RESTORE DATABASE

```
mysql -u [username] -p -e "CREATE DATABASE kendaraan_restore;
```

2. Buatlah tabel menggunakan DDL (Data Definition Language) berdasarkan spesifikasi yang telah dibuat berdasarkan soal nomor 1 topik Teknis Aplikasi

JAWABAN:

CREATE TABLE kendaraan (

```
id INT PRIMARY KEY AUTO_INCREMENT,  
jenis VARCHAR(50) NOT NULL,  
merk VARCHAR(50),  
model VARCHAR(50),  
tahun_produksi INT  
);
```

```
ALTER TABLE kendaraan  
ADD warna VARCHAR(20);
```

```
ALTER TABLE kendaraan  
MODIFY COLUMN tahun_produksi YEAR;
```

```
ALTER TABLE kendaraan  
MODIFY COLUMN tahun_produksi YEAR;
```

```
CREATE TABLE mobil (  
id INT PRIMARY KEY AUTO_INCREMENT,  
kendaraan_id INT,  
jumlah_pintu INT,  
tipe_mesin VARCHAR(50),  
FOREIGN KEY (kendaraan_id) REFERENCES kendaraan(id)  
);
```

```
CREATE TABLE motor (  
id INT PRIMARY KEY AUTO_INCREMENT,  
kendaraan_id INT,  
jenis_motor VARCHAR(50),  
kapasitas_mesin INT,  
FOREIGN KEY (kendaraan_id) REFERENCES kendaraan(id)  
);
```

3. Buatlah store procedure untuk melakukan CRUD pada salah satu table yang telah dibuat berdasarkan soal nomor 2 pada topik ini

JAWABAN:

DELIMITER //

```
CREATE PROCEDURE InsertKendaraan(  

```

```
    IN p_jenis VARCHAR(50),
    IN p_merk VARCHAR(50),
    IN p_model VARCHAR(50),
    IN p_tahun_produksi INT
)
BEGIN
    INSERT INTO kendaraan (jenis, merk, model, tahun_produksi)
    VALUES (p_jenis, p_merk, p_model, p_tahun_produksi);
END //
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE PROCEDURE UpdateKendaraan(
```

```
    IN p_id INT,
    IN p_jenis VARCHAR(50),
    IN p_merk VARCHAR(50),
    IN p_model VARCHAR(50),
    IN p_tahun_produksi INT
)
```

```
BEGIN
```

```
    UPDATE kendaraan
```

```
    SET jenis = p_jenis,
```

```
        merk = p_merk,
```

```
        model = p_model,
```

```
        tahun_produksi = p_tahun_produksi
```

```
WHERE id = p_id;  
END //
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE PROCEDURE DeleteKendaraan(  
    IN p_id INT  
)  
BEGIN  
    DELETE FROM kendaraan  
    WHERE id = p_id;  
END //  
  
DELIMITER ;
```

4. Buatlah trigger untuk mencatat aktivitas ketika insert, update dan delete masing-masing tabel, tampung kedalam sebuah tabel dengan nama histories

JAWABAN:

```
CREATE TABLE histories (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    table_name VARCHAR(50),  
    operation_type VARCHAR(10),  
    operation_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    old_data TEXT,  
    new_data TEXT  
);
```

```
DELIMITER //
```

```
CREATE TRIGGER after_kendaraan_insert
AFTER INSERT ON kendaraan
FOR EACH ROW
BEGIN
    INSERT INTO histories (table_name, operation_type, new_data)
    VALUES ('kendaraan', 'INSERT', CONCAT('{\"id\":', NEW.id, ', \"jenis\":', NEW.jenis, ', \"merk\":',
NEW.merk, ', \"model\":', NEW.model, ', \"tahun_produksi\":', NEW.tahun_produksi, '}'));
END //
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE TRIGGER after_kendaraan_update
AFTER UPDATE ON kendaraan
FOR EACH ROW
BEGIN
    INSERT INTO histories (table_name, operation_type, old_data, new_data)
    VALUES (
        'kendaraan',
        'UPDATE',
        CONCAT('{\"id\":', OLD.id, ', \"jenis\":', OLD.jenis, ', \"merk\":', OLD.merk, ', \"model\":',
OLD.model, ', \"tahun_produksi\":', OLD.tahun_produksi, '}'),
        CONCAT('{\"id\":', NEW.id, ', \"jenis\":', NEW.jenis, ', \"merk\":', NEW.merk, ', \"model\":',
NEW.model, ', \"tahun_produksi\":', NEW.tahun_produksi, '}')
    );
END //
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE TRIGGER after_kendaraan_delete
AFTER DELETE ON kendaraan
FOR EACH ROW
BEGIN
    INSERT INTO histories (table_name, operation_type, old_data)
    VALUES (
        'kendaraan',
        'DELETE',
```

```

        CONCAT({'id':, OLD.id, ', "jenis":', OLD.jenis, ', "merk":', OLD.merk, ', "model":',
OLD.model, ', "tahun_produksi":', OLD.tahun_produksi, '})
    );
END //

```

```

DELIMITER ;

```

5. Buatlah query CTE (Common Table Expressions) untuk menampilkan data

```

WITH KendaraanHistories AS (
    SELECT
        k.id AS kendaraan_id,
        k.jenis,
        k.merk,
        k.model,
        k.tahun_produksi,
        h.operation_type,
        h.operation_time
    FROM kendaraan k
    LEFT JOIN histories h
    ON k.id = JSON_EXTRACT(h.old_data, '$.id') -- Menyesuaikan untuk format JSON
    WHERE h.table_name = 'kendaraan'
    ORDER BY h.operation_time DESC
)
SELECT * FROM KendaraanHistories;

```