

## TEKNIS DATABASE

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

### Jawaban:

#### - Backup

Langkah-langkah untuk melakukan backup database mariadb salah satunya adalah menggunakan **mysqldump**. Caranya adalah dengan memasukkan command di bawah ini pada command line ubuntu

```
mysqldump -u [username] -p [database_name] > [backup_file].sql
```

username merupakan nama pengguna dari mariadb, dalam project yang dibuat, digunakan username **root** dan nama databasenya adalah **airplane**, sehingga untuk menbackup mariadb pada aplikasi yang dibuat, dapat menggunakan command berikut

```
mysqldump -u root -p airplane > airplane.sql
```

File sql akan tersimpan di dalam direktori laravel yang kita pilih.

#### - Restore

Untuk melakukan restore database mariadb, dapat menggunakan command di bawah ini

```
mysql -u root -p airplane < airplane.sql
```

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

**Jawaban:**

- DDL untuk menabahkan table parent 'pesawat'  

```
CREATE TABLE pesawat (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    code VARCHAR(50) NOT NULL,  
    produsen VARCHAR(50) NOT NULL,  
    tahun YEAR NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP  
);
```
- DDL untuk menabahkan table child 1 'pesawat\_tempur'  

```
CREATE TABLE 'pesawat_tempur' (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    pesawat_id CHAR(225) NOT NULL,  
    senjata VARCHAR(100) NOT NULL,  
    kecepatan_maks INT (11) NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP  
);
```
- DDL untuk menabahkan table child 2 'pesawat\_penumpang'  

```
CREATE TABLE 'pesawat_penumpang' (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    pesawat_id CHAR(225) NOT NULL,  
    no_pesawat VARCHAR(100) NOT NULL,  
    penumpang_maks INT (11) NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP  
);
```

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

**Jawaban:**

- Store Procedure untuk melakukan **Create Data**

```
CREATE PROCEDURE InsertPesawat(  
    IN p_code VARCHAR(50),  
    IN p_produken VARCHAR(50),  
    IN p_tahun YEAR  
)  
BEGIN  
    INSERT INTO pesawat (code, produken, tahun)  
    VALUES (p_code, p_produken, p_tahun);  
END
```

**Cara execute:**

```
CALL InsertPesawat('Cessna 172', 'Cessna', 1956);
```

- Store Procedure untuk melakukan **Read Data**

```
CREATE PROCEDURE GetAllPesawat()  
BEGIN  
    SELECT * FROM pesawat;  
END
```

**Cara execute:**

```
CALL GetAllPesawat();
```

- Store Procedure untuk melakukan **Update Data**

```
CREATE PROCEDURE UpdatePesawat(  
    IN p_id INT,  
    IN p_code VARCHAR(50),  
    IN p_produken VARCHAR(50),  
    IN p_tahun YEAR  
)  
BEGIN  
    UPDATE pesawat  
    SET code = p_code,  
        produken = p_produken,  
        tahun = p_tahun  
    WHERE id = p_id;  
END
```

**Cara execute:**

```
CALL UpdatePesawat(1, 'Cessna 172 Updated', 'Cessna', 1957);
```

- Store Procedure untuk melakukan **Delete Data**

```
CREATE PROCEDURE DeletePesawat(  
    IN p_id INT  
)  
BEGIN  
    DELETE FROM pesawat WHERE id = p_id;
```

END

**Cara execute:**

**CALL** DeletePesawat (1) ;

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

**Jawaban:**

Langkah yang harus dilakukan dalam mencatat aktivitas ketika terjadi perubahan pada data tabel atau biasa disebut data logs adalah sebagai berikut:

- Buat table dengan nama **histories**  

```
CREATE TABLE histories (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    table_name VARCHAR(50) NOT NULL,  
    action_type ENUM('INSERT', 'UPDATE', 'DELETE') NOT NULL,  
    record_id INT NOT NULL,  
    old_data TEXT,  
    new_data TEXT,  
    action_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```
- Buat trigger yang akan berjalan ketika insert data pesawat dilakukan  

```
CREATE TRIGGER insert_logs  
AFTER INSERT ON pesawat  
FOR EACH ROW  
BEGIN  
    INSERT INTO histories (table_name, action_type, record_id,  
new_data)  
VALUES ('pesawat', 'INSERT', NEW.id, CONCAT('code:',  
NEW.code, ', produsen:', NEW.produksen, ', tahun:',  
NEW.tahun));  
END
```
- Buat trigger yang akan berjalan ketika update data pesawat dilakukan  

```
CREATE TRIGGER after_pesawat_update  
AFTER UPDATE ON pesawat  
FOR EACH ROW  
BEGIN  
    INSERT INTO histories (table_name, action_type, record_id,  
old_data, new_data)  
VALUES ('pesawat', 'UPDATE', OLD.id,  
CONCAT('code:', OLD.code, ', produsen:', OLD.produksen,  
, tahun:', OLD.tahun),  
CONCAT('code:', NEW.code, ', produsen:', NEW.produksen,  
, tahun:', NEW.tahun));  
END
```
- Buat trigger yang akan berjalan ketika delete data pesawat dilakukan  

```
CREATE TRIGGER delete_logs  
AFTER DELETE ON pesawat  
FOR EACH ROW  
BEGIN  
    INSERT INTO histories (table_name, action_type, record_id,  
old_data)  
VALUES ('pesawat', 'DELETE', OLD.id,  
CONCAT('code:', OLD.code, ', produsen:', OLD.produksen,  
, tahun:', OLD.tahun));  
END
```

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

**Jawaban:**

```
WITH cte_pesawat_penumpang AS (  
    SELECT * FROM pesawat_penumpang pp  
)  
  
SELECT  
    pesawat.*,  
    pp.penumpang_maks,  
    pp.no_pesawat  
FROM pesawat  
JOIN cte_pesawat_penumpang pp  
ON pesawat.id = pp.pesawat_id
```

---

```
WITH cte_pesawat_tempur AS (  
    SELECT * FROM pesawat_tempur pt  
)  
  
SELECT  
    pesawat.*,  
    pt.kecepatan_maks,  
    pt.senjata  
FROM pesawat  
JOIN cte_pesawat_tempur pt  
ON pesawat.id = pt.pesawat_id
```