1. **BACKUP & RESTORE DATABASE**
    - Backup
      step :
        - login ke mysql menggunakan perintah sudo mysql -u root -p
        - gunakan perintah mysqldump -u root -p nama_database > nama_database_backup.sql
    - Restore :
      step :
        - login ke mysql menggunakan perintah sudo mysql -u root -p
        - create database dengan perinnah create database nama_database_baru
        - gunakan perintah mysql -u root -p nama_database_baru < nama_database_backup.sql

2. **DDL**

```
CREATE TABLE universitas (
        id INT AUTO_INCREMENT PRIMARY KEY,
        name VARCHAR(255) NOT NULL,
        alamat VARCHAR(255) NOT NULL,
        created_at DATETIME,
        updated_at DATETIME
);
```

3. **Stores Procedure**
    - Create

```
DELIMITER //
CREATE PROCEDURE CreateUniv(
        IN name VARCHAR(255),
        IN address VARCHAR(255)
)
BEGIN
        INSERT INTO universitas (
                nama, alamat, created_at, updated_at
        ) VALUES (
                name, address, NOW(), NOW()
        );
END //
DELIMITER ;
CALL CreateUniv('OPQ', 'Jatiasih');
```

    - Read

```
DELIMITER //
CREATE PROCEDURE ReadUniv()
BEGIN
        SELECT * FROM universitas;
END //
DELIMITER ;
CALL ReadUniv();
```

- Update

```
DELIMITER //
CREATE PROCEDURE UpdateUniv(
        IN univ_id INT,
        IN name VARCHAR(255),
        IN address VARCHAR(255)
)
BEGIN
        UPDATE universitas SET
                nama = name,
                alamat = address,
                updated_at = NOW()
        WHERE id = univ_id;
END //
DELIMITER ;
CALL UpdateUniv(1, 'XYZ', 'Bekasi'');
```

- Delete

```
DELIMITER //
CREATE PROCEDURE DeleteUniv(IN univ_id INT)
BEGIN
        DELETE FROM universitas WHERE id = univ_id;
END //
DELIMITER ;
CALL DeleteUniv(2);
```

4. Trigger

- Buat Tabel histories

```
CREATE TABLE histories (
        id INT AUTO_INCREMENT PRIMARY KEY,
        univ_id INT,
        action VARCHAR(50),
        old_nama VARCHAR(255),
        new_nama VARCHAR(255),
        old_alamat VARCHAR(255),
        new_alamat VARCHAR(255),
        changed_at DATETIME
 );
```

- Trigger Insert

```
DELIMITER //
CREATE TRIGGER after_univ_insert
AFTER INSERT ON universitas FOR EACH ROW
BEGIN
        INSERT INTO histories (
                id, action, new_nama, new_alamat, changed_at
        ) VALUES (
```

```
            NEW.id, 'INSERT', NEW.nama, NEW.alamat, NOW()
        );
    END //
    DELIMITER ;
```

- Trigger UPDATE

```
    DELIMITER //
    CREATE TRIGGER after_univ_update
    AFTER UPDATE ON universitas FOR EACH ROW
    BEGIN
        INSERT INTO histories (
            id, action, old_nama, new_nama, old_alamat, new_alamat,
        changed_at
        ) VALUES (
            OLD.id, 'UPDATE', OLD.nama, NEW.nama, OLD.alamat,
        NEW.alamat, NOW()
        );
    END //
    DELIMITER ;
```

- Buat Trigger untuk DELETE

```
    DELIMITER //
    CREATE TRIGGER after_univ_delete
    AFTER DELETE ON universitas FOR EACH ROW
    BEGIN
        INSERT INTO histories (
            id, action, old_nama, old_alamat, changed_at
        ) VALUES (
            OLD.id, 'DELETE', OLD.nama, OLD.alamat, NOW()
        );
    END //
    DELIMITER;
```

5. CTE

```
WITH CTE AS (
    SELECT h.univ_id, h.action, h.old_nama, h.new_nama, h.old_alamat, h.new_alamat,
    h.changed_at,
    ROW_NUMBER() OVER (
        PARTITION BY h.univ_id ORDER BY h.changed_at DESC
    ) AS rn
    FROM histories h
)
SELECT s.id, s.nama, s.alamat, s.created_at, s.updated_at,
lh.action, lh.old_nama, lh.new_nama, lh.old_alamat, lh.new_alamat, lh.changed_at
FROM universitas s
LEFT JOIN CTE lh ON s.id = lh.univ_id AND lh.rn = 1
ORDER BY s.id;
```