

TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK
MODUL X
DATA STORAGE BAGIAN 1



Disusun Oleh :
Lintang Suminar Tyas Wening / 2211104009
SE-06-01

Asisten Praktikum :
Muhammad Faza Zulian Gesit Al Barru
Aisyah Hasna Aulia

Dosen Pengampu :
Yudha Islami Sulistya, S.Kom., M.Cs

PROGRAM STUDI S1 Software Engineering
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

TUGAS PENDAHULUAN

SOAL

1. Jelaskan secara singkat fungsi SQLite dalam pengembangan aplikasi mobile!
2. Apa saja yang dimaksud dengan operasi CRUD? Berikan penjelasan singkat untuk masing-masing operasi!
3. Tuliskan kode SQL untuk membuat tabel bernama **users** dengan kolom berikut :
 - id (integer, primary key, auto increment)
 - name (text)
 - email (text)
 - createdAt (timestamp, default value adalah waktu sekarang)
4. Sebutkan langkah-langkah utama untuk menggunakan plugin sqflite di dalam Flutter!
5. Lengkapi kode berikut untuk membaca semua data dari tabel **users** menggunakan sqflite.

```
static Future<List<Map<String, dynamic>>> getUsers() async {  
    final db = await SQLHelper.db();  
    return db.query(______);  
}
```

JAWAB:

1. SQLite adalah sistem manajemen basis data (database) relasional yang ringan, cepat, dan mudah digunakan, sering dimanfaatkan sebagai penyimpanan data lokal. SQLite memungkinkan pengelolaan data seperti pengguna, preferensi, atau cache secara efisien tanpa memerlukan server, sehingga aplikasi tetap berfungsi meskipun offline.
2. Operasi CRUD adalah singkatan dari Create, Read, Update, dan Delete, yang merupakan empat operasi dasar dalam pengelolaan data.
 - Create : Menambahkan data baru ke dalam database
 - Read : Mengambil atau membaca data yang sudah ada dari database
 - Update : Mengubah data yang sudah ada dalam database
 - Delete : Menghapus data dari database
3. CREATE TABLE users (
 id INTEGER PRIMARY KEY AUTOINCREMENT,
 name TEXT,
 email TEXT,
 createdAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
4. Langkah-langkah utama untuk menggunakan plugin sqflite di dalam Flutter sebagai berikut :

- a.) Tambahkan plugin sqflite dan path untuk mengelola lokasi file basis data di file pubspec.yaml

```
dependencies:  
  flutter:  
  
  sqflite: ^2.2.0  
  path: ^1.8.3
```

- b.) Lalu jalankan perintah “flutter pub get”

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER  
PS D:\SEMESTER 5\flutter_application_1_basdat> flutter pub get  
Resolving dependencies...  
Downloading packages...  
  async 2.11.0 (2.12.0 available)  
  boolean_selector 2.1.1 (2.1.2 available)  
  characters 1.3.0 (1.3.1 available)  
  clock 1.1.1 (1.1.2 available)
```

- c.) Import Plugin di File Dart yang diperlukan

```
1 import 'package:flutter/material.dart';  
2 import 'package:sqflite/sqflite.dart';  
3 import 'package:path/path.dart';
```

- d.) Buat fungsi untuk menginisialisasi basis data SQLite

```
Future<Database> initializeDB() async {  
  final dbPath = await getDatabasesPath();  
  return openDatabase(  
    join(dbPath, 'my_database.db'),  
    onCreate: (db, version) async {  
      await db.execute('''  
        CREATE TABLE users(  
          id INTEGER PRIMARY KEY AUTOINCREMENT,  
          name TEXT NOT NULL,  
          email TEXT NOT NULL  
        )  
      ''');  
    },  
    version: 1,  
  );  
}
```

- e.) Selanjutnya membuat operasi CRUD

- Create : Untuk menambahkan data ke tabel

```
Future<void> insertUser(Database db, String name, String email) async {  
  await db.insert(  
    'users',  
    {'name': name, 'email': email},  
    conflictAlgorithm: ConflictAlgorithm.replace,  
  );  
}
```

- Read : Untuk membaca dan mengambil semua data dari tabel

```
Future<List<Map<String, dynamic>>> getUsers(Database db) async {  
  return await db.query('users');  
}
```

- Update : Untuk memperbarui data tertentu

```
Future<void> updateUser(Database db, int id, String name) async {
  await db.update(
    'users',
    {'name': name},
    where: 'id = ?',
    whereArgs: [id],
  );
}
```

- Delete : Untuk menghapus data

```
Future<void> deleteUser(Database db, int id) async {
  await db.delete(
    'users',
    where: 'id = ?',
    whereArgs: [id],
  );
}
```

- Integrasikan fungsi di atas ke dalam logika aplikasi, misalnya dengan memanggil initializeDB() saat aplikasi dimulai:

```
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  final db = await initializeDB();

  // Contoh: Menambahkan data
  await insertUser(db, "John Doe", "johndoe@example.com");

  // Contoh: Membaca data
  final users = await getUsers(db);
  print(users);
}
```

5. Berikut kode lengkap untuk membaca semua data dari tabel *users*

```
static Future<List<Map<String, dynamic>>> getUsers() async {
  final db = await SQLHelper.db();
  return db.query('users');
}
```

Source Code:

```
static Future<List<Map<String, dynamic>>> getUsers() async {
  final db = await SQLHelper.db();

  // Membaca semua data dari tabel users
  return db.query('users');
}
```

Jadi kode yang dilengkapi adalah 'users'