

**TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL X
DATA STORAGE (BAGIAN I)**



Disusun Oleh :

M. Edgar Nadhif

2211104019

SE0601

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! SQLite digunakan untuk menyimpan data secara lokal di perangkat mobile, memungkinkan aplikasi mengelola data tanpa koneksi internet.
2. Apa saja yang dimaksud dengan operasi CRUD? Berikan penjelasan singkat untuk masing-masing operasi!

CRUD adalah singkatan dari Create, Read, Update, dan Delete, yang merupakan operasi dasar dalam manajemen data:

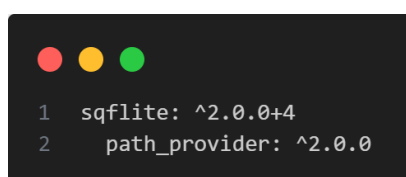
1. Create: Menambahkan data baru ke dalam basis data.
 2. Read: Membaca atau mengambil data yang sudah ada.
 3. Update: Memperbarui atau mengubah data yang sudah ada.
 4. Delete: Menghapus data dari basis data.
3. Tuliskan kode SQL untuk membuat tabel bernama **users** dengan kolom berikut :
 - id (integer, primary key, auto increment)
 - name (text)
 - email (text)
 - createdAt (timestamp, default value adalah waktu sekarang)

Jawab :

```
CREATE TABLE users (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    name TEXT,  
    email TEXT,  
    createdAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

4. Sebutkan langkah-langkah utama untuk menggunakan plugin sqflite di dalam Flutter!

1. Tambahkan dependensi: Masukkan sqflite ke dalam file pubspec.yaml:



```
1  sqflite: ^2.0.0+4  
2  path_provider: ^2.0.0
```

2. Import sqflite dan path_provider di file Dart tempat kamu akan menggunakan database:

```
1 import 'package:sqflite/sqflite.dart';
2 import 'package:path_provider/path_provider.dart';
```

3. Buat fungsi untuk membuka database :

```
1 Future<Database> openDatabase() async {
2   final directory = await getApplicationDocumentsDirectory();
3   final path = '${directory.path}/my_db.db';
4   return openDatabase(path, version: 1, onCreate: (db, version) async {
5     await db.execute('CREATE TABLE items(id INTEGER PRIMARY KEY, name TEXT)');
6   });
7 }
```

4. Menambah data ke database

Fungsi untuk memasukkan data :

```
1 Future<void> insertItem(Database db, Map<String, dynamic> item) async {
2   await db.insert('items', item, conflictAlgorithm: ConflictAlgorithm.replace);
3 }
```

5. Fungsi untuk membaca data:

```
1 Future<List<Map<String, dynamic>>> getItems(Database db) async {
2   return await db.query('items');
3 }
4
```

6. Menutup database

Setelah selesai, pastikan untuk menutup database:

```
1 await db.close();
```

5. Lengkapi kode berikut untuk membaca semua data dari tabel *users* menggunakan

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



```
1 static Future<List<Map<String, dynamic>>> getUsers() async {  
2     final db = await SQLHelper.db();  
3     return db.query('users'); // Mengambil semua data dari tabel 'users'  
4 }
```