

**TUGAS PENDAHULUAN**  
**PEMROGRAMAN PERANGKAT BERGERAK**  
**MODUL X**  
**DATA STORAGE (BAGIAN I)**



**Disusun Oleh :**

**Mohammad Dhimas Afrizal / 2211104023**

**SE06-01**

**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

### JAWABAN

1. SQLite adalah **sistem manajemen basis data relasional (RDBMS)** yang ringan dan sering digunakan dalam pengembangan aplikasi mobile. Fungsi utama SQLite adalah:

- **Menyimpan Data secara Lokal:** SQLite digunakan untuk menyimpan data aplikasi secara lokal di perangkat pengguna tanpa memerlukan server eksternal. Cocok untuk aplikasi offline.
- **Pengelolaan Basis Data yang Mudah:** SQLite memungkinkan pengelolaan data menggunakan perintah SQL standar (seperti SELECT, INSERT, UPDATE, DELETE).
- **Ringan dan Cepat:** SQLite memiliki ukuran yang kecil dan efisien, sehingga tidak membebani performa aplikasi mobile.
- **Tidak Memerlukan Konfigurasi Server:** Berbeda dengan RDBMS lain, SQLite bersifat self-contained dan tidak memerlukan proses server untuk menjalankan operasinya.
- **Portabilitas Data:** File database SQLite dapat dengan mudah dipindahkan ke perangkat lain karena hanya berupa file tunggal.

Contoh penggunaan: aplikasi catatan, penyimpanan preferensi pengguna, atau caching data API.

2. Operasi CRUD adalah empat fungsi utama yang digunakan dalam pengelolaan data pada basis data, yaitu:

- **Create (Membuat):**  
Operasi ini digunakan untuk menambahkan data baru ke dalam basis data.

Contoh: Menambahkan pengguna baru ke tabel pengguna dengan perintah SQL INSERT INTO.

- **Read (Membaca):**  
Operasi ini digunakan untuk mengambil atau membaca data yang ada di basis data tanpa mengubahnya.

Contoh: Mengambil informasi semua pengguna dari tabel dengan perintah SQL SELECT.

- **Update (Memperbarui):**  
Operasi ini digunakan untuk mengubah data yang sudah ada di basis data.

Contoh: Memperbarui alamat email pengguna dengan perintah SQL UPDATE.

- **Delete (Menghapus):**

Operasi ini digunakan untuk menghapus data dari basis data.

Contoh: Menghapus pengguna tertentu dari tabel dengan perintah SQL DELETE.

CRUD adalah dasar dari pengelolaan data di aplikasi modern, baik untuk aplikasi web maupun mobile.

3. Tuliskan kode SQL untuk membuat tabel bernama *users* dengan kolom berikut :

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

Penjelasan:

- id: Kolom dengan tipe data INTEGER, menjadi primary key, dan auto increment untuk menambah nilai secara otomatis.
- name dan email: Kolom dengan tipe data TEXT untuk menyimpan nama dan email.
- createdAt: Kolom dengan tipe data TIMESTAMP, dan menggunakan DEFAULT CURRENT\_TIMESTAMP untuk menyimpan waktu saat data dimasukkan.

4. **Tambahkan dependensi sqflite di pubspec.yaml:**

Buka file pubspec.yaml dan tambahkan sqflite dan path (untuk mengelola lokasi file database) di bagian dependencies.

```
dependencies:  
  sqflite: ^2.0.0+3  
  path: ^1.8.0
```

**Import package:**

Di file Dart yang akan menggunakan database, import sqflite dan path.

```
import 'package:sqflite/sqflite.dart';  
import 'package:path/path.dart';
```

**Inisialisasi dan buka database:**

Buat fungsi untuk membuka atau membuat database jika belum ada.

```
Future<Database> openDatabase() async {  
    var databasesPath = await getDatabasesPath();  
    String path = join(databasesPath, 'my_database.db');  
  
    return await openDatabase(path, version: 1, onCreate:  
    (Database db, int version) async {  
        await db.execute('''CREATE TABLE users(  
            id INTEGER PRIMARY KEY AUTOINCREMENT,  
            name TEXT,  
            email TEXT,  
            createdAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
        )''');  
    });  
}
```

**Operasikan data (CRUD):**

Gunakan operasi CRUD seperti insert(), query(), update(), dan delete() untuk berinteraksi dengan database.

```
Future<int> insertUser(Database db, Map<String, dynamic>  
user) async {  
    return await db.insert('users', user);  
}
```

**Tutup database:**

Jangan lupa untuk menutup database setelah selesai digunakan.

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

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

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

Penjelasan:

- `db.query('users')` akan mengambil semua data dari tabel `users`. Fungsi `query` ini mengembalikan hasil dalam bentuk `List<Map<String, dynamic>>`, di mana setiap `Map` berisi pasangan `key-value` untuk setiap kolom dalam tabel `users`.