

**TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL VII
DATA STORAGE**



Disusun Oleh :

Nadia Putri Rahmaniar / 2211104012

S1SE0601

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

1. Jelaskan secara singkat fungsi SQLite dalam pengembangan aplikasi mobile!

SQLite adalah sistem manajemen basis data (DBMS) yang ringan yang tertanam langsung dalam aplikasi. Tidak memerlukan server terpisah, sehingga cocok untuk aplikasi mobile yang membutuhkan banyak sumber daya. Dalam proses pembuatan aplikasi untuk perangkat seluler, SQLite juga sebagai penyimpanan data lokal yang dapat digunakan untuk menyimpan dan mengelola data pengguna, seperti data login, preferensi, dan konten aplikasi yang dapat diakses secara offline. SQLite juga mendukung operasi CRUD, yaitu membuat, membaca, mengubah, dan menghapus. Selain itu, API-nya yang sederhana dan mudah digunakan membuat SQLite mudah digunakan untuk aplikasi mobile.


2. Apa saja yang dimaksud dengan operasi CRUD? Berikan penjelasan singkat untuk masing-masing operasi!

Fungsi utama yang digunakan dalam pengelolaan database adalah CRUD, yang merupakan singkatan dari Create, Read, Update, dan Delete. Fungsi ini biasanya digunakan dalam aplikasi pengelolaan database seperti MySQL, Oracle, SQL Server, dan MongoDB, dll. Berikut adalah penjelasan tentang fungsi CRUD:

- Create : Untuk menambahkan data baru ke dalam database, misalnya untuk menambahkan data pengguna baru untuk aplikasi.
- Read : mengambil dan menampilkan data yang tersimpan dalam database tanpa mengubahnya, seperti menampilkan verifikasi dari data yang dimasukkan pengguna.
- Update adalah operasi yang digunakan untuk mengubah atau memperbarui data yang ada dalam database. Contohnya, memperbarui data alamat user.
- Delete adalah operasi yang digunakan untuk menghapus data dari dalam database. Contohnya, admin menghapus data user.

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)



```


1 CREATE TABLE users (
2   id INTEGER PRIMARY KEY AUTOINCREMENT,
3   name TEXT,
4   email TEXT,
5   createdAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP
6 );

```

- a. id (integer, primary key, auto increment) : menggunakan tipe data integer yang memiliki auto increment dan berfungsi sebagai kunci utama, sehingga nilainya otomatis bertambah setiap kali data baru ditambahkan.
- b. name (text) : menggunakan tipe data teks untuk menyimpan nama pengguna.
- c. email (text) : menggunakan tipe data text yang digunakan untuk menyimpan alamat email pengguna.
- d. createdAt : agar otomatis menyimpan waktu saat data ditambahkan, gunakan tipe data timestamp dan nilai defaultnya adalah CURRENT_TIMESTAMP.

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

- a. Menambahkan dependensi sqflite dan path_provider di pubspec.yaml




```

dependencies:
sqflite: ^2.2.5
path_provider: ^2.0.15

```

- b. Import package sqflite dan path di dalam file yang akan digunakan untuk mengatur database.



```

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

```

- c. Buat fungsi untuk membuka dan membuat database

```
1 Future<Database> initDatabase() async {
2   final dbPath = await getDatabasesPath();
3   final path = join(dbPath, 'my_database.db');
4
5   return openDatabase(
6     path,
7     onCreate: (db, version) {
8       return db.execute(
9         'CREATE TABLE users(id INTEGER PRIMARY KEY, name TEXT, email
10        TEXT)',
11       );
12     },
13     version: 1,
14   );
15 }
16
```

- d. Buat fungsi untuk CRUD

- Create

```
1 Future<void> insertUser(Database db, Map<String, dynamic> user) async
2 {
3   await db.insert(
4     'users',
5     user,
6     conflictAlgorithm: ConflictAlgorithm.replace,
7   );
8 }
9
```

- Read

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

- Update

```
1 Future<void> updateUser(Database db, Map<String, dynamic> user)
2 async {
3   await db.update(
4     'users',
5     user,
6     where: 'id = ?',
7     whereArgs: [user['id']],
8   );
9 }
```

- Delete

```
1 Future<void> deleteUser(Database db, int id) async {
2     await db.delete(
3         'users',
4         where: 'id = ?',
5         whereArgs: [id],
6     );
7 }
```

- e. Panggil fungsi CRUD di dalam widget yang diperlukan
- f. Untuk mencegah kebocoran data, gunakan perintah berikut untuk menutup koneksi database saat tidak digunakan:

```
1 Future<void> closeDatabase(Database db) async {
2     await db.close();
3 }
```

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

```
static Future<>> 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 await db.query('users');
4 }
```