

**TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK**

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



Disusun Oleh :

Rizky Hanifa Afania / 2211104017

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

SOAL

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

Jawab:

SQLite adalah sistem manajemen basis data (DBMS) yang ringan dan tertanam langsung dalam aplikasi. Database ini tidak memerlukan server terpisah sehingga cocok untuk aplikasi mobile dengan sumber daya terbatas. Dalam pengembangan aplikasi mobile, SQLite berfungsi sebagai penyimpanan data lokal yang dapat digunakan untuk menyimpan dan mengelola data pengguna secara efisien, seperti data login, preferensi, dan konten aplikasi yang dapat diakses secara offline. SQLite mendukung operasi CRUD yaitu *create*, *read*, *update*, dan *delete*. Selain itu, SQLite mempunyai API yang sederhana dan mudah digunakan, sehingga mudah diimplementasikan dalam aplikasi mobile.

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

Jawab:

CRUD adalah singkatan dari *Create*, *Read*, *Update*, dan *Delete* yaitu fungsi dasar yang digunakan dalam pengelolaan database. Biasanya fungsi ini digunakan dalam aplikasi pengelolaan database seperti MySQL, Oracle, SQL Server, MongoDB, dll. Berikut ini adalah penjelasan dari setiap fungsi CRUD:

- **Create**

Create adalah operasi yang digunakan untuk menambahkan data baru ke dalam database. Contohnya saat memasukkan data registrasi pengguna baru suatu aplikasi

- **Read**

Read adalah operasi yang digunakan untuk mengambil dan menampilkan data yang tersimpan dalam database tanpa mengubahnya. Contohnya ketika menampilkan verifikasi dari data yang telah diinput pengguna.

- **Update**

Update adalah operasi yang digunakan untuk mengubah atau memperbarui data yang ada dalam database. Contohnya, memperbarui kata sandi pengguna.

- **Delete**

Delete adalah operasi yang digunakan untuk menghapus data dari dalam database. Contohnya, menghapus produk dari keranjang belanja pengguna pada aplikasi e-commerce.

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  
);
```

- **id**

Menggunakan tipe data integer yang berfungsi sebagai primary key dan dengan auto increment, sehingga nilainya otomatis bertambah setiap kali data baru dimasukkan.

- **name**

Menggunakan tipe data text yang digunakan untuk menyimpan nama pengguna.

- **email**

Menggunakan tipe data text yang digunakan untuk menyimpan alamat email pengguna.

- **createdAt**

Menggunakan tipe datatimestamp dan menggunakan CURRENT_TIMESTAMP sebagai nilai default agar otomatis menyimpan waktu saat data ditambahkan.

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

Jawab:

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

```
dependencies:  
  sqflite: ^2.2.5  
  path_provider: ^2.0.15
```

Setelah itu jalankan perintah “flutter pub get” di terminal untuk mengunduh dependensi

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

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

- c. Buat fungsi untuk membuka dan membuat database

```
Future<Database> initDatabase() async {
  final dbPath = await getDatabasesPath();
  final path = join(dbPath, 'my_database.db');

  return openDatabase(
    path,
    onCreate: (db, version) {
      return db.execute(
        'CREATE TABLE users(id INTEGER PRIMARY KEY, name TEXT, email TEXT)',
      );
    },
    version: 1,
  );
}
```

- d. Buat fungsi untuk CRUD

- Create

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

- Read

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

- Update

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

- Delete

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

```
);  
}
```

- e. Panggil fungsi CRUD di dalam widget yang diperlukan
- f. Tutup koneksi database saat tidak digunakan untuk menghindari kebocoran data, dengan perintah berikut:

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

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(_____); }
```

Jawab:

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