TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK

MODUL X
DATA STORAGE (BAGIAN I)

Disusun Oleh :

Aorinka Anendya Chazanah / 2211104013

S1 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

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('users'); /
}
```

JAWABAN

1. **Fungsi SQLite dalam Pengembangan Aplikasi Mobile**

   SQLite adalah *Database Management System* yang ringan dan terintegrasi dalam banyak platform, termasuk Android dan iOS. Fungsinya dalam pengembangan aplikasi mobile adalah menyimpan dan mengelola data secara lokal di perangkat tanpa memerlukan server eksternal. SQLite cocok untuk aplikasi yang membutuhkan penyimpanan data offline, seperti daftar tugas, catatan, atau cache.
2. CRUD adalah singkatan dari Create, Read, Update, Delete, yang merupakan empat operasi utama untuk mengelola data dalam database.
    - Create: Menambahkan data baru ke dalam tabel.
    - Read: Membaca atau mengambil data yang tersimpan.
    - Update: Memperbarui data yang sudah ada.
    - Delete: Menghapus data dari tabel.
3. Membuat tabel “users”

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

4. Langkah-langkah menggunakan plugin sqflite
    1) Tambahkan plugin sqflite di pubspec.yaml.

```
dependencies:
  flutter:
    sdk: flutter
  sqflite: ^2.0.0
  path: ^1.8.0
```

    2) Ketikkan “flutter pub get” pada terminal.
    3) Buat file helper untuk mengelola database, termasuk fungsi untuk membuat tabel, di sini bernama “database_helper.dart”
    4) Import package

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

    5) Implementasikan fungsi untuk menambahkan, membaca, memperbarui, dan menghapus data.
    6) Integrasikan fungsi CRUD ke dalam logika aplikasi untuk digunakan di UI.
5. Melengkapi kode untuk membaca semua data pada tabel “users”

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

class SQLHelper {
  static Future<Database> db() async {
    final path = await getDatabasesPath();
    return openDatabase(
      join(path, 'app_database.db'),
      onCreate: (db, version) {
        return db.execute(
          'CREATE TABLE users (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT, email TEXT,
createdAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP)',
        );
      },
      version: 1,
    );
  }

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

// Create
Future<int> addUser(String name, String email) async {
  final db = await SQLHelper.db();
  return await db.insert('users', {'name': name, 'email': email});
}

// Update
Future<int> updateUser(int id, String name, String email) async {
  final db = await SQLHelper.db();
  return await db.update(
    'users',
    {'name': name, 'email': email},
    where: 'id = ?',
    whereArgs: [id],
  );
}

// Delete
Future<int> deleteUser(int id) async {
```

```
  final db = await SQLHelper.db();
  return await db.delete('users', where: 'id = ?', whereArgs: [id]);
}

class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  List<Map<String, dynamic>> _users = [];

  @override
  void initState() {
    super.initState();
    fetchUsers();
  }

  void fetchUsers() async {
    final users = await SQLHelper.getUsers();
    setState(() {
      _users = users;
    });
    print(_users); // Debugging: Print data ke console
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('SQLite Example'),
          centerTitle: true,
        ),
        body: _users.isEmpty
            ? Center(child: Text('No users found'))
            : ListView.builder(
                itemCount: _users.length,
                itemBuilder: (context, index) {
                  final user = _users[index];
                  return ListTile(
                    title: Text(user['name']),
                    subtitle: Text(user['email']),
                  );
                },
              ),
      ),
    );
  }
}

void main() {
  runApp(MyApp());
}
```

Deskripsi Program

Program ini merupakan contoh aplikasi Flutter yang menggunakan SQLite untuk menyimpan dan mengelola data pengguna dalam bentuk tabel. Aplikasi ini menggunakan package sqflite untuk berinteraksi dengan basis data SQLite dan paket path untuk menangani jalur file basis data. Class SQLHelper bertanggung jawab untuk mengatur koneksi ke basis data dan menyediakan fungsi untuk membaca data pengguna menggunakan metode getUsers(), yang mengembalikan semua data pengguna dari tabel users. Fungsi lain yang disediakan adalah untuk menambahkan, memperbarui, dan menghapus data pengguna dari basis data menggunakan metode addUser(), updateUser(), dan deleteUser().

Kode tersebut dapat menampilkan daftar pengguna dalam widget ListView.builder, yang mengambil data dari basis data dan menampilkannya dalam bentuk ListTile yang mencakup nama dan email pengguna. Fungsi fetchUsers() dipanggil di dalam metode initState() untuk mengambil dan menampilkan data pengguna saat aplikasi pertama kali dimulai. Jika tidak ada data pengguna, aplikasi akan menampilkan pesan "No users found". Aplikasi ini memberikan contoh bagaimana cara mengelola data secara lokal menggunakan SQLite dan menampilkan hasilnya dalam antarmuka pengguna Flutter.