

LAPORAN PRAKTIKUM
Modul 10
“Data Storgae (Bagian 1)”



Disusun Oleh:
Muhammad Ralfi - 2211104054
Kelas SE-06-2

Dosen :
Yudha Islami Sulistya, S.Kom., M.Cs.

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

1. Tujuan

- Mahasiswa mampu memahami konsep layout pada Flutter
- Mahasiswa dapat mengimplementasikan desain user interface pada Flutter

2. Landasan Teori

a. Sqlite

SQLite adalah mesin basis data ringan yang menyediakan sistem manajemen basis data relasional dalam paket kecil dan efisien. SQLite mengatur informasi ke dalam tabel, mirip dengan spreadsheet, sehingga informasi dapat ditemukan dan digunakan dengan mudah. Sebagai alat yang kecil dan efisien, SQLite tidak memerlukan banyak pengaturan atau pemeliharaan, menjadikannya pilihan populer bagi banyak pengembang.

Mengapa Menggunakan SQLite pada flutter?

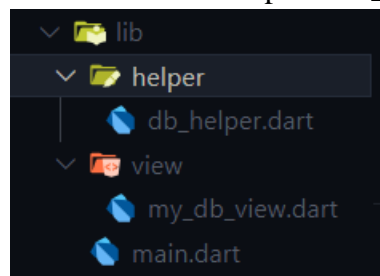
Flutter adalah kerangka kerja pengembangan aplikasi seluler populer yang memungkinkan pembangunan aplikasi lintas platform berkinerja tinggi dengan satu basis kode. Flutter menyediakan pustaka dan alat lengkap yang mendukung pembuatan aplikasi kompleks. Persistensi data adalah aspek penting dari banyak aplikasi seluler, dan di sinilah SQLite berperan. SQLite merupakan pilihan ideal untuk pengembangan aplikasi seluler karena ringan, cepat, dan mendukung transaksi. SQLite juga didukung luas di berbagai platform, termasuk Android, iOS, dan Web.

b. SQL helper

Dalam Flutter, SQL Helper biasanya merujuk pada penggunaan paket seperti sqflite untuk mengelola database SQLite, kelas helper biasanya digunakan untuk menyimpan atau membuat database, tabel, dan melakukan CRUD (create,read,update,delete).

3. Guided

- Instalasi Packet **sqflite** dan **path**
- Untuk menambahkan depedensi pada project bisa dengan klik CTRL+SHIFT+P > add depedency > sqflite, lalu klik. Begitu juga untuk menambahkan depedensi path.
- Membuat folder helper > db_helper.dart dan folder view > my_db_view.dart.



d. File db_helper

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

class DatabaseHelper {
  static final DatabaseHelper _instance = DatabaseHelper._internal();
```

```
static Database? _database;

// factory constructor
factory DatabaseHelper() {
    return _instance;
}
// private constructor
DatabaseHelper._internal();

Future<Database> get database async {
    if (_database != null) return _database!;
    {
        _database = await _initDatabase();
        return _database!;
    }
}
// mendapatkan path untuk database
Future<Database> _initDatabase() async {
    String path = join(
        await getDatabasesPath(), 'my_prakdatabase.db'); // membuka
database
    return await openDatabase(
        path,
        version: 1,
        onCreate: _onCreate,
    );
}

Future<void> _onCreate(Database db, int version) async {
    await db.execute('''
CREATE TABLE my_table(
id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
title TEXT,
description TEXT,
createdAt TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP)
''');
}

Future<int> insert(Map<String, dynamic> row) async {
    Database db = await database;
    return await db.insert('my_table', row);
}

Future<List<Map<String, dynamic>>> queryAllRows() async {
    Database db = await database;
    return await db.query('my_table');
}

Future<int> update(Map<String, dynamic> row) async {
    Database db = await database;
    int id = row['id'];
    return await db.update('my_table', row, where: 'id = ?', whereArgs:
[id]);
}

Future<int> delete(int id) async {
    Database db = await database;
```

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

e. File my_db_view.dart

```
import 'package:flutter/material.dart';
import 'package:gd10/helper/db_helper.dart';

class dbView extends StatefulWidget {
  const dbView({super.key});

  @override
  State<dbView> createState() => _dbViewState();
}

class _dbViewState extends State<dbView> {
  final DatabaseHelper dbHelper = DatabaseHelper();
  List<Map<String, dynamic>> _dbData = [];
  final TextEditingController _titleController =
TextEditingController();
  final TextEditingController _descriptionController =
TextEditingController();
  @override
  void initState() {
    _refreshData();
    super.initState();
  }

  @override
  void dispose() {
    _titleController.dispose();
    _descriptionController.dispose();
    super.dispose();
  }

  void _refreshData() async {
    final data = await dbHelper.queryAllRows();
    setState(() {
      _dbData = data;
    });
  }

  void _addData() async {
    await dbHelper.insert({
      'title': _titleController.text,
      'description': _descriptionController.text
    });
    _titleController.clear();
    _descriptionController.clear();
    _refreshData();
  }

  void _updateData(int id) async {
    await dbHelper.update({
      'id': id,
      'title': _titleController.text,
      'description': _descriptionController.text
    }, id);
  }
}
```

```

    });
    _titleController.clear();
    _descriptionController.clear();
    _refreshData();
  }

  void _deleteData(int id) async {
    await dbHelper.delete(id);
    _refreshData();
  }

  void _showEditDialog(Map<String, dynamic> item) {
    _titleController.text = item['title'];
    _descriptionController.text = item['description'];

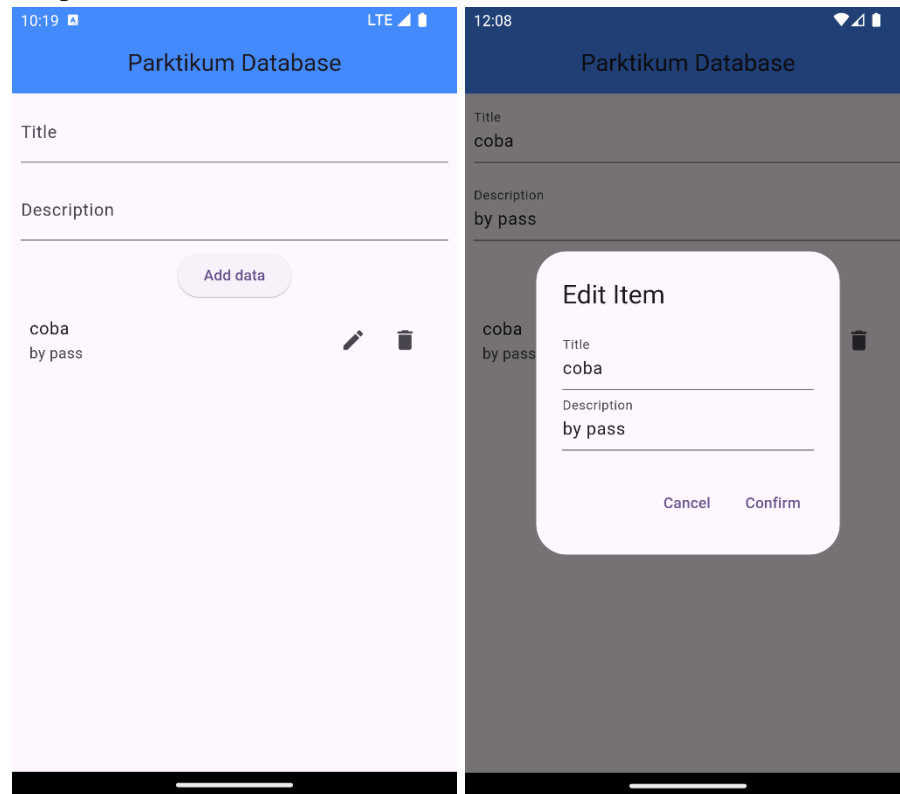
    showDialog(
      context: context,
      builder: (context) {
        return AlertDialog(
          title: Text('Edit Item'),
          content: Column(
            mainAxisAlignment: MainAxisAlignment.min,
            children: [
              TextField(
                controller: _titleController,
                decoration: InputDecoration(labelText: 'Title'),
              ),
              TextField(
                controller: _descriptionController,
                decoration: InputDecoration(labelText:
'Description'),
              ),
            ],
          ),
          actions: [
            TextButton(
              onPressed: () {
                Navigator.of(context).pop();
              },
              child: Text('Cancel')),
            TextButton(
              onPressed: () {
                _updateData(item['id']);
                Navigator.of(context).pop();
              },
              child: Text('Confirm'))
          ],
        );
      }
    );
  }

  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Parktikum Database"),
        centerTitle: true,
        backgroundColor: Colors.blueAccent,

```

```
    ),
    body: Column(
      children: [
        Padding(
          padding: const EdgeInsets.all(8.0),
          child: TextField(
            controller: _titleController,
            decoration: InputDecoration(labelText: 'Title'),
          ),
        ),
        Padding(
          padding: const EdgeInsets.all(8.0),
          child: TextField(
            controller: _descriptionController,
            decoration: InputDecoration(labelText: 'Description'),
          ),
        ),
        ElevatedButton(
          onPressed: _addData,
          child: Text('Add data'),
        ),
        Expanded(
          child: ListView.builder(
            itemCount: _dbData.length,
            itemBuilder: (context, index) {
              final item = _dbData[index];
              return ListTile(
                title: Text(item['title']),
                subtitle: Text(item['description']),
                trailing: Row(
                  mainAxisAlignment: MainAxisAlignment.min,
                  children: [
                    IconButton(
                      onPressed: () => _showEditDialog(item),
                      icon: Icon(Icons.edit),
                    ),
                    IconButton(
                      onPressed: () => _deleteData(item['id']),
                      icon: Icon(Icons.delete),
                    ),
                  ],
                ),
              );
            },
          ),
        ),
      ],
    ),
  ),
);
}
```

Output:



4. Unguided

(Soal) Buatlah sebuah project aplikasi Flutter dengan SQLite untuk menyimpan data biodata mahasiswa yang terdiri dari nama, NIM, domisili, dan hobi. Data yang dimasukkan melalui form akan ditampilkan dalam daftar di halaman utama. Alur Aplikasi:

- Form Input: Buat form input untuk menambahkan biodata mahasiswa, dengan kolom: - Nama - Nim - Alamat – Hobi
- Tampilkan Daftar Mahasiswa: Setelah data berhasil ditambahkan, tampilkan daftar semua data mahasiswa yang sudah disimpan di halaman utama.
- Implementasikan fitur Create (untuk menyimpan data mahasiswa) dan Read (untuk menampilkan daftar mahasiswa yang sudah disimpan).

a. File db_helper.dart

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

class DatabaseHelper {
  DatabaseHelper._privateConstructor();
  static final DatabaseHelper instance = DatabaseHelper._privateConstructor();

  static Database? _database;

  Future<Database> get database async {
    if (_database != null) return _database!;
    _database = await _initDatabase();
    return _database!;
  }
}
```

```

Future<Database> _initDatabase() async {
  String path = join(await getDatabasesPath(), 'students.db');
  return await openDatabase(
    path,
    version: 1,
    onCreate: (db, version) {
      return db.execute('''
        CREATE TABLE students (
          id INTEGER PRIMARY KEY AUTOINCREMENT,
          nama TEXT,
          nim TEXT,
          alamat TEXT,
          hobi TEXT
        )
      ''');
    },
  );
}

Future<int> insertStudent(Map<String, dynamic> student) async {
  Database db = await instance.database;
  return await db.insert('students', student);
}

Future<List<Map<String, dynamic>>> getStudents() async {
  Database db = await instance.database;
  return await db.query('students', orderBy: 'id');
}
}

```

b. File mahasiswa.dart

```

import 'package:flutter/material.dart';
import 'db_helper.dart';

class AddStudentPage extends StatefulWidget {
  const AddStudentPage({Key? key}) : super(key: key);

  @override
  State<AddStudentPage> createState() => _AddStudentPageState();
}

class _AddStudentPageState extends State<AddStudentPage> {
  final _formKey = GlobalKey<FormState>();
  final _nameController = TextEditingController();
  final _nimController = TextEditingController();
  final _addressController = TextEditingController();
  final _hobbyController = TextEditingController();

  void _saveStudent() async {
    if (_formKey.currentState!.validate()) {
      await DatabaseHelper.instance.insertStudent({
        'nama': _nameController.text,
        'nim': _nimController.text,
        'alamat': _addressController.text,
        'hobi': _hobbyController.text,
      });
      Navigator.pop(context, true); // Return to previous page
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(

```



```

    appBar: AppBar(
      backgroundColor: const Color.fromRGBO(82, 170, 94, 1.0),
      centerTitle: true,
      title: const Text('Tambah Biodata Mahasiswa'),
    ),
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Form(
        key: _formKey,
        child: Column(
          children: [
            TextFormField(
              controller: _nameController,
              decoration: const InputDecoration(labelText: 'Nama'),
              validator: (value) => value!.isEmpty ? 'Nama tidak boleh
kosong' : null,
            ),
            TextFormField(
              controller: _nimController,
              decoration: const InputDecoration(labelText: 'NIM'),
              validator: (value) => value!.isEmpty ? 'NIM tidak boleh
kosong' : null,
            ),
            TextFormField(
              controller: _addressController,
              decoration: const InputDecoration(labelText: 'Alamat'),
              validator: (value) => value!.isEmpty ? 'Alamat tidak boleh
kosong' : null,
            ),
            TextFormField(
              controller: _hobbyController,
              decoration: const InputDecoration(labelText: 'Hobi'),
              validator: (value) => value!.isEmpty ? 'Hobi tidak boleh
kosong' : null,
            ),
            const SizedBox(height: 20),
            ElevatedButton(
              onPressed: _saveStudent,
              child: const Text('Simpan'),
            ),
          ],
        ),
      ),
    ),
  );
}

```

c. File home.dart

```

import 'package:flutter/material.dart';
import 'db_helper.dart';
import 'mahasiswa.dart';

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  List<Map<String, dynamic>> _students = [];

```

```

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

void _loadStudents() async {
  final data = await DatabaseHelper.instance.getStudents();
  setState(() {
    _students = data;
  });
}

void _navigateToAddPage() async {
  final result = await Navigator.push(
    context,
    MaterialPageRoute(builder: (context) => const AddStudentPage()),
  );
  if (result == true) _loadStudents();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: const Color.fromRGB(82, 170, 94, 1.0),
      centerTitle: true,
      title: const Text('SQLite Biodata Mahasiswa'),
    ),
    body: _students.isEmpty
      ? const Center(
        child: Text('Belum ada data mahasiswa.'),
      )
      : ListView.builder(
        itemCount: _students.length,
        itemBuilder: (context, index) {
          final student = _students[index];
          return Card(
            margin: const EdgeInsets.all(8.0),
            child: ListTile(
              title: Text(student['nama']),
              subtitle: Text(
                'NIM: ${student['nim']}\nAlamat:
                ${student['alamat']}\nHobi: ${student['hobi']}',
              ),
            ),
          );
        },
      ),
    floatingActionButton: FloatingActionButton(
      onPressed: _navigateToAddPage,
      child: const Icon(Icons.add),
      backgroundColor: const Color.fromRGB(82, 170, 94, 1.0),
    ),
  );
}

```

d. File main.dart

```

import 'package:biodata_mhs/home.dart';
import 'package:flutter/material.dart';
// import 'db_helper.dart';

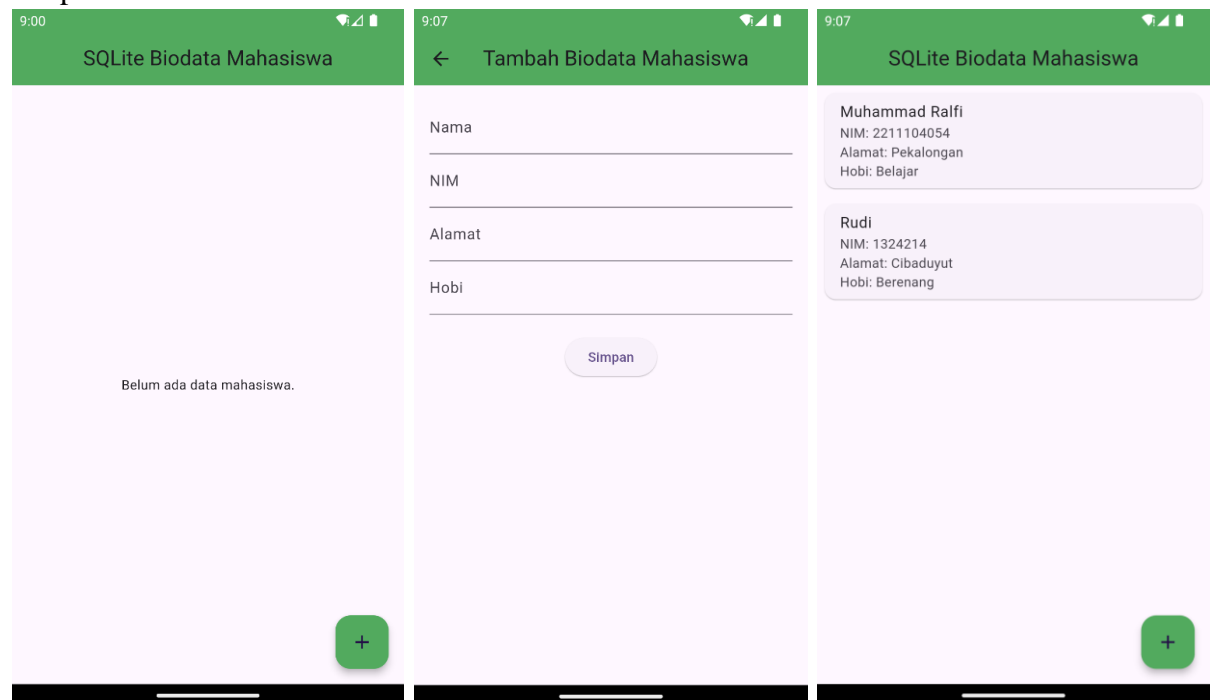
```

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

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const HomePage(),
    );
  }
}
```

e. Output



5. Kesimpulan

SQLite adalah mesin basis data relasional yang ringan dan efisien, ideal untuk aplikasi seluler karena mudah digunakan dan tidak memerlukan konfigurasi rumit. Dalam pengembangan aplikasi dengan Flutter, SQLite mendukung penyimpanan data yang cepat dan stabil di berbagai platform seperti Android dan iOS. Penggunaan paket seperti `sqflite` sebagai SQL Helper mempermudah pengelolaan database dengan menyediakan fungsi untuk membuat, membaca, memperbarui, dan menghapus data. Dengan ini, pengembang dapat fokus pada pengembangan aplikasi tanpa harus mengkhawatirkan kompleksitas manajemen database.