

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL 10
DATA STORAGE (BAGIAN 1)**



Disusun Oleh :

Aji Prasetyo Nugroho / 2211104049

S1SE-06-2

Asisten Praktikum :

Muhammad Faza Zulian Gesit Al Barru

Aisyah Hasna Aulia

Dosen Pengampu :

Yudha Islami Sulistya

**PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

A. GUIDED

- db_helper.dart

Source code :

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

// Kelas DatabaseHelper untuk mengelola database
class DatabaseHelper {
  // Singleton instance
  static final DatabaseHelper _instance = DatabaseHelper._internal();
  static Database? _database;

  // Factory constructor untuk mengembalikan singleton instance
  factory DatabaseHelper() => _instance;

  // Private constructor
  DatabaseHelper._internal();

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

  // Inisialisasi database
  Future<Database> _initDatabase() async {
    // Mendapatkan path untuk database
    String path = join(await getDatabasesPath(), 'my_ajiprstdatabase.db');

    // Membuka atau membuat database
    return await openDatabase(
      path,
      version: 1,
      onCreate: _onCreate,
    );
  }

  // Membuat tabel pada saat database pertama kali dibuat
  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
      )
    ''');
  }

  // Insert data ke tabel
  Future<int> insert(Map<String, dynamic> row) async {
    final db = await database;
    return await db.insert('my_table', row);
  }

  // Query semua data dari tabel
  Future<List<Map<String, dynamic>>> queryAllRows() async {
    final db = await database;
    return await db.query('my_table');
  }

  // Update data dalam tabel berdasarkan ID
  Future<int> update(Map<String, dynamic> row) async {
    final db = await database;
    final id = row['id'] as int;
    return await db.update('my_table', row, where: 'id = ?', whereArgs: [id]);
  }

  // Hapus data dari tabel berdasarkan ID
  Future<int> delete(int id) async {
    final db = await database;
    return await db.delete('my_table', where: 'id = ?', whereArgs: [id]);
  }
}
```

- my_db_view.dart

Source Code :

```
import 'package:flutter/material.dart';
import 'package:pertemuan_10/helper/db_helper.dart';
import 'package:sqflite/sqflite.dart';

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

  @override
  State<MyDatabaseView> createState() => _MyDatabaseViewState();
}

class _MyDatabaseViewState extends State<MyDatabaseView> {
  final DatabaseHelper dbHelper = DatabaseHelper();
  List<Map<String, dynamic>> _dbData = [];
  final TextEditingController _titleController = TextEditingController();
  final TextEditingController _descriptionController = TextEditingController();

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

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

  // Metode untuk memperbarui data dari database
  void _refreshData() async {
    final data = await dbHelper.queryAllRows();
    setState(() {
      _dbData = data;
    });
  }

  // Metode untuk menambah data ke database
  void _addData() async {
    if (_titleController.text.isEmpty || _descriptionController.text.isEmpty)
      return;
    await dbHelper.insert(
      {'title': _titleController.text,
       'description': _descriptionController.text},
    );
    _titleController.clear();
    _descriptionController.clear();
    _refreshData();
  }

  // Metode untuk memperbarui data dalam database
  void _updateData(int id) async {
    if (_titleController.text.isEmpty || _descriptionController.text.isEmpty)
      return;
    await dbHelper.update(
      {'id': id,
       'title': _titleController.text,
       'description': _descriptionController.text},
    );
    _titleController.clear();
    _descriptionController.clear();
    _refreshData();
  }

  // Metode untuk menghapus data dari database
  void _deleteData(int id) async {
    await dbHelper.delete(id);
    _refreshData();
  }

  // Menampilkan dialog untuk mengedit data
  void _showEditDialog(Map<String, dynamic> item) {
    _titleController.text = item['title'];
    _descriptionController.text = item['description'];

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Praktikum Database - sqflite'),
        backgroundColor: Colors.green,
      ),
      body: Column(
        children: [
          Padding(
            padding: const EdgeInsets.all(8.0),
            child: TextField(
              controller: _titleController,
              decoration: const InputDecoration(
                labelText: 'Title',
              ),
            ),
          ),
          Padding(
            padding: const EdgeInsets.all(8.0),
            child: TextField(
              controller: _descriptionController,
              decoration: const InputDecoration(
                labelText: 'Description',
              ),
            ),
          ),
          ElevatedButton(
            onPressed: _addData,
            child: const 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(
                        icon: const Icon(Icons.edit),
                        onPressed: () => _showEditDialog(item),
                      ),
                      IconButton(
                        icon: const Icon(Icons.delete),
                        onPressed: () => _deleteData(item['id']),
                      ),
                    ],
                  ),
                ),
              ),
            ),
          ),
        ],
      ),
    );
  }
}
```

- main.dart

Source Code :

```
import 'package:flutter/material.dart';
import 'package:pertemuan_10/view/my_db_view.dart';

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: MyDatabaseView());
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});

  final String title;

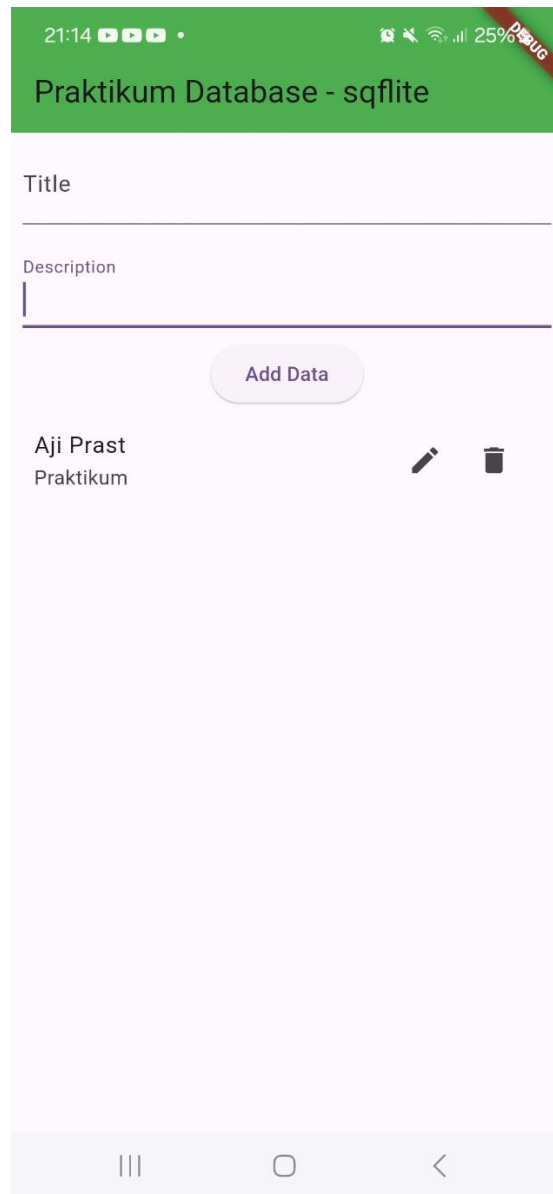
  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            const Text(
              'You have pushed the button this many times:',
            ),
            Text(
              '$_counter',
              style: Theme.of(context).textTheme.headlineMedium,
            ),
          ],
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment',
        child: const Icon(Icons.add),
      ),
    );
  }
}
```

Output :



Deskripsi Program

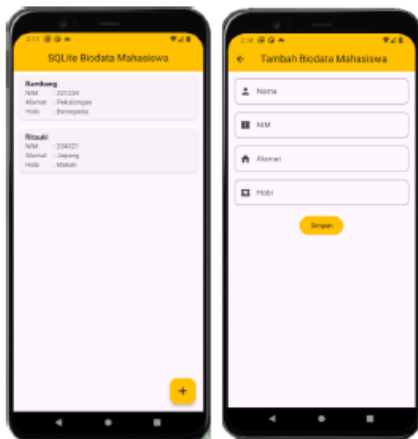
Program ini adalah sebuah aplikasi Flutter yang menggunakan SQLite untuk mengelola biodata mahasiswa. Operasinya meliputi penyimpanan data (Create) dan penampilan data (Read). Aplikasi ini terdiri dari beberapa komponen utama: `db_helper.dart` sebagai pengelola basis data untuk membuat tabel, menyimpan data, dan membaca data dari SQLite; `main.dart` digunakan untuk menjalankan aplikasi; `home_page.dart` bertugas menampilkan daftar mahasiswa dengan desain kotak yang rapi menggunakan widget `Card`; sedangkan `add_biodata_page.dart` berfungsi sebagai formulir input untuk menambahkan biodata baru, yang terdiri dari nama, NIM, alamat, dan hobi mahasiswa. Antarmuka aplikasi didesain dengan `AppBar` berwarna hijau dan tombol tambah yang juga berwarna hijau untuk menciptakan pengalaman pengguna yang nyaman dan menarik.

B. Unguided

1. (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).
- Contoh output:



Source Code :

- a. add_biodata_page.dart

Source code :

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

class AddBiodataPage extends StatelessWidget {
  final TextEditingController namaController = TextEditingController();
  final TextEditingController nimController = TextEditingController();
  final TextEditingController alamatController = TextEditingController();
  final TextEditingController hobiController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Tambah Biodata Mahasiswa')),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          children: [
            TextField(
              controller: namaController,
              decoration: InputDecoration(labelText: 'Nama'),
            ),
            TextField(
              controller: nimController,
              decoration: InputDecoration(labelText: 'NIM'),
            ),
            TextField(
              controller: alamatController,
              decoration: InputDecoration(labelText: 'Alamat'),
            ),
            TextField(
              controller: hobiController,
              decoration: InputDecoration(labelText: 'Hobi'),
            ),
            SizedBox(height: 20),
            ElevatedButton(
              onPressed: () async {
                final data = {
                  'nama': namaController.text,
                  'nim': nimController.text,
                  'alamat': alamatController.text,
                  'hobi': hobiController.text,
                };
                await DatabaseHelper().insertBiodata(data);
                Navigator.pop(context);
              },
              child: Text('Simpan'),
            ),
          ],
        ),
      ),
    );
  }
}
```

- b. biodata.dart

Source code :

```
class Biodata {
  int? id;
  String nama;
  String nim;
  String alamat;
  String hobi;

  Biodata({
    this.id,
    required this.nama,
    required this.nim,
    required this.alamat,
    required this.hobi,
  });

  Map<String, dynamic> toMap() {
    return {
      'id': id,
      'nama': nama,
      'nim': nim,
      'alamat': alamat,
      'hobi': hobi,
    };
  }

  factory Biodata.fromMap(Map<String, dynamic> map) {
    return Biodata(
      id: map['id'],
      nama: map['nama'],
      nim: map['nim'],
      alamat: map['alamat'],
      hobi: map['hobi'],
    );
  }
}
```


Source code :

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

class DatabaseHelper {
  static final DatabaseHelper _instance = DatabaseHelper._internal();
  factory DatabaseHelper() => _instance;
  static Database? _database;

  DatabaseHelper._internal();

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

  Future<Database> _initDatabase() async {
    final databasePath = await getDatabasesPath();
    final path = join(databasePath, 'biodata_mahasiswa.db');

    return await openDatabase(
      path,
      version: 1,
      onCreate: (db, version) async {
        await db.execute('''
          CREATE TABLE biodata (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            nama TEXT,
            nim TEXT,
            alamat TEXT,
            hobi TEXT
          )
        ''');
      },
    );
  }

  Future<int> insertBiodata(Map<String, dynamic> data) async {
    final db = await database;
    return await db.insert('biodata', data);
  }

  Future<List<Map<String, dynamic>>> getBiodata() async {
    final db = await database;
    return await db.query('biodata');
  }
}
```

Source code :

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

class HomePage extends StatefulWidget {
  @override
  _HomePageState createState() => _HomePageState();
}

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


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

  Future<void> _fetchBiodata() async {
    final data = await DatabaseHelper().getBiodata();
    setState(() {
      _biodataList = data;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('SQLite Biodata Mahasiswa'),
        backgroundColor: Colors.green,
      ),
      body: _biodataList.isEmpty
        ? Center(
            child: Text(
              'Belum ada data!',
              style: TextStyle(fontSize: 18),
            ),
          )
        : ListView.builder(
            padding: EdgeInsets.all(16.0),
            itemCount: _biodataList.length,
            itemBuilder: (context, index) {
              final biodata = _biodataList[index];
              return Card(
                margin: EdgeInsets.only(bottom: 16.0),
                elevation: 4,
                shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(8.0),
                ),
                child: Padding(
                  padding: const EdgeInsets.all(16.0),
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      Text(
                        'Nama: ${biodata['nama']}',
                        style: TextStyle(
                          fontSize: 16,
                          fontWeight: FontWeight.bold,
                        ),
                      ),
                      SizedBox(height: 8),
                      Text(
                        'NIM: ${biodata['nim']}',
                        style: TextStyle(fontSize: 14),
                      ),
                      SizedBox(height: 8),
                      Text(
                        'Alamat: ${biodata['alamat']}',
                        style: TextStyle(fontSize: 14),
                      ),
                      SizedBox(height: 8),
                      Text(
                        'Hobi: ${biodata['hobi']}',
                        style: TextStyle(fontSize: 14),
                      ),
                    ],
                  ),
                ),
              ),
            ),
          ),
      floatingActionButton: FloatingActionButton(
        onPressed: () async {
          await Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => AddBiodataPage()),
          );
          _fetchBiodata();
        },
        backgroundColor: Colors.green,
        child: Icon(Icons.add),
      ),
    );
  }
}
```

e. main.dart

Source code :



```
import 'package:flutter/material.dart';
import 'home_page.dart';

void main() {
  runApp(MaterialApp(
    debugShowCheckedModeBanner: false,
    home: HomePage(),
  ));
}
```

Output :

20:52

25%

← Tambah Biodata Mahasiswa

Nama

Aji Prasetyo Nugroho

NIM

2211104049

Alamat

Gumelar

Hobi

Baca Buku

Simpan

20:52



25%

SQLite Biodata Mahasiswa

Nama: Aji Prasetyo Nugroho

NIM: 2211104049

Alamat: Gumelar

Hobi: Baca buku



Penjelasan Program

Program ini merupakan aplikasi Flutter yang memanfaatkan SQLite untuk menyimpan dan menampilkan informasi biodata mahasiswa. Aplikasi ini terdiri dari beberapa halaman, termasuk halaman utama yang menampilkan daftar biodata dalam format yang teratur menggunakan Card, serta halaman formulir untuk menambahkan data baru. Setiap entri biodata mahasiswa mencakup informasi seperti Nama, NIM, Alamat, dan Hobi, yang disimpan dalam database lokal SQLite melalui fungsi CRUD (Create dan Read) yang dikelola oleh helper database. Tampilan aplikasi juga menghadirkan AppBar berwarna hijau dan tombol tambah (FloatingActionButton) yang serasi dalam warna, menawarkan antarmuka yang menarik dan konsisten.