LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK

MODUL X

DATA STORAGE (BAGIAN I)

Disusun Oleh :

Dhiya Ulhaq Ramadhan 2211104053

SE-06-02

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

GUIDED

- **lib/main.dart**

lib > main.dart

```dart
import 'package:flutter/material.dart';
import 'package:data_storage/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: 'Data Storage Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.blue),
        useMaterial3: true,
      ),
      home: const MyDatabaseView(),
    );
  }
}
```

- **lib/view/my_db_view.dart**

lib > view > my_db_view.dart > _MyDatabaseViewState > build

```dart
import 'package:flutter/material.dart';
import 'package:data_storage/helper/db_helper.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() {
    _refreshData();
    super.initState();
  }

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

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

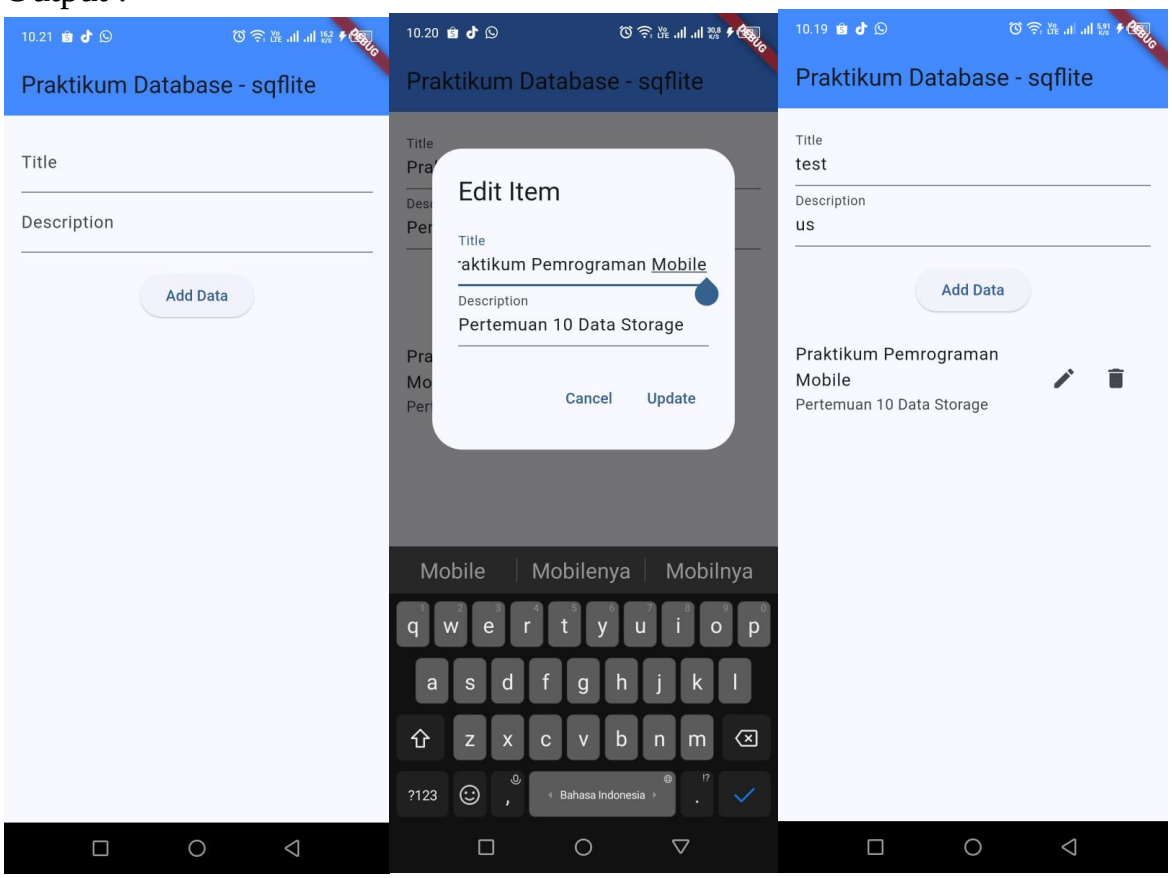
- **lib/helper/db_helper.dart**

lib > helper > db_helper.dart > DatabaseHelper > DatabaseHelper

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

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

  factory DatabaseHelper() {
    return _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_prakdatabase.db');
    //membuka database
    return await openDatabase(
      path,
      version: 1,
      onCreate: _onCreate,
    );
  }
```

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

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

  //update data
  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]);
  }

  //hapus data
  Future<int> delete(int id) async {
    Database db = await database;
    return await db.delete('my_table', where: 'id = ?', whereArgs: [id]);
  }
}
```

Output :



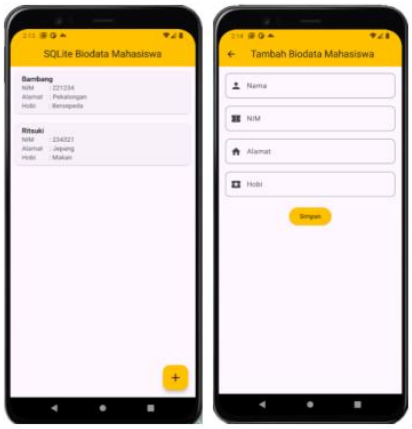
Penjelasan Program

Program yang diajarkan pada pertemuan ini yaitu aplikasi manajemen data menggunakan Flutter dan SQLite yang memungkinkan pengguna melakukan operasi penyimpanan data secara lengkap. Aplikasi ini dirancang untuk membuat, membaca, mengubah, dan menghapus catatan melalui antarmuka mobile yang sederhana dan intuitif. Pada layar utama, terdapat AppBar berwarna biru dengan judul "Praktikum Database - sqflite", di bawahnya terdapat formulir input yang memungkinkan pengguna memasukkan judul dan deskripsi catatan. Tombol "Add Data" memungkinkan pengguna menambahkan catatan baru ke dalam database. Bagian bawah layar menampilkan daftar catatan yang tersimpan dalam format ListTile, dengan setiap item dilengkapi tombol edit dan delete. Ketika tombol edit ditekan, muncul dialog untuk mengubah data, sedangkan tombol delete langsung menghapus catatan dari database. Seluruh operasi data dilakukan menggunakan DatabaseHelper yang mengelola koneksi dan transaksi dengan database SQLite lokal, dengan metode asynchronous untuk menjaga responsivitas antarmuka pengguna.

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:

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



Jawaban :

lib/main.dart

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

Run | Debug | Profile
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'SQLite Biodata Mahasiswa',
      theme: ThemeData(
        primarySwatch: Colors.indigo,
        scaffoldBackgroundColor: Colors.white,
        fontFamily: 'Roboto',
      ), // ThemeData
      home: const StudentListView(),
      debugShowCheckedModeBanner: false,
    ); // MaterialApp
  }
}
```

lib/view/my_db_view.dart

lib > view > my_db_view.dart > _StudentListViewState

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

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

  @override
  State<StudentListView> createState() => _StudentListViewState();
}

class _StudentListViewState extends State<StudentListView> {
  final DatabaseHelper _databaseHelper = DatabaseHelper();
  List<Map<String, dynamic>> _studentList = [];

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

  Future<void> _loadStudents() async {
    final students = await _databaseHelper.getStudents();
    setState(() {
      _studentList = students;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          'Biodata Mahasiswa',
          style: GoogleFonts.poppins(
            fontSize: 20,
            fontWeight: FontWeight.w600,
            color: Colors.white,
          ),
        ), // Text
        backgroundColor: const Color(0xFF5C6BC0),
        elevation: 0,
      ), // AppBar
      body: Container(
        decoration: const BoxDecoration(
          gradient: LinearGradient(
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
            colors: [Color(0xFFE8EAF6), Colors.white],
          ), // LinearGradient
        ), // BoxDecoration
        child: _studentList.isEmpty
            ? const Center(
                child: Text(
                  'Belum ada data mahasiswa',
                  style: TextStyle(
                    fontSize: 16,
                    fontWeight: FontWeight.w500,
                    color: Color(0xFF757575),
                  ), // TextStyle
                ), // Text
              ) // Center
            : ListView.builder(
                padding: const EdgeInsets.all(16),
                itemCount: _studentList.length,
                itemBuilder: (context, index) {
                  return Card(
                    elevation: 4,
                    shape: RoundedRectangleBorder(
                      borderRadius: BorderRadius.circular(12),
                    ), // RoundedRectangleBorder
                    child: Padding(
                      padding: const EdgeInsets.all(16),
                      child: Column(
```

```
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      Text(
                        _studentList[index]['name'] ?? '',
                        style: GoogleFonts.poppins(
                          fontSize: 18,
                          fontWeight: FontWeight.w600,
                          color: const Color(0xFF424242),
                        ),
                      ), // Text
                      const SizedBox(height: 8),
                      _buildInfoRow(
                        Icons.numbers,
                        'NIM',
                        _studentList[index]['nim'] ?? '',
                      ),
                      _buildInfoRow(
                        Icons.location_on,
                        'Alamat',
                        _studentList[index]['address'] ?? '',
                      ),
                      _buildInfoRow(
                        Icons.sports_esports,
                        'Hobi',
                        _studentList[index]['hobby'] ?? '',
                      ),
                    ],
                  ), // Column
                ), // Padding
              ); // Card
            },
          ), // ListView.builder
        ), // Container
        floatingActionButton: FloatingActionButton(
          onPressed: () async {
            await Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => const AddStudentView(),
              ), // MaterialPageRoute
            );
            _loadStudents();
          },
          backgroundColor: const Color(0xFF5C6BC0),
          foregroundColor: Colors.white,
          child: const Icon(Icons.add),
        ), // FloatingActionButton
      ); // Scaffold
    }
```

```dart
  Widget _buildInfoRow(IconData icon, String label, String value) {
    return Padding(
      padding: const EdgeInsets.only(bottom: 8),
      child: Row(
        children: [
          Icon(icon, size: 16, color: const Color(0xFF757575)),
          const SizedBox(width: 8),
          Text(
            '$label: ',
            style: GoogleFonts.poppins(
              fontSize: 14,
              fontWeight: FontWeight.w500,
              color: const Color(0xFF757575),
            ),
          ), // Text
          Expanded(
            child: Text(
              value,
              style: GoogleFonts.poppins(
                fontSize: 14,
                fontWeight: FontWeight.w500,
                color: const Color(0xFF424242),
              ),
            ), // Text
          ), // Expanded
        ],
      ), // Row
    ); // Padding
  }
}

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

  @override
  State<AddStudentView> createState() => _AddStudentViewState();
}

class _AddStudentViewState extends State<AddStudentView> {
  final _formKey = GlobalKey<FormState>();
  final _nameController = TextEditingController();
  final _nimController = TextEditingController();
  final _addressController = TextEditingController();
  final _hobbyController = TextEditingController();
  final DatabaseHelper _databaseHelper = DatabaseHelper();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'Tambah Biodata Mahasiswa',
          style: TextStyle(color: Colors.white),
        ), // Text
        backgroundColor: Colors.indigo,
        elevation: 0,
        iconTheme: const IconThemeData(color: Colors.white),
      ), // AppBar
      body: Container(
        decoration: BoxDecoration(
          gradient: LinearGradient(
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
            colors: [Colors.indigo.shade100, Colors.white],
          ), // LinearGradient
        ), // BoxDecoration
        child: Form(
```

```
key: _formKey,
child: ListView(
  padding: const EdgeInsets.all(16),
  children: [
    _buildTextField(
      controller: _nameController,
      label: 'Nama',
      icon: Icons.person,
      validator: (value) {
        if (value == null || value.isEmpty) {
          return 'Nama tidak boleh kosong';
        }
        return null;
      },
    ),
    const SizedBox(height: 16),
    _buildTextField(
      controller: _nimController,
      label: 'NIM',
      icon: Icons.numbers,
      validator: (value) {
        if (value == null || value.isEmpty) {
          return 'NIM tidak boleh kosong';
        }
        return null;
      },
    ),
    const SizedBox(height: 16),
    _buildTextField(
      controller: _addressController,
      label: 'Alamat',
      icon: Icons.location_on,
      validator: (value) {
        if (value == null || value.isEmpty) {
          return 'Alamat tidak boleh kosong';
        }
        return null;
```

```
229                 },
230               ),
231               const SizedBox(height: 16),
232               _buildTextField(
233                 controller: _hobbyController,
234                 label: 'Hobi',
235                 icon: Icons.sports_esports,
236                 validator: (value) {
237                   if (value == null || value.isEmpty) {
238                     return 'Hobi tidak boleh kosong';
239                   }
240                   return null;
241                 },
242               ),
243               const SizedBox(height: 24),
244               ElevatedButton(
245                 onPressed: () async {
246                   if (_formKey.currentState!.validate()) {
247                     await _databaseHelper.insertStudent({
248                       'name': _nameController.text,
249                       'nim': _nimController.text,
250                       'address': _addressController.text,
251                       'hobby': _hobbyController.text,
252                     });
253                     Navigator.pop(context);
254                   }
255                 },
256                 style: ElevatedButton.styleFrom(
257                   backgroundColor: Colors.indigo,
258                   padding: const EdgeInsets.symmetric(vertical: 16),
259                   shape: RoundedRectangleBorder(
260                     borderRadius: BorderRadius.circular(8),
261                   ), // RoundedRectangleBorder
262                 ),
263                 child: const Text(
264                   'Simpan',
265                   style: TextStyle(fontSize: 16),
```

```
                ), // Text
              ), // ElevatedButton
            ],
          ), // ListView
        ), // Form
      ), // Container
    ); // Scaffold
  }

  Widget _buildTextField({
    required TextEditingController controller,
    required String label,
    required IconData icon,
    required String? Function(String?) validator,
  }) {
    return TextFormField(
      controller: controller,
      decoration: InputDecoration(
        labelText: label,
        prefixIcon: Icon(icon, color: Colors.indigo),
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(8),
        ), // OutlineInputBorder
        enabledBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(8),
          borderSide: BorderSide(color: Colors.grey.shade400),
        ), // OutlineInputBorder
        focusedBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(8),
          borderSide: const BorderSide(color: Colors.indigo),
        ), // OutlineInputBorder
        filled: true,
        fillColor: Colors.white,
      ), // InputDecoration
      validator: validator,
    ); // TextFormField
```

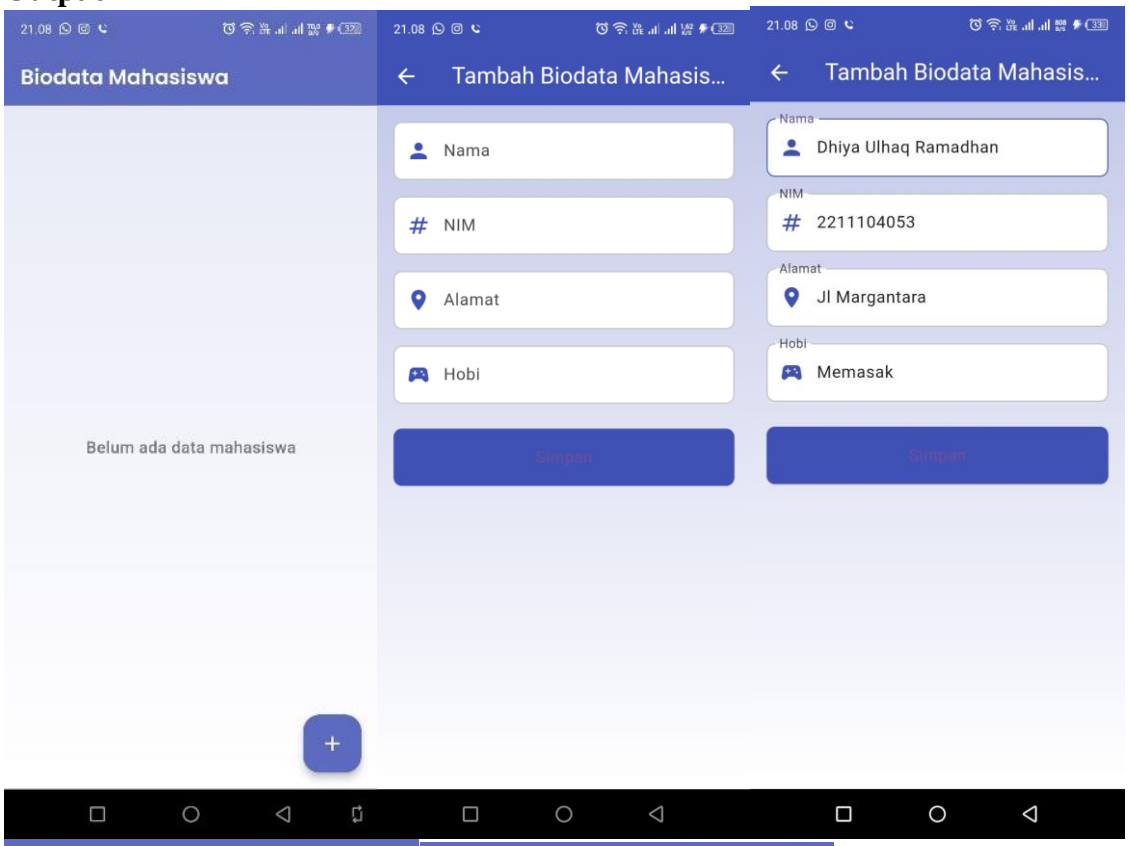
```
  }

  @override
  void dispose() {
    _nameController.dispose();
    _nimController.dispose();
    _addressController.dispose();
    _hobbyController.dispose();
    super.dispose();
  }
}
```

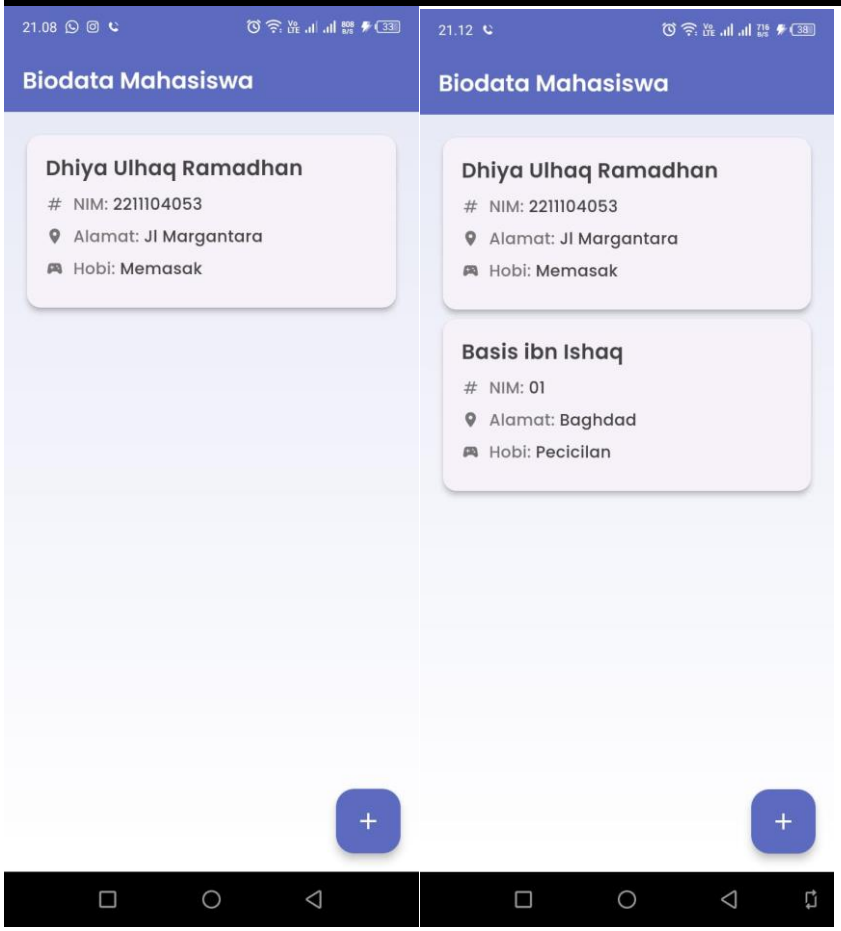
lib/helper/db_helper.dart

lib > helper > db_helper.dart > DatabaseHelper > _initializeDB

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

class DatabaseHelper {
  static DatabaseHelper? _databaseHelper;
  static late Database _database;

  DatabaseHelper._internal() {
    _databaseHelper = this;
  }

  factory DatabaseHelper() => _databaseHelper ?? DatabaseHelper._internal();

  Future<Database> get database async {
    _database = await _initializeDB();
    return _database;
  }

  static const String _tableName = 'students';

  Future<Database> _initializeDB() async {
    var db = openDatabase(
      join(await getDatabasesPath(), 'student_db.db'),
      onCreate: (db, version) async {
        await db.execute(
          '''CREATE TABLE $_tableName(
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            name TEXT,
            nim TEXT,
            address TEXT,
            hobby TEXT
          )''',
        );
      },
      version: 1,
    );
    return db;
  }

  Future<void> insertStudent(Map<String, dynamic> row) async {
    final Database db = await database;
    await db.insert(_tableName, row);
  }

  Future<List<Map<String, dynamic>>> getStudents() async {
    final Database db = await database;
    return await db.query(_tableName);
  }
}
```

Output


Biodata Mahasiswa
Belum ada data mahasiswa
Tambah Biodata Mahasis...
Nama
NIM
Alamat
Hobi
Simpan
Tambah Biodata Mahasis...
Nama
Dhiya Ulhaq Ramadhan
NIM
2211104053
Alamat
Jl Margantara
Hobi
Memasak
Simpan


Biodata Mahasiswa
Dhiya Ulhaq Ramadhan
NIM: 2211104053
Alamat: Jl Margantara
Hobi: Memasak
Biodata Mahasiswa
Dhiya Ulhaq Ramadhan
NIM: 2211104053
Alamat: Jl Margantara
Hobi: Memasak
Basis ibn Ishaq
NIM: 01
Alamat: Baghdad
Hobi: Pecicilan

Penjelasan Program
Saat program dijalankan, main.dart akan dieksekusi dan menjalankan widget MyApp. MyApp merupakan StatelessWidget yang mengatur tema aplikasi menggunakan Material Design dan mengarahkan ke halaman utama StudentListView.
StudentListView adalah StatefulWidget yang bertugas menampilkan daftar mahasiswa. Dalam state-nya (_StudentListViewState), terdapat DatabaseHelper untuk mengelola operasi database. Saat initState() dipanggil, method _loadStudents() dijalankan untuk mengambil data mahasiswa dari database.
DatabaseHelper sendiri merupakan kelas yang mengelola koneksi dan operasi pada database SQLite. Saat pertama kali diakses, database akan diinisialisasi dengan nama 'student_db.db' dan membuat tabel 'students' yang memiliki kolom id, name, nim, address, dan hobby.

Tampilan utama aplikasi terdiri dari AppBar berwarna biru dengan judul "Biodata Mahasiswa" dan daftar mahasiswa yang ditampilkan menggunakan ListView.builder. Setiap item mahasiswa ditampilkan dalam bentuk Card yang menunjukkan nama, NIM, alamat, dan hobi. Jika belum ada data mahasiswa, akan ditampilkan pesan "Belum ada data mahasiswa".
Di bagian bawah layar terdapat FloatingActionButton berwarna biru yang memungkinkan pengguna untuk menambahkan mahasiswa baru. Ketika tombol ditekan, Navigator.push() akan mengarahkan ke halaman AddStudentView.
AddStudentView adalah StatefulWidget yang bertugas menampilkan formulir untuk menambahkan mahasiswa baru. Formulir ini terdiri dari input untuk Nama, NIM, Alamat, dan Hobi. Saat pengguna mengisi formulir dan menekan tombol "Simpan", method _databaseHelper.insertStudent() akan dipanggil untuk menyimpan data baru ke database. Setelah data disimpan, Navigator.pop() akan menutup halaman AddStudentView dan kembali ke halaman StudentListView.
Setiap kali terjadi perubahan data (penambahan mahasiswa baru), method _loadStudents() akan dipanggil untuk memperbarui tampilan daftar mahasiswa.