LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK

MODUL X

DATA STORAGE (BAGIAN 1

Disusun Oleh :

Muhammad Abdul Aziz / 2211104026

SE0601

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

File helper/db_helper.dart :

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

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

  // factory constructor untuk mengembalikan instance singleton
  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!;
    }
  }

  // inisiasi 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,
    );
  }

  //membuat tabel saat db 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)
  ''');
  }

  // metode memasukan data ke dalam tabel
  Future<int> insert(Map<String, dynamic> row) async {
    Database db = await database;
    int result = await db.insert('my_table', row);
    print('Inserted row: $result');  // Log untuk memeriksa hasil insert
    return result;
  }

  // metode mengambil semua data dari tabel
  Future<List<Map<String, dynamic>>> queryAllRows() async {
    Database db = await database;
    List<Map<String, dynamic>> result = await db.query('my_table');
    print('Fetched data: $result');  // Log untuk memeriksa data yang diambil
    return result;
  }

  // metode untuk memperbarui data dalam tabel
  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]);
  }

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

File view/my_db_view.dart :

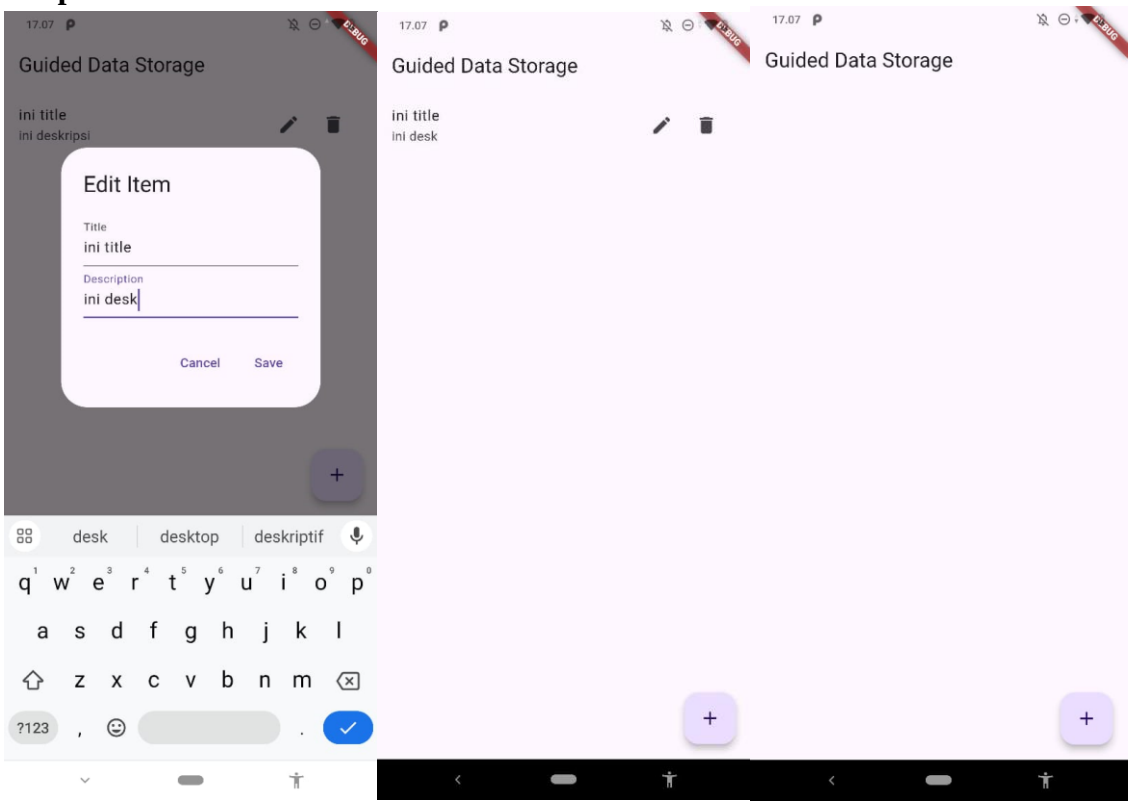
```dart
import 'package:flutter/material.dart';
import 'package:guided/helper/db_helper.dart'; // Pastikan path-nya benar

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();
  }

  // Ubah menjadi Future<void> dan beri async
  Future<void> _refreshData() async {
    final data = await dbHelper.queryAllRows();
    print(data); // Tambahkan log ini
    setState(() {
      _dbData = data;
    });
  }

  // Ubah menjadi Future<void> dan beri async
  Future<void> _addData() async {
    await dbHelper.insert({
      'title': _titleController.text,
      'description': _descriptionController.text,
    });
    _titleController.clear();
    _descriptionController.clear();
    await _refreshData(); // Pastikan menunggu refresh data setelah insert
  }

  // Ubah menjadi Future<void> dan beri async
  Future<void> _updateData(int id) async {
    await dbHelper.update({
      'id': id,
      'title': _titleController.text,
      'description': _descriptionController.text,
    });
    _titleController.clear();
    _descriptionController.clear();
    await _refreshData(); // Pastikan menunggu refresh data setelah update
  }

  // Ubah menjadi Future<void> dan beri async
  Future<void> _deleteData(int id) async {
    await dbHelper.delete(id);
    await _refreshData(); // Pastikan menunggu refresh data setelah delete
  }

  // 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(
            mainAxisSize: MainAxisSize.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: const Text('Save'),
            ),
          ],
        );
      },
    );
  }
```

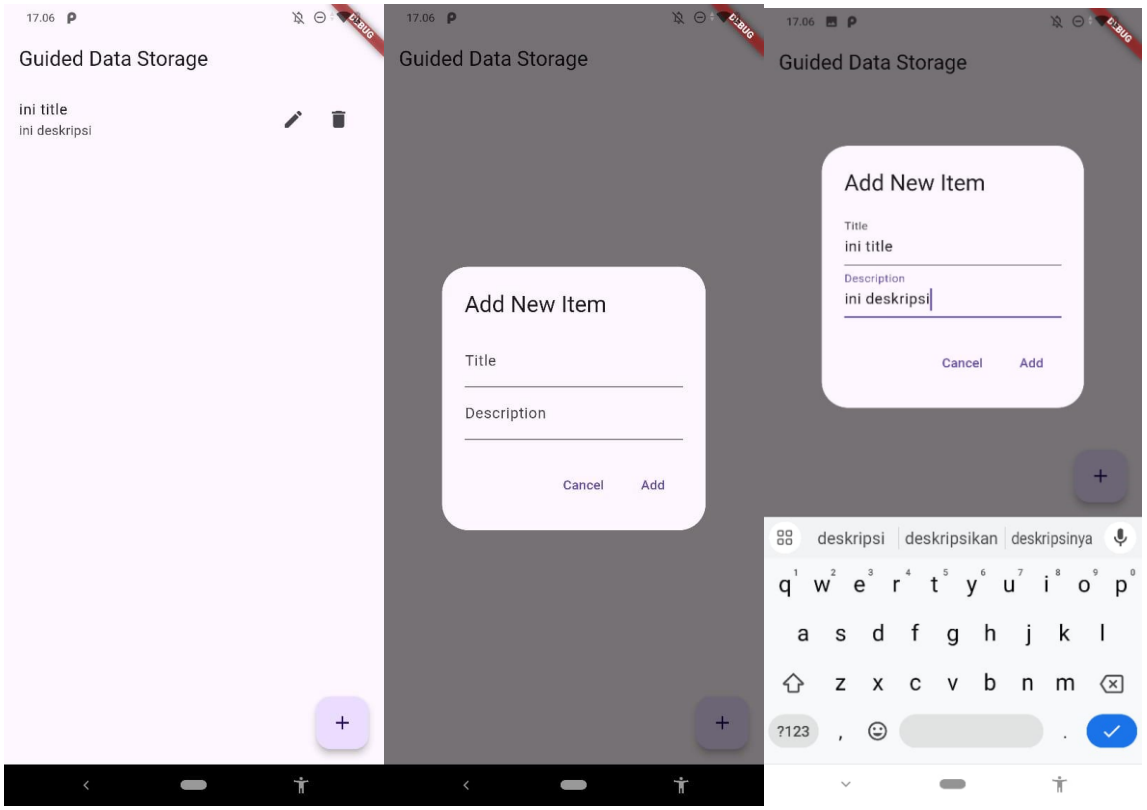
```dart

  // Menampilkan dialog untuk menambahkan data
  void _showAddDialog() {
    _titleController.clear();
    _descriptionController.clear();

    showDialog(
      context: context,
      builder: (context) {
        return AlertDialog(
          title: Text('Add New Item'),
          content: Column(
            mainAxisSize: MainAxisSize.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: () {
                _addData();
                Navigator.of(context).pop();
              },
              child: const Text('Add'),
            ),
          ],
        );
      },
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Guided Data Storage'),
      ),
      body: ListView.builder(
        itemCount: _dbData.length,
        itemBuilder: (context, index) {
          final item = _dbData[index];
          print('Displaying item: ${item['title']}');
          return ListTile(
            title: Text(item['title'] ?? 'No Title'),
            subtitle: Text(item['description'] ?? 'No Description'),
            trailing: Row(
              mainAxisSize: MainAxisSize.min,
              children: [
                IconButton(
                  icon: Icon(Icons.edit),
                  onPressed: () => _showEditDialog(item),
                ),
                IconButton(
                  icon: Icon(Icons.delete),
                  onPressed: () => _deleteData(item['id']),
                ),
              ],
            ),
          );
        },
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _showAddDialog,
        child: Icon(Icons.add),
      ),
    );
  }
}

```

File main.dart :

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Guided Data Storage',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: MyDatabaseView(),
    );
  }
}
```

Output :





Penjelasan :

Kode ini adalah aplikasi Flutter dengan integrasi SQLite untuk pengelolaan data menggunakan paket `sqflite`. `DatabaseHelper` adalah kelas singleton untuk mengelola database SQLite, menyediakan fungsi CRUD (Create, Read, Update, Delete) pada tabel `my_table`. `MyDatabaseView` adalah widget utama dengan antarmuka pengguna untuk menampilkan data dari database dalam bentuk daftar, serta menyediakan fitur untuk menambah, mengedit, dan menghapus data melalui dialog input. Komponen utama aplikasi meliputi pengaturan database, logika pengelolaan data, serta antarmuka berbasis Flutter. Aplikasi ini diinisialisasi melalui `main()` yang memuat `MyDatabaseView`sebagai halaman utama.

UNGUIDED

File helper/db_helper.dart :

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

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

  factory DatabaseHelper() => _instance;

  DatabaseHelper._internal();

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

  Future<Database> _initDatabase() async {
    String path = join(await getDatabasesPath(), 'my_database.db');
    return await openDatabase(
      path,
      version: 1,
      onCreate: _onCreate,
    );
  }

  Future<void> _onCreate(Database db, int version) async {
    await db.execute('''
      CREATE TABLE mahasiswa(
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        nama TEXT,
        nim TEXT,
        alamat TEXT,
        hobi TEXT
      )
    ''');
  }

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

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

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

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

```

File view/add_mahasiswa.dart

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

class AddMahasiswaView extends StatefulWidget {
  final Function refreshData;

  const AddMahasiswaView({Key? key, required this.refreshData})
      : super(key: key);

  @override
  _AddMahasiswaViewState createState() => _AddMahasiswaViewState();
}

class _AddMahasiswaViewState extends State<AddMahasiswaView> {
  final _namaController = TextEditingController();
  final _nimController = TextEditingController();
  final _alamatController = TextEditingController();
  final _hobiController = TextEditingController();
  final dbHelper = DatabaseHelper();

  void addMahasiswa() async {
    await dbHelper.insert({
      'nama': _namaController.text,
      'nim': _nimController.text,
      'alamat': _alamatController.text,
      'hobi': _hobiController.text,
    });
    widget.refreshData();
    Navigator.pop(context);
  }

  // Method to build a text field with icon
  Widget buildTextField({
    required TextEditingController controller,
    required String label,
    required IconData icon,
  }) {
    return TextField(
      controller: controller,
      decoration: InputDecoration(
        labelText: label,
        prefixIcon: Icon(icon),
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(8),
        ),
      ),
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'Tambah Mahasiswa',
          style: TextStyle(fontWeight: FontWeight.bold, color: Colors.white),
        ),
        centerTitle: true,
        backgroundColor: Colors.brown,
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          children: [
            buildTextField(
              controller: _namaController,
              label: 'Nama',
              icon: Icons.person,
            ),
            const SizedBox(height: 16),
            buildTextField(
              controller: _nimController,
              label: 'NIM',
              icon: Icons.book_outlined,
            ),
            const SizedBox(height: 16),
            buildTextField(
              controller: _alamatController,
              label: 'Alamat',
              icon: Icons.location_on,
            ),
            const SizedBox(height: 16),
            buildTextField(
              controller: _hobiController,
              label: 'Hobi',
              icon: Icons.try_sms_star,
            ),
            const SizedBox(height: 20),
            ElevatedButton(
              onPressed: addMahasiswa,
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.brown,
                padding:
                    const EdgeInsets.symmetric(vertical: 12, horizontal: 20),
                shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(8),
                ),
              ),
              child: const Text(
                'Simpan',
                style: TextStyle(
                    fontSize: 16, color: Color.fromARGB(255, 255, 255, 255)),
              ),
            ),
          ],
        ),
      ),
    );
  }
}

```

File view/main_view.dart

```dart
import 'package:flutter/material.dart';
import 'package:unguided/helper/db_helper.dart';
import 'package:unguided/view/add_mahasiswa_view.dart';

class MainView extends StatefulWidget {
  @override
  _MainViewState createState() => _MainViewState();
}

class _MainViewState extends State<MainView> {
  final dbHelper = DatabaseHelper();
  List<Map<String, dynamic>> mahasiswaList = [];

  void refreshData() async {
    final data = await dbHelper.queryAllRows();
    setState(() {
      mahasiswaList = data;
    });
  }

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'Biodata Mahasiswa',
          style: TextStyle(
              fontWeight: FontWeight.bold,
              color: Color.fromARGB(255, 255, 255, 255)),
        ),
        centerTitle: true,
        backgroundColor: Colors.brown,
      ),
      body: mahasiswaList.isEmpty
          ? const Center(
              child: Text(
                'Belum ada data mahasiswa.',
                style: TextStyle(fontSize: 16, color: Colors.grey),
              ),
            )
          : ListView.builder(
              itemCount: mahasiswaList.length,
              padding: const EdgeInsets.symmetric(vertical: 10),
              itemBuilder: (context, index) {
                final item = mahasiswaList[index];
                return Card(
                  margin:
                      const EdgeInsets.symmetric(vertical: 8, horizontal: 16),
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(12),
                  ),
                  elevation: 3,
                  child: ListTile(
                    contentPadding: const EdgeInsets.all(16),
                    title: Text(
                      item['nama'],
                      style: const TextStyle(
                          fontWeight: FontWeight.bold, fontSize: 18),
                    ),
                    subtitle: Column(
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
                        const SizedBox(height: 8),
                        Text('NIM: ${item['nim']}'),
                        Text('Alamat: ${item['alamat']}'),
                        Text('Hobi: ${item['hobi']}'),
                      ],
                    ),
                  ),
                );
              },
            ),
      floatingActionButton: FloatingActionButton(
        backgroundColor: Colors.brown,
        child: const Icon(Icons.add),
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => AddMahasiswaView(refreshData: refreshData),
            ),
          );
        },
      ),
    );
  }
}

```

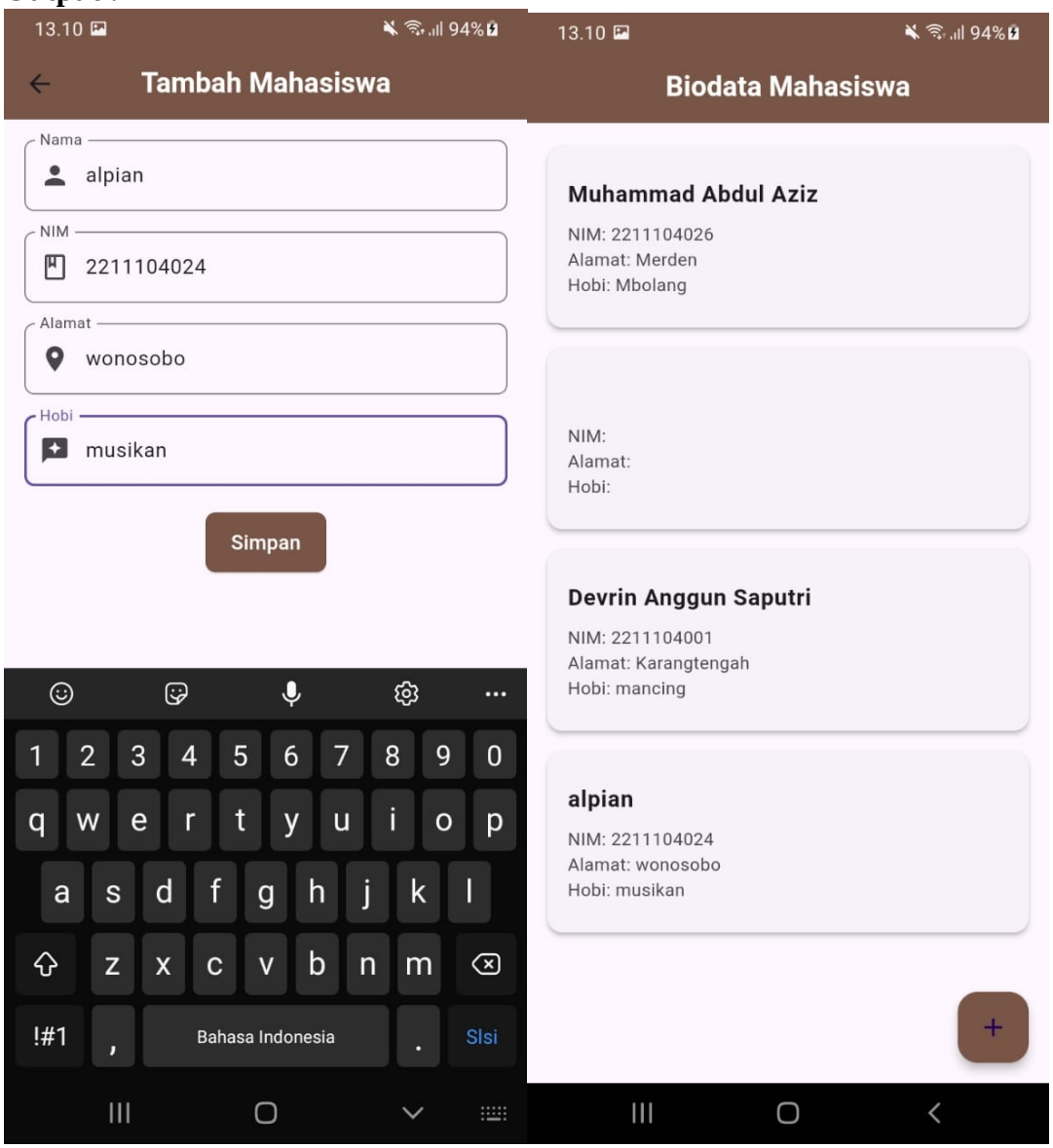
File model/mahasiswa_model.dart :

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

  Mahasiswa(
      {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,
    };
  }
}

```

File main.dart

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: MainView(),
    );
  }
}

```

Output :



Penjelasan :

Aplikasi ini adalah implementasi Flutter untuk pengelolaan data mahasiswa menggunakan SQLite dengan fokus pada operasi Create dan Read. Kelas `DatabaseHelper` bertanggung jawab mengelola database, termasuk membuat tabel `mahasiswa` dengan kolom `id`, `nama`, `nim`, `alamat`, dan `hobi`. Data mahasiswa dapat ditambahkan menggunakan halaman `AddMahasiswaView`, di mana pengguna mengisi form dengan informasi mahasiswa, yang kemudian disimpan ke database melalui metode `insert`. Data yang telah tersimpan ditampilkan dalam halaman `MainView` menggunakan ListView dengan widget Card. Antarmuka ini memberikan pengalaman sederhana untuk menambahkan data mahasiswa baru dan melihat daftar mahasiswa yang tersimpan di database.