LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK

MODUL X
DATA STORAGE

Disusun Oleh :

Satria Ariq Adelard

Dompas/2211104033SE 06 2

Asisten Praktikum :

Muhammad Faza Zulian Gesit Al Barru

Aisyah Hasna Aulia

Dosen Pengampu :

Yudha Islami Sulistya

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAKFAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO2024

1. GUIDED

a. Data Storage

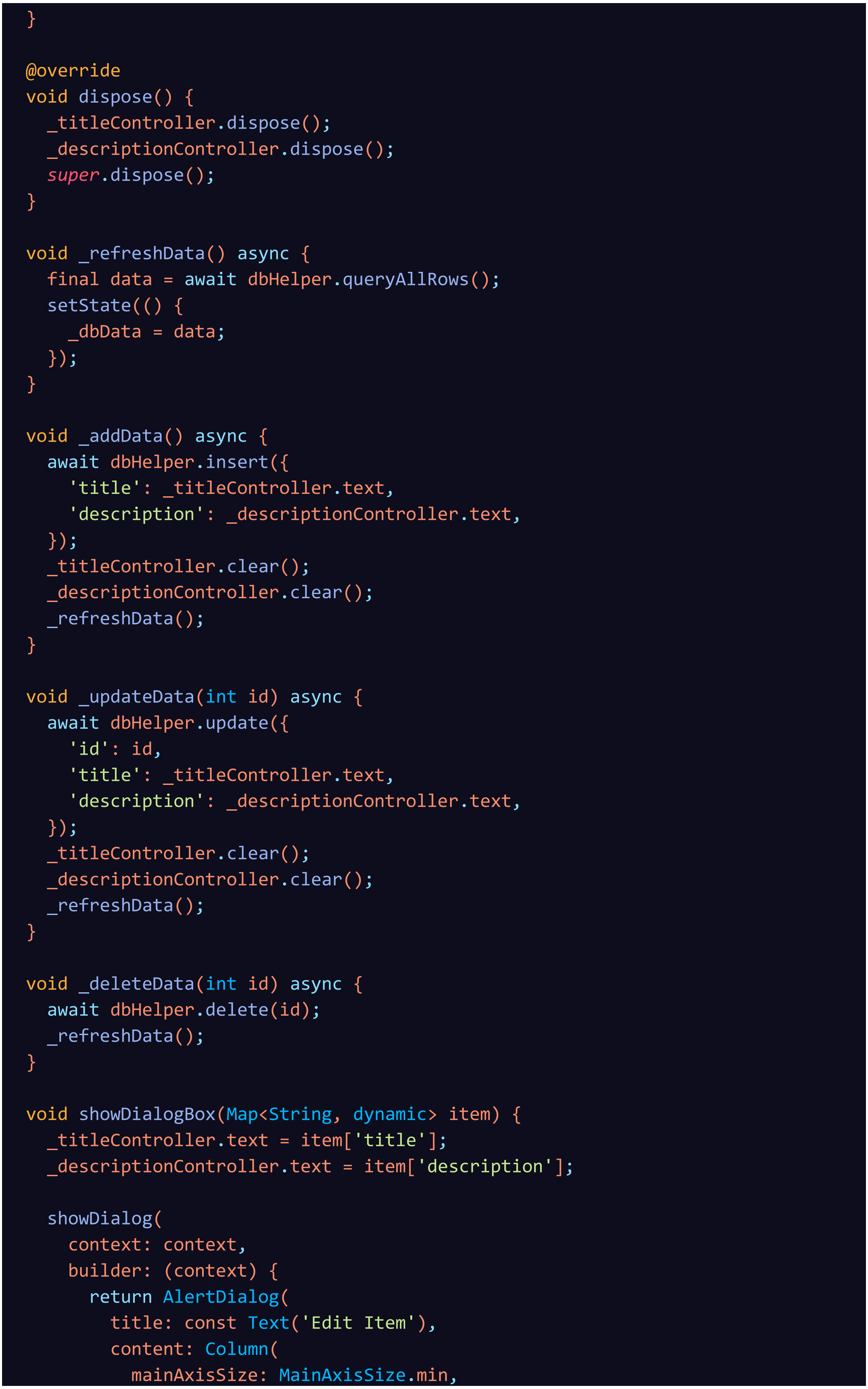
Source Code

- main.dart

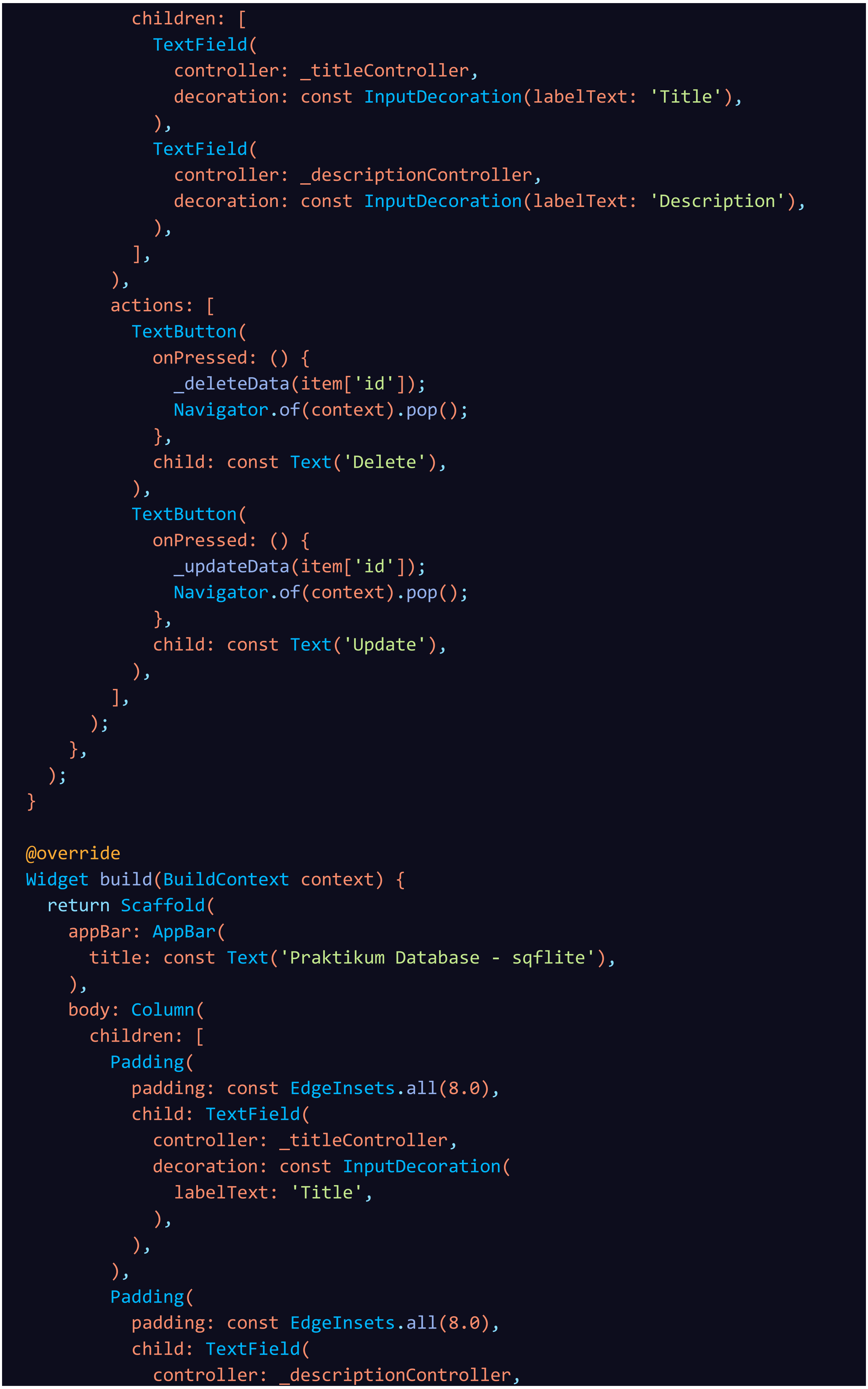
```
import 'package:flutter/material.dart';
import 'package:pertemuan_10/view/my_db_view.dart';

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

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

  @override
Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Database Storage (Bagian I)',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      debugShowCheckedModeBanner: false,
      home: const MyDatabaseView(),
    );
  }
}
```

- my_db_view.dart

```
import 'package:flutter/material.dart';
import 'package:pertemuan_10/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();
```



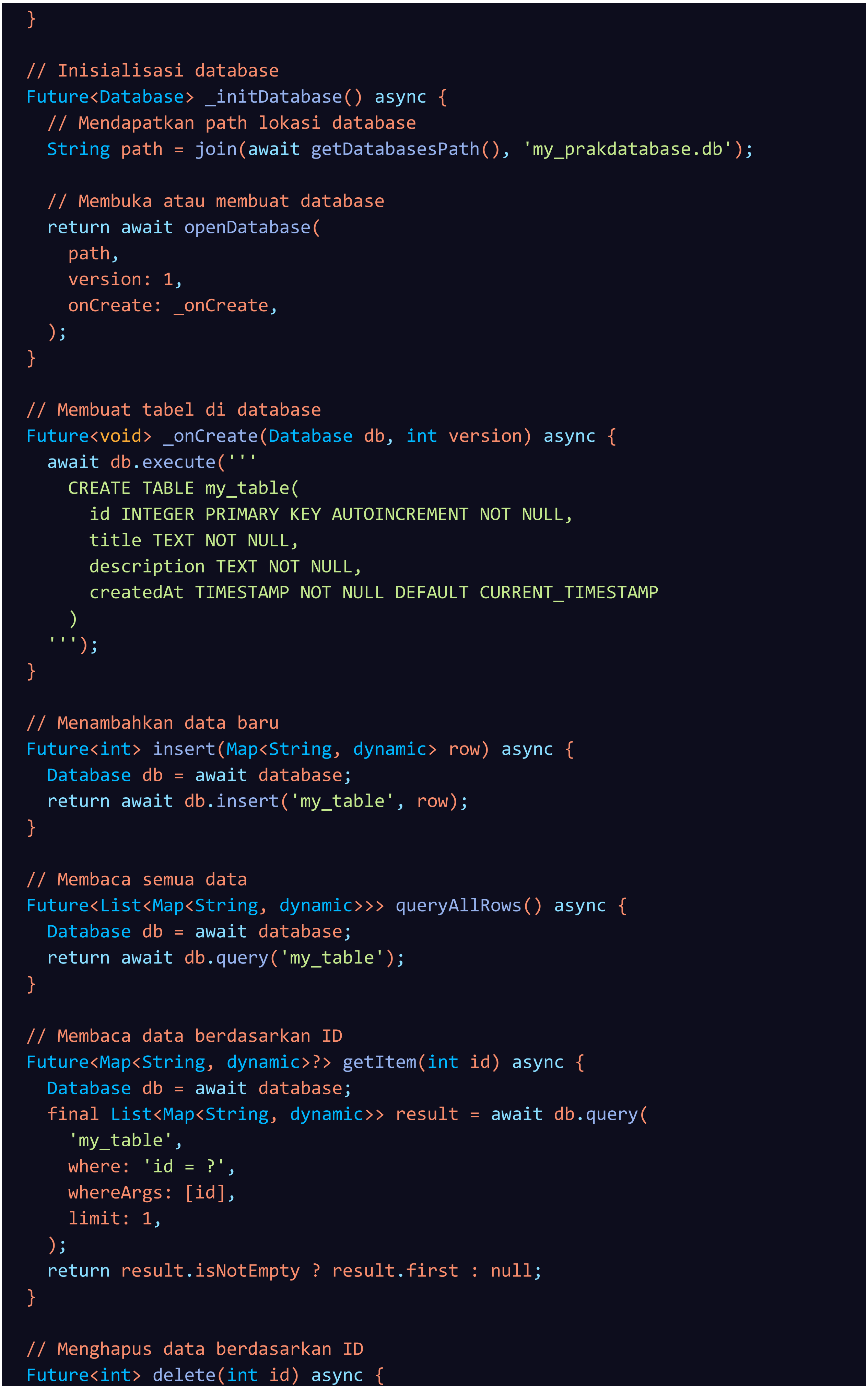
```
  }

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

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

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

  void showDialogBox(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,
```



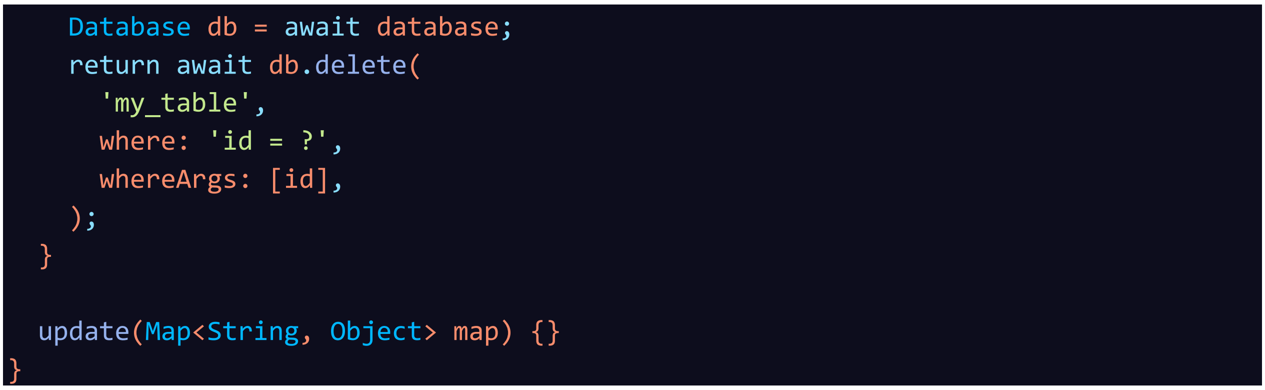
```
          children: [
            TextField(
              controller: _titleController,
              decoration: const InputDecoration(labelText: 'Title'),
            ),
            TextField(
              controller: _descriptionController,
              decoration: const InputDecoration(labelText: 'Description'),
            ),
          ],
        ),
        actions: [
          TextButton(
            onPressed: () {
              _deleteData(item['id']);
              Navigator.of(context).pop();
            },
            child: const Text('Delete'),
          ),
          TextButton(
            onPressed: () {
              _updateData(item['id']);
              Navigator.of(context).pop();
            },
            child: const Text('Update'),
          ),
        ],
      );
    },
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Praktikum Database - sqflite'),
    ),
    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) {
                return ListTile(
                  title: Text(_dbData[index]['title']),
                  subtitle: Text(_dbData[index]['description']),
                  trailing: IconButton(
                    icon: const Icon(Icons.edit),
                    onPressed: () => showDialogBox(_dbData[index]),
                  ),
                );
              },
            ),
          ),
        ],
      ),
    );
  }
}
```

- db_helper.dart

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

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

  // Constructor factory untuk mengembalikan instance
  factory DatabaseHelper() {
    return _instance;
  }

  // Constructor privat untuk singleton
  DatabaseHelper._internal();

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

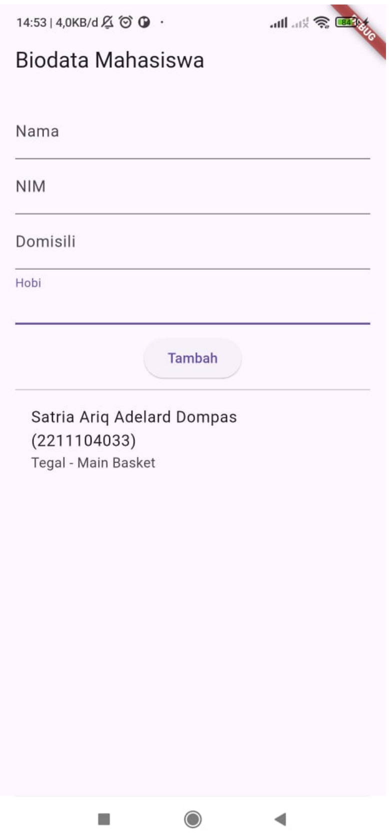


```
  }

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

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

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

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

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

  // Membaca data berdasarkan ID
  Future<Map<String, dynamic>?> getItem(int id) async {
    Database db = await database;
    final List<Map<String, dynamic>> result = await db.query(
      'my_table',
      where: 'id = ?',
      whereArgs: [id],
      limit: 1,
    );
    return result.isNotEmpty ? result.first : null;
  }

  // Menghapus data berdasarkan ID
  Future<int> delete(int id) async {
```



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

  update(Map<String, Object> map) {}
}
```

Output



Biodata Mahasiswa

Nama

NIM

Domisili

Hobi

Satria Ariq Adelard Dompas (2211104033)
Tegal - Main Basket

Deskripsi

Aplikasi Flutter ini adalah program sederhana yang pakai SQLite untuk menyimpan dan mengelola data langsung di perangkat. File main.dart jadi titik awal aplikasi, yang akan menampilkan halaman utama lewat widget MyDatabaseView (diatur dalam my_db_view.dart). Di halaman ini, pengguna bisa masukin, edit, atau hapus data berupa **title** dan **description**.

Semua fungsi CRUD (Create, Read, Update, Delete) dikelola pakai helper di file db_helper.dart. Di situ, SQLite dipakai buat bikin tabel database bernama my_table, yang punya kolom id, title, description, dan createdAt. Aplikasi ini pakai TextField buat input data, sementara ListView dipakai buat nampilin daftar data yang sudah disimpan. Ada juga tombol buat edit atau hapus data langsung dari daftar. Desainnya simpel banget karena pakai Material Design, dan semua pengelolaan state diatur pakai setState. Jadi, aplikasi ini cukup responsif dan gampang digunakan.

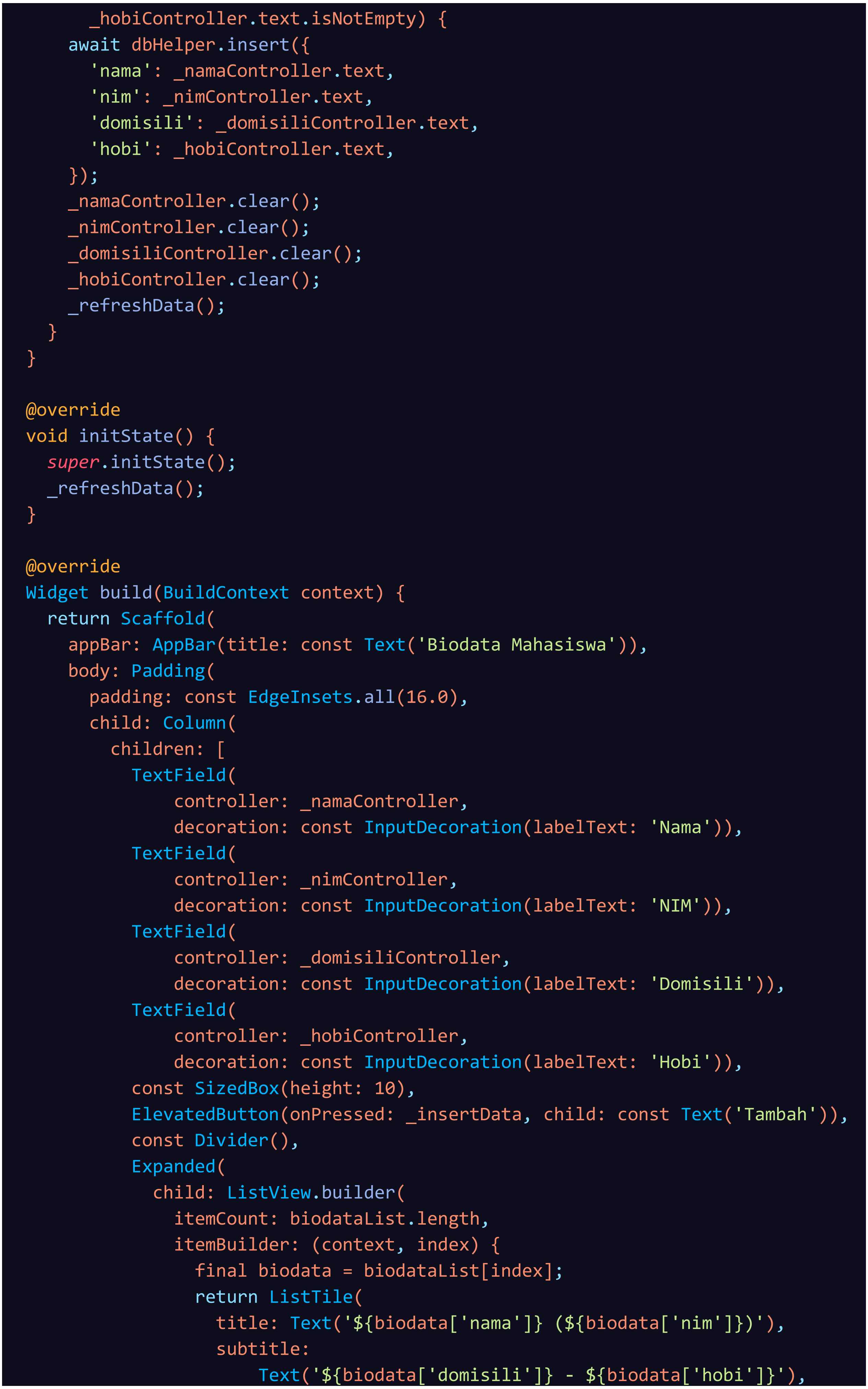
2. UNGUIDED

a. Soal 1

Source Code

- main.dart

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Biodata Mahasiswa',
      theme: ThemeData(primarySwatch: Colors.blue),
      home: const HomePage(),
    );
  }
}

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

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

class _HomePageState extends State<HomePage> {
  final dbHelper = DatabaseHelper.instance;
  List<Map<String, dynamic>> biodataList = [];

  final _namaController = TextEditingController();
  final _nimController = TextEditingController();
  final _domisiliController = TextEditingController();
  final _hobiController = TextEditingController();

  void _refreshData() async {
    final data = await dbHelper.queryAll();
    setState(() {
      biodataList = data;
    });
  }

  void _insertData() async {
    if (_namaController.text.isNotEmpty &&
        _nimController.text.isNotEmpty &&
        _domisiliController.text.isNotEmpty &&
```

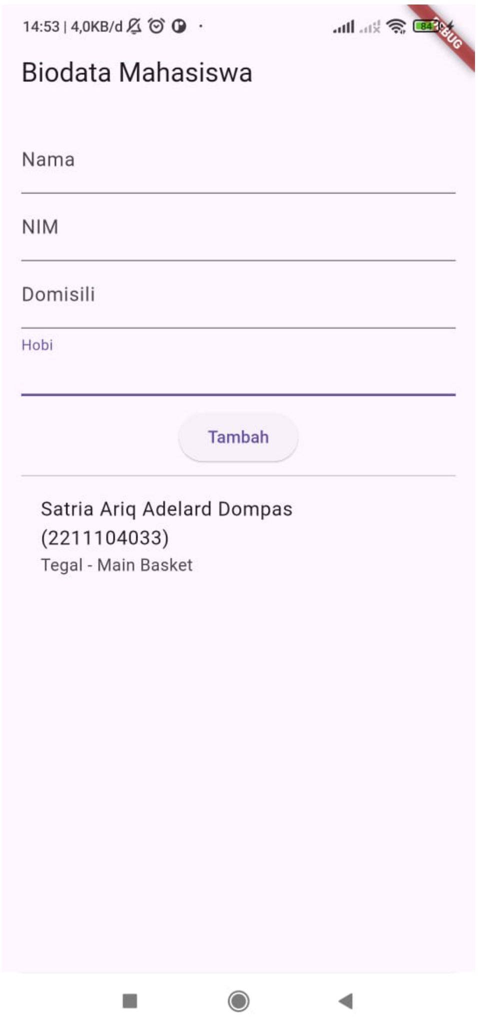


```
        _hobiController.text.isNotEmpty) {
      await dbHelper.insert({
        'nama': _namaController.text,
        'nim': _nimController.text,
        'domisili': _domisiliController.text,
        'hobi': _hobiController.text,
      });
      _namaController.clear();
      _nimController.clear();
      _domisiliController.clear();
      _hobiController.clear();
      _refreshData();
    }
  }

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Biodata Mahasiswa')),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          children: [
            TextField(
                controller: _namaController,
                decoration: const InputDecoration(labelText: 'Nama')),
            TextField(
                controller: _nimController,
                decoration: const InputDecoration(labelText: 'NIM')),
            TextField(
                controller: _domisiliController,
                decoration: const InputDecoration(labelText: 'Domisili')),
            TextField(
                controller: _hobiController,
                decoration: const InputDecoration(labelText: 'Hobi')),
            const SizedBox(height: 10),
            ElevatedButton(onPressed: _insertData, child: const Text('Tambah')),
            const Divider(),
            Expanded(
              child: ListView.builder(
                itemCount: biodataList.length,
                itemBuilder: (context, index) {
                  final biodata = biodataList[index];
                  return ListTile(
                    title: Text('${biodata['nama']} (${biodata['nim']})'),
                    subtitle:
                        Text('${biodata['domisili']} - ${biodata['hobi']}'),
                  );
                },
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

- database_helper.dart

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

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

  static Database? _database;

  DatabaseHelper._internal();

  factory DatabaseHelper() => instance;

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

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

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

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

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

Output



Satria Ariq Adelard Dompas (2211104033)
Tegal - Main Basket

Deskripsi

Aplikasi Flutter ini adalah program sederhana yang digunakan untuk menyimpan dan menampilkan data biodata mahasiswa dengan memanfaatkan SQLite sebagai database lokal. Di file main.dart, struktur utama aplikasinya diatur, termasuk halaman utama bernama HomePage. Di halaman ini, pengguna bisa memasukkan data seperti nama, NIM, domisili, dan hobi melalui form dengan TextField. Setelah data diisi, pengguna bisa menyimpannya ke database SQLite lewat fungsi insertData, dan data yang sudah tersimpan akan langsung ditampilkan dalam daftar menggunakan ListView.

Sedangkan file database_helper.dart bertugas untuk mengelola database. File ini mengatur pembuatan tabel biodata yang punya kolom id, nama, nim, domisili, dan hobi, serta menyediakan fungsi CRUD (khususnya untuk menyimpan dan membaca data). Dengan cara kerja ini, aplikasi bisa menyimpan data lokal, memperbarui daftar secara langsung, dan memberikan pengalaman pengguna yang praktis dan interaktif.