

Muhammad Ikhsan Al Hakim

2311104064

Unguided 10

Main.dart

```
import 'package:flutter/material.dart';
import 'package:sqflite/sqflite.dart';
import 'package:path/path.dart' as path;

void main() async {
    WidgetsFlutterBinding.ensureInitialized(); // 🔥 WAJIB
    runApp(const MyApp());
}

/* ===== APP ===== */
class MyApp extends StatelessWidget {
    const MyApp({super.key});

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

/* ===== DATABASE ===== */
class DatabaseHelper {
    static Database? _database;

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

    static Future<Database> _initDB() async {
        final dbPath = path.join(await getDatabasesPath(), 'mahasiswa.db');

        print('DB PATH: $dbPath');

        return await openDatabase(
            dbPath,
            version: 1,
            onCreate: (db, version) async {

```

```

print('DB CREATED');
await db.execute("""
    CREATE TABLE mahasiswa(
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        nama TEXT,
        nim TEXT,
        alamat TEXT,
        hobi TEXT
    )
""");
},
);
}
}

static Future<int> insertMahasiswa(Map<String, dynamic> data) async {
    final db = await database;
    final result = await db.insert('mahasiswa', data);
    print('INSERT RESULT: $result');
    return result;
}

static Future<List<Map<String, dynamic>>> getMahasiswa() async {
    final db = await database;
    return await db.query('mahasiswa', orderBy: 'id DESC');
}
}

/* ===== HOME PAGE ===== */
class HomePage extends StatefulWidget {
    const HomePage({super.key});

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

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

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

    Future<void> loadData() async {
        final data = await DatabaseHelper.getMahasiswa();
        if (!mounted) return;
        setState(() {
            mahasiswa = data;
        });
    }
}

```

```
}

@Override
Widget build(BuildContext context) {
return Scaffold(
  appBar: AppBar(title: const Text('SQLite Biodata Mahasiswa')),
  body: mahasiswa.isEmpty
    ? const Center(child: Text('Belum ada data'))
    : ListView.builder(
        itemCount: mahasiswa.length,
        itemBuilder: (context, index) {
          final mhs = mahasiswa[index];
          return ListTile(
            title: Text(mhs['nama']),
            subtitle: Text(
              'NIM: ${mhs['nim']} | '
              'Alamat: ${mhs['alamat']} | '
              'Hobi: ${mhs['hobi']}',
            ),
          );
        },
      ),
  floatingActionButton: FloatingActionButton(
    onPressed: () async {
      final result = await Navigator.push(
        context,
        MaterialPageRoute(builder: (_) => const AddPage()),
      );
      if (result == true) loadData();
    },
    child: const Icon(Icons.add),
  ),
)
}

/* ===== ADD PAGE ===== */
class AddPage extends StatefulWidget {
const AddPage({super.key});

@Override
State<AddPage> createState() => _AddPageState();
}

class _AddPageState extends State<AddPage> {
final nama = TextEditingController();
final nim = TextEditingController();
final alamat = TextEditingController();
final hobi = TextEditingController();
```

```
Future<void> save() async {
  if (nama.text.isEmpty ||
      nim.text.isEmpty ||
      alamat.text.isEmpty ||
      hobi.text.isEmpty) {
    ScaffoldMessenger.of(
      context,
    ).showSnackBar(const SnackBar(content: Text('Semua field wajib diisi')));
    return;
  }

  final result = await DatabaseHelper.insertMahasiswa({
    'nama': nama.text,
    'nim': nim.text,
    'alamat': alamat.text,
    'hobi': hobi.text,
  });

  if (!mounted) return;

  if (result > 0) {
    ScaffoldMessenger.of(
      context,
    ).showSnackBar(const SnackBar(content: Text('Data tersimpan')));
    await Future.delayed(const Duration(milliseconds: 400));
    Navigator.pop(context, true);
  }
}

@Override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: const Text('Tambah Data')),
    body: Padding(
      padding: const EdgeInsets.all(16),
      child: Column(
        children: [
          TextField(
            controller: nama,
            decoration: const InputDecoration(labelText: 'Nama'),
          ),
          TextField(
            controller: nim,
            decoration: const InputDecoration(labelText: 'NIM'),
          ),
          TextField(
            controller: alamat,
            decoration: const InputDecoration(labelText: 'Alamat'),
          ),
          TextField(

```

```
        controller: hobi,
        decoration: const InputDecoration(labelText: 'Hobi'),
    ),
    const SizedBox(height: 20),
    ElevatedButton(onPressed: save, child: const Text('Simpan')),
],
),
),
);
}
}
```

Kode ini merupakan aplikasi Flutter sederhana yang memanfaatkan **SQLite (sqflite)** untuk menyimpan dan menampilkan data biodata mahasiswa secara lokal. Aplikasi diawali dengan inisialisasi Flutter menggunakan `WidgetsFlutterBinding.ensureInitialized()` agar proses database dapat berjalan dengan benar. Kelas `DatabaseHelper` berfungsi sebagai pengelola database yang mencakup pembuatan database dan tabel mahasiswa, penyimpanan data (Create), serta pengambilan data (Read). Halaman utama (`HomePage`) menampilkan daftar biodata mahasiswa yang tersimpan dalam bentuk list, sementara halaman `AddPage` menyediakan form input untuk menambahkan data mahasiswa berupa nama, NIM, alamat, dan hobi. Data yang dimasukkan akan divalidasi terlebih dahulu, kemudian disimpan ke dalam database SQLite dan langsung ditampilkan kembali pada halaman utama, sehingga aplikasi ini memenuhi fungsi penyimpanan dan penampilan data secara sederhana dan efektif.

Hasil Run:

