



## UNGUIDED MODUL 10



Disusun Oleh :  
**Nama : Ganes Gemi Putra**  
**Kelas : SE-07-02**  
**NIM : (2311104075)**

**Dosen : YUDHA ISLAMI SULISTYA**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY**  
**PURWOKERTO**  
**2025**

## Pendahuluan

Aplikasi ini adalah sebuah proyek Flutter yang dikembangkan untuk mengelola biodata mahasiswa menggunakan database SQLite. Aplikasi ini memungkinkan pengguna untuk melakukan operasi CRUD (Create, Read, Update, Delete) pada data mahasiswa, termasuk nama, NIM, alamat, dan hobi. Aplikasi ini cocok digunakan sebagai contoh implementasi penyimpanan data lokal dalam aplikasi mobile.

## Teknologi yang Digunakan

- **Flutter:** Framework untuk pengembangan aplikasi mobile cross-platform (Android, iOS, Web).
- **SQLite:** Database relasional yang digunakan untuk penyimpanan data lokal.
- **sqflite:** Plugin Flutter untuk mengakses SQLite.
- **sqlite\_common\_ffi:** Untuk dukungan SQLite di platform desktop dan web.
- **path:** Untuk mengelola path file database.

## Struktur Proyek

Proyek ini terdiri dari beberapa file utama di direktori lib/:

- main.dart: File utama aplikasi yang menjalankan widget utama dan halaman home.
- database\_helper.dart: Kelas helper untuk mengelola koneksi dan operasi database.
- form\_tambah.dart: Form untuk menambah data mahasiswa baru.
- form\_edit.dart: Form untuk mengedit data mahasiswa yang sudah ada.
- pubspec.yaml: File konfigurasi proyek yang mencantumkan dependensi dan metadata aplikasi.

## Komponen Utama

### 1. DatabaseHelper (database\_helper.dart)

Kelas ini menggunakan pola Singleton untuk memastikan hanya ada satu instance database. Fungsi utama:

- initDB(): Membuat database dan tabel "mahasiswa" dengan kolom id (primary key auto-increment), nama, nim, alamat, hobi.
- create(): Menyisipkan data baru ke tabel.
- read(): Mengambil semua data dari tabel.
- update(): Memperbarui data berdasarkan ID.
- delete(): Menghapus data berdasarkan ID. Database disimpan di path yang sesuai dengan platform (web atau mobile).

### 2. HomePage (main.dart)

Halaman utama aplikasi yang menampilkan daftar mahasiswa dalam bentuk ListView. Fitur:

- Menampilkan data mahasiswa dengan desain kartu yang menarik, termasuk avatar dengan inisial nama.
- Tombol edit dan hapus untuk setiap item.
- Floating Action Button untuk menambah data baru.
- Menggunakan FutureBuilder untuk menangani loading dan data kosong.

### 3. FormTambah (form\_tambah.dart)

Form untuk menambah mahasiswa baru dengan field:

- Nama
- NIM
- Alamat
- Hobi Form ini menggunakan Form dan TextFormField dengan validasi agar semua field wajib diisi. Setelah menyimpan, kembali ke halaman utama dan refresh data.

### 4. FormEdit (form\_edit.dart)

Form untuk mengedit data mahasiswa yang sudah ada. Field sama dengan FormTambah, tetapi sudah terisi dengan data lama. Tombol simpan di AppBar untuk menyimpan perubahan.

### Cara Kerja Aplikasi

1. **Inisialisasi:** Saat aplikasi dijalankan, DatabaseHelper menginisialisasi database dan membuat tabel jika belum ada.
  2. **Tampilan Utama:** HomePage memuat data dari database dan menampilkannya dalam list. Jika belum ada data, menampilkan pesan "Belum ada data mahasiswa".
  3. **Menambah Data:** Klik FAB, isi form, simpan ke database, refresh list.
  4. **Mengedit Data:** Klik ikon edit pada item, ubah data di form, simpan perubahan.
- Menghapus Data:** Klik ikon hapus, konfirmasi penghapusan, refresh list. Aplikasi mendukung platform web dan mobile, dengan penyesuaian path database.

### SOURCECODE/SS OUTPUT :

main.dart :

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

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

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

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

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

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

class _HomePageState extends State<HomePage> {
  late Future<List<Map<String, dynamic>>> datamahasiswa;

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

  void refreshData() {
    setState(() {
      datamahasiswa = DatabaseHelper().getalldata();
    });
  }

  void tambahData(Map<String, dynamic> data) {
    setState(() {
      datamahasiswa.add(data);
    });
  }

  void editData(Map<String, dynamic> data, int index) {
    setState(() {
      datamahasiswa[index] = data;
    });
  }

  void hapusData(int index) {
    setState(() {
      datamahasiswa.removeAt(index);
    });
  }

  void konfirmasiHapus(int index) {
    showDialog(
      context: context,
      builder: (context) {
        return AlertDialog(
          title: Text("Konfirmasi"),
          content: Text("Anda yakin ingin menghapus data ini?"),
          actions: [
            TextButton(
              onPressed: () {
                Navigator.of(context).pop();
              },
              child: Text("Batal"),
            ),
            TextButton(
              onPressed: () {
                hapusData(index);
                Navigator.of(context).pop();
              },
              child: Text("Hapus"),
            ),
          ],
        );
      },
    );
  }

  void bukaFormTambah() {
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => FormTambah()),
    );
  }

  void bukaFormEdit(Map<String, dynamic> data) {
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => FormEdit(data)),
    );
  }

  void bukaFormHapus(Map<String, dynamic> data) {
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => FormHapus(data)),
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Daftar Mahasiswa"),
        centerTitle: true,
      ),
      body: Container(
        padding: EdgeInsets.all(16),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.start,
          children: [
            ElevatedButton(
              onPressed: bukaFormTambah,
              child: Text("Tambah Data"),
            ),
            SizedBox(
              height: 16,
            ),
            FutureBuilder(
              future: datamahasiswa,
              builder: (context, snapshot) {
                if (snapshot.hasData) {
                  return ListView.builder(
                    itemCount: snapshot.data!.length,
                    itemBuilder: (context, index) {
                      Map<String, dynamic> data = snapshot.data![index];
                      return ListTile(
                        title: Text(data['nim']),
                        subtitle: Text(data['nama']),
                        trailing: Row(
                          mainAxisSize: MainAxisSize.min,
                          children: [
                            IconButton(
                              icon: Icon(Icons.edit),
                              onPressed: () {
                                bukaFormEdit(data);
                              },
                            ),
                            IconButton(
                              icon: Icon(Icons.delete),
                              onPressed: () {
                                konfirmasiHapus(index);
                              },
                            ),
                          ],
                        ),
                      );
                    },
                  );
                } else {
                  return Center(
                    child: CircularProgressIndicator(),
                  );
                }
              },
            ),
          ],
        ),
      ),
    );
  }
}
```

```
void refreshData() {
    datamahasiswa = DatabaseHelper().read();
    setState(() {});
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: const Text("SQLite Biodata Mahasiswa"),
            backgroundColor: Colors.amber[700],
        ),
        body: FutureBuilder(
            future: datamahasiswa,
            builder: (context, snapshot) {
                if (snapshot.connectionState == ConnectionState.waiting) {
                    return const Center(child: CircularProgressIndicator());
                }

                if (!snapshot.hasData || snapshot.data!.isEmpty) {
                    return const Center(child: Text("Belum ada data mahasiswa"));
                }

                final data = snapshot.data!;

                return ListView.builder(
                    padding: const EdgeInsets.all(16),
                    itemCount: data.length,
                    itemBuilder: (context, index) {
                        final m = data[index];

                        return InkWell(
                            borderRadius: BorderRadius.circular(20),
                            onTap: () {},
                            child: Container(
                                margin: const EdgeInsets.only(bottom: 16),
                                decoration: BoxDecoration(
                                    borderRadius: BorderRadius.circular(20),
                                    gradient: LinearGradient(
                                        colors: [
                                            Colors.white,
                                            Colors.grey.shade50,
                                        ],
                                        begin: Alignment.topLeft,
                                        end: Alignment.bottomRight,
                                    ),
                                    boxShadow: [
                                        BoxShadow(
                                            color: Colors.black.withOpacity(0.07),
                                            blurRadius: 12,
                                        )
                                    ],
                                ),
                            ),
                        );
                    }
                );
            }
        );
}
```

```
        offset: const Offset(0, 5),
    ),
],
),
child: Padding(
    padding: const EdgeInsets.all(18.0),
child: Row(
    mainAxisAlignment: MainAxisAlignment.start,
children: [
    // Avatar Aesthetic
Container(
    width: 60,
    height: 60,
decoration: BoxDecoration(
    shape: BoxShape.circle,
gradient: LinearGradient(
    colors: [
        Colors.amber.shade700,
        Colors.orange.shade400,
    ],
    begin: Alignment.topLeft,
    end: Alignment.bottomRight,
),
boxShadow: [
    BoxShadow(
        color: Colors.amber.withOpacity(0.4),
        blurRadius: 10,
        offset: const Offset(0, 4),
    ),
],
),
),
child: Center(
    child: Text(
        m['nama'][0].toUpperCase(),
style: const TextStyle(
        color: Colors.white,
        fontSize: 26,
        fontWeight: FontWeight.bold,
),
),
),
),
),
),

const SizedBox(width: 18),

// Text Section
Expanded(
    child: Column(
        mainAxisAlignment: MainAxisAlignment.start,
children: [
    // NAMA
```

```

Text(
  m['nama'],
  style: TextStyle(
    fontSize: 22,
    fontWeight: FontWeight.bold,
    color: Colors.grey.shade900,
  ),
),

const SizedBox(height: 8),
Divider(color: Colors.grey.shade300, thickness: 1),

const SizedBox(height: 10),

// NIM
Row(
  children: [
    Icon(Icons.badge_rounded,
      size: 22, color: Colors.amber.shade700),
    const SizedBox(width: 10),
    Text("NIM: ${m['nim']}",
      style: TextStyle(
        fontSize: 16, color: Colors.grey.shade800)),
  ],
),

const SizedBox(height: 8),

// ALAMAT
Row(
  children: [
    Icon(Icons.location_on_rounded,
      size: 22, color: Colors.red.shade400),
    const SizedBox(width: 10),
    Text("Alamat: ${m['alamat']}",
      style: TextStyle(
        fontSize: 16, color: Colors.grey.shade800)),
  ],
),

const SizedBox(height: 8),

// HOBI
Row(
  children: [
    Icon(Icons.favorite_rounded,
      size: 22, color: Colors.pink.shade400),
    const SizedBox(width: 10),
    Text("Hobi: ${m['hobi']}",
      style: TextStyle(
        fontSize: 16, color: Colors.grey.shade800)),
  ],
)

```



```
        ],
      ),
    const SizedBox(height: 10),
    Row(
      mainAxisAlignment: MainAxisAlignment.end,
      children: [
        IconButton(
          icon: const Icon(Icons.edit, color: Colors.blue),
          onPressed: () {
            // Navigate to edit form
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (_) => FormEdit(mahasiswa: m),
              ),
            ).then((_) => reloadData());
          },
        ),
        IconButton(
          icon: const Icon(Icons.delete, color: Colors.red),
          onPressed: () async {
            await DatabaseHelper().delete(m['id']);
            reloadData();
          },
        ),
      ],
    ),
  ],
),
floatingActionButton: FloatingActionButton(
  backgroundColor: Colors.amber[700],
  onPressed: () async {
    await Navigator.push(
      context,
      MaterialPageRoute(builder: (_) => const FormTambah()),
    );
    reloadData();
  },
  child: const Icon(Icons.add),
),
```

```
    );  
}  
}
```

**database\_helper.dart :**

```
import 'package:path/path.dart';  
import 'package:flutter/foundation.dart' show kIsWeb;  
import 'package:sqflite_common_ffi/sqflite_ffi.dart';  
import 'package:sqflite_common_ffi_web/sqflite_ffi_web.dart';  
  
class DatabaseHelper {  
  static final DatabaseHelper _instance = DatabaseHelper._internal();  
  static Database? _database;  
  
  DatabaseHelper._internal() {  
    if (kIsWeb) {  
      sqfliteFfiInit();  
      databaseFactory = databaseFactoryFFiWeb;  
    } else {  
      sqfliteFfiInit();  
    }  
  }  
  
  factory DatabaseHelper() {  
    return _instance;  
  }  
  
  Future<Database> get database async {  
    if (_database != null) return _database!;  
    _database = await initDB();  
    return _database!;  
  }  
  
  Future<Database> initDB() async {  
    String path;  
    if (kIsWeb) {  
      path = 'mahasiswa.db';  
    } else {  
      path = join(await getDatabasesPath(), "mahasiswa.db");  
    }  
  
    return await openDatabase(  
      path,  
      version: 1,  
      onCreate: (db, version) async {  
        await db.execute(''  
          CREATE TABLE mahasiswa (  
            id INTEGER PRIMARY KEY AUTOINCREMENT,  
            nama TEXT,  
        )  
      );  
    }  
  }  
}
```

```

        nim TEXT,
        alamat TEXT,
        hobi TEXT
    )
    """);
},
);
}

Future<void> create(Map<String, dynamic> data) async {
    final db = await database;
    await db.insert("mahasiswa", data);
}

Future<List<Map<String, dynamic>>> read() async {
    final db = await database;
    return await db.query("mahasiswa");
}

Future<void> update(int id, Map<String, dynamic> data) async {
    final db = await database;
    await db.update("mahasiswa", data, where: "id = ?", whereArgs: [id]);
}

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

```

#### form\_edit.dart :

```

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

class FormEdit extends StatefulWidget {
final Map<String, dynamic> mahasiswa;

const FormEdit({super.key, required this.mahasiswa});

@Override
State<FormEdit> createState() => _FormEditState();
}

class _FormEditState extends State<FormEdit> {
late TextEditingController namaController;
late TextEditingController nimController;
late TextEditingController alamatController;
late TextEditingController hobiController;
}

```

```
@override
void initState() {
    super.initState();
    namaController = TextEditingController(text: widget.mahasiswa['nama']);
    nimController = TextEditingController(text: widget.mahasiswa['nim']);
    alamatController = TextEditingController(text: widget.mahasiswa['alamat']);
    hobiController = TextEditingController(text: widget.mahasiswa['hobi']);
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: const Text("Edit Biodata Mahasiswa"),
            backgroundColor: Colors.amber[700],
            actions: [
                IconButton(
                    icon: const Icon(Icons.save),
                    onPressed: () async {
                        if (namaController.text.isEmpty ||
                            nimController.text.isEmpty ||
                            alamatController.text.isEmpty ||
                            hobiController.text.isEmpty) {
                            ScaffoldMessenger.of(context).showSnackBar(
                                const SnackBar(content: Text("Semua field harus diisi!")));
                        }
                        return;
                    }
                ),
                await DatabaseHelper().update(widget.mahasiswa['id'], {
                    "nama": namaController.text,
                    "nim": nimController.text,
                    "alamat": alamatController.text,
                    "hobi": hobiController.text,
                });

                if (mounted) {
                    ScaffoldMessenger.of(context).showSnackBar(
                        const SnackBar(content: Text("Data berhasil diperbarui!")),
                    );
                    Navigator.pop(context);
                }
            ],
        ),
        body: SingleChildScrollView(
            padding: const EdgeInsets.all(16.0),
            child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [

```



```
const Text(  
    "Edit Biodata Mahasiswa",  
    style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),  
,  
const SizedBox(height: 20),  
TextField(  
    controller: namaController,  
    decoration: InputDecoration(  
        labelText: "Nama",  
        prefixIcon: const Icon(Icons.person),  
        border: OutlineInputBorder(  
            borderRadius: BorderRadius.circular(10),  
    ),  
,  
),  
const SizedBox(height: 16),  
TextField(  
    controller: nimController,  
    decoration: InputDecoration(  
        labelText: "NIM",  
        prefixIcon: const Icon(Icons.badge),  
        border: OutlineInputBorder(  
            borderRadius: BorderRadius.circular(10),  
    ),  
,  
),  
const SizedBox(height: 16),  
TextField(  
    controller: alamatController,  
    decoration: InputDecoration(  
        labelText: "Alamat",  
        prefixIcon: const Icon(Icons.home),  
        border: OutlineInputBorder(  
            borderRadius: BorderRadius.circular(10),  
    ),  
,  
),  
const SizedBox(height: 16),  
TextField(  
    controller: hobiController,  
    decoration: InputDecoration(  
        labelText: "Hobi",  
        prefixIcon: const Icon(Icons.sports_soccer),  
        border: OutlineInputBorder(  
            borderRadius: BorderRadius.circular(10),  
    ),  
,  
),  
],
```

```
});  
}  
}
```

**form\_tambah.dart :**

```
import 'package:flutter/material.dart';  
import 'database_helper.dart';  
  
class FormTambah extends StatefulWidget {  
  const FormTambah({super.key});  
  
  @override  
  State<FormTambah> createState() => _FormTambahState();  
}  
  
class _FormTambahState extends State<FormTambah> {  
  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: const Text("Tambah Biodata Mahasiswa"),  
        backgroundColor: Colors.amber[700],  
        actions: [  
          IconButton(  
            icon: const Icon(Icons.person_add),  
            onPressed: () {  
              // Optional: Add functionality if needed  
            },  
          ),  
        ],  
      ),  
      body: SingleChildScrollView(  
        padding: const EdgeInsets.all(16.0),  
        child: Column(  
          crossAxisAlignment: CrossAxisAlignment.start,  
          children: [  
            const Text(  
              "Masukkan Biodata Mahasiswa",  
              style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),  
            ),  
            const SizedBox(height: 20),  
            TextField(  
              controller: namaController,  
              decoration: InputDecoration(  
                
```

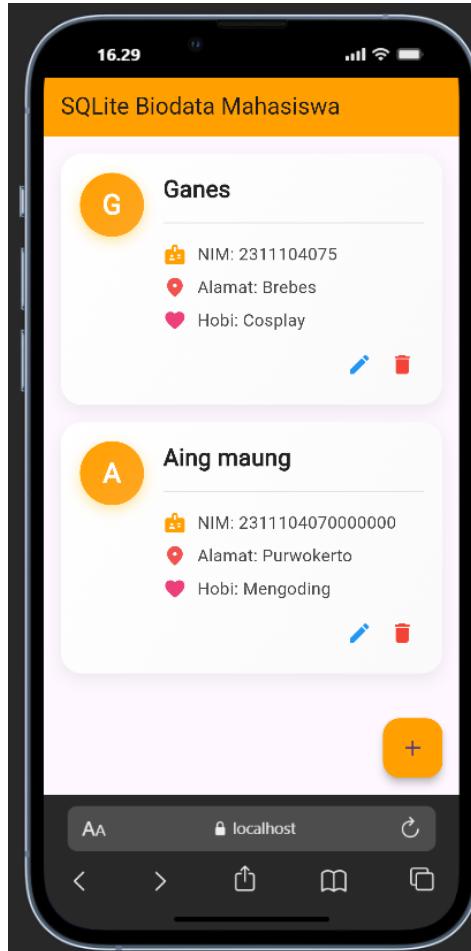
```
labelText: "Nama",
prefixIcon: const Icon(Icons.person),
border: OutlineInputBorder(
  borderRadius: BorderRadius.circular(10),
),
),
),
),
const SizedBox(height: 16),
TextField(
  controller: nimController,
  decoration: InputDecoration(
    labelText: "NIM",
    prefixIcon: const Icon(Icons.badge),
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(10),
    ),
  ),
),
),
const SizedBox(height: 16),
TextField(
  controller: alamatController,
  decoration: InputDecoration(
    labelText: "Alamat",
    prefixIcon: const Icon(Icons.home),
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(10),
    ),
  ),
),
),
),
const SizedBox(height: 16),
TextField(
  controller: hobiController,
  decoration: InputDecoration(
    labelText: "Hobi",
    prefixIcon: const Icon(Icons.sports_soccer),
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(10),
    ),
  ),
),
),
),
const SizedBox(height: 30),
Center(
  child: ElevatedButton.icon(
    onPressed: () async {
      if (namaController.text.isEmpty ||
          nimController.text.isEmpty ||
          alamatController.text.isEmpty ||
          hobiController.text.isEmpty) {
        ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(content: Text("Semua field harus diisi!")),
        );
      }
    },
  ),
);
```

```
        return;
    }
    await DatabaseHelper().create({
        'nama': namaController.text,
        'nim': nimController.text,
        'alamat': alamatController.text,
        'hobi': hobiController.text,
    });

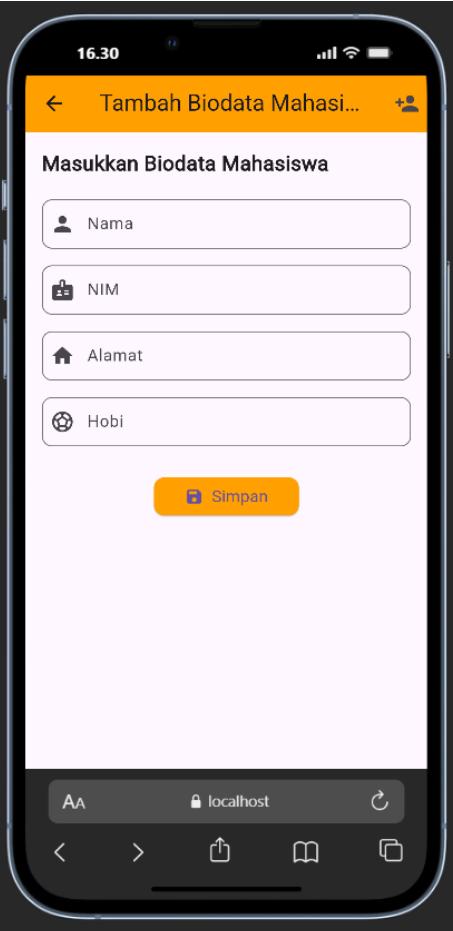
    if (mounted) {
        Navigator.pop(context);
    }
},
style: ElevatedButton.styleFrom(
    backgroundColor: Colors.amber[700],
    padding: const EdgeInsets.symmetric(horizontal: 30, vertical: 15),
    shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10),
    ),
),
icon: const Icon(Icons.save),
label: const Text("Simpan", style: TextStyle(fontSize: 16)),
),
),
],
),
),
);
}
}
```

**SCRENSHOOT :**

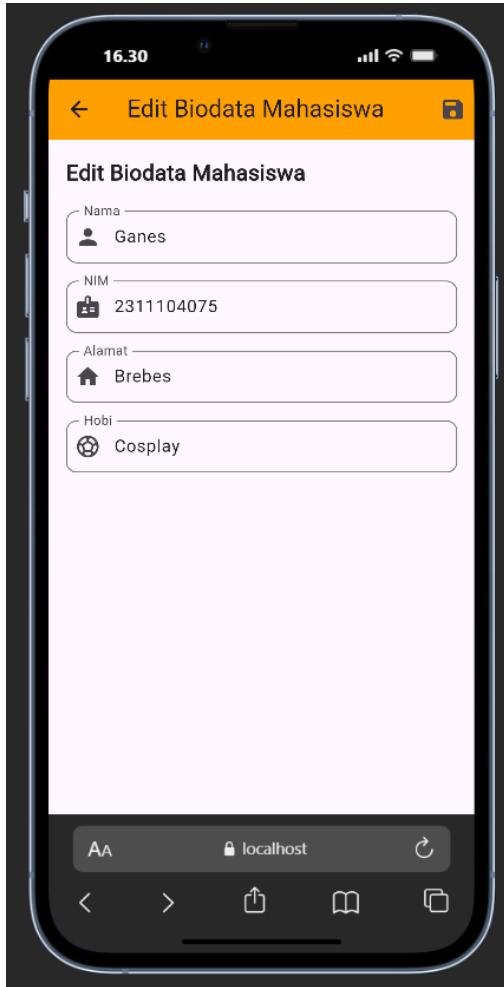
**SQLite Biodata Mahasiswa :**



**Tambah Biodata Mahasiswa :**



### EDIT BIODATA MAHASISWA :



### Kesimpulan

Aplikasi ini menunjukkan implementasi dasar penyimpanan data lokal menggunakan SQLite di Flutter. Dengan struktur yang sederhana dan fitur CRUD lengkap, aplikasi ini dapat dijadikan dasar untuk pengembangan aplikasi yang lebih kompleks. Kode yang digunakan mengikuti best practices Flutter, seperti penggunaan StatefulWidget untuk state management dan async/await untuk operasi database.

