

**TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK
MODUL X**

DATA STORAGE (BAGIAN I)



Disusun Oleh :

Nama: Arwin Nabiel Arroffif

NIM: 2211104057

Kelas: 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

DATA STORAGE (BAGIAN I)

1. Pengenalan SQLite

SQLite adalah database relasional yang merupakan penyimpanan data secara offline untuk sebuah mobile app (pada *local storage*, lebih tepatnya pada *cache memory* aplikasi). SQLite memiliki CRUD (*create, read, update dan delete*). Empat operasi tersebut penting dalam sebuah penyimpanan. Untuk struktur database pada SQLite, sama seperti SQL pada umumnya, variabel dan tipe data yang dimiliki tidak jauh berbeda dengan SQL. Untuk informasi terkait basic SQL ada pada link berikut.

2. SQL Helper Dasar

Dalam Flutter, SQL Helper biasanya merujuk pada penggunaan paket seperti *sqflite* untuk mengelola database SQLite. SQL Helper merupakan class untuk membuat beberapa method yang berkaitan dengan perubahan data. *sqflite* adalah plugin Flutter yang memungkinkan untuk melakukan operasi CRUD (Create, Read, Update, Delete) pada database SQLite.

a) Read

Di package *sqflite*, kita akan menggunakan `query()` untuk membaca data yang ada pada database. *Sqflite* di Flutter kita dapat membuat query menggunakan bermacam-macam perintah, seperti `where`, `groupBy`, `orderBy`, dan `having`. Selain itu, kita juga bisa membaca satu data atau banyak data sekaligus.

Untuk hasil praktikum ada di github bagian GUIDED

Unguided

Tugas Mandiri (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:

Form Input: Buat form input untuk menambahkan biodata mahasiswa, dengan kolom:

Nama

Nim

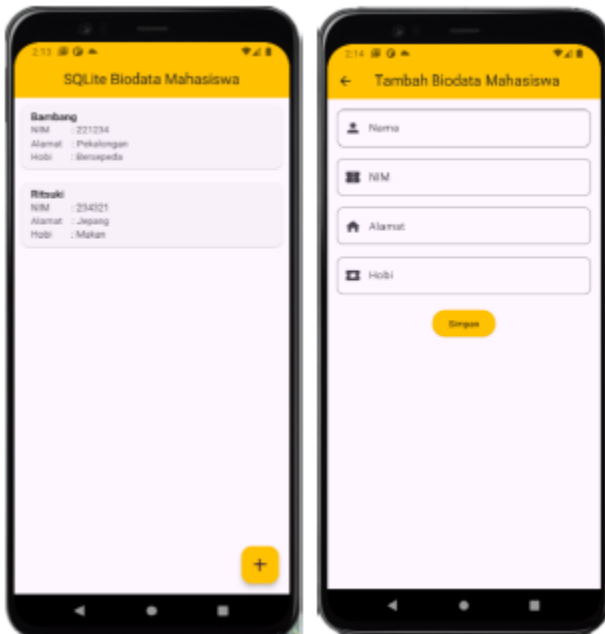
Alamat

Hobi

Tampilkan Daftar Mahasiswa: Setelah data berhasil ditambahkan, tampilkan daftar semua data mahasiswa yang sudah disimpan di halaman utama.

Implementasikan fitur Create (untuk menyimpan data mahasiswa) dan Read (untuk menampilkan daftar mahasiswa yang sudah disimpan).

Contoh output:



Note: Jangan lupa sertakan source code, screenshot output, dan deskripsi program. Kreativitas menjadi nilai tambah.

JAWAB

main.dart

```
import 'package:flutter/material.dart';
import 'package:laprak_10/view/home_page.dart';

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

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.grey, // Ubah tema
          menjadi abu-abu
        ),
        home: const HomePage(),
      );
    }
  }
}

```

home_page.dart

```

import 'package:flutter/material.dart';
import 'package:laprak_10/helper/db_helper.dart';
import 'package:laprak_10/view/add_edit_page.dart';

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

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

class _HomePageState extends State<HomePage> {
  final DatabaseHelper dbHelper = DatabaseHelper();
  List<Map<String, dynamic>> _dbData = [];

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

```

```

void _refreshData() async {
    final data = await dbHelper.queryAllRows();
    setState(() {
        _dbData = data;
    });
}

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

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: const Text('SQLite Biodata Mahasiswa'),
            backgroundColor: Colors.grey,
        ),
        body: ListView.builder(
            itemCount: _dbData.length,
            itemBuilder: (context, index) {
                final item = _dbData[index];
                return Card(
                    margin: const
EdgeInsets.symmetric(vertical: 8, horizontal: 16),
                    child: Padding(
                        padding: const EdgeInsets.all(16.0),
                        child: Column(
                            crossAxisAlignment:
CrossAxisAlignment.start,

```

```

        children: [
          Text(
            item['nama'],
            style: const TextStyle(
              fontSize: 18,
              fontWeight: FontWeight.bold,
            ),
          ),
          const SizedBox(height: 8),
          Text('NIM: ${item['nim']}'),
          Text('Alamat: ${item['alamat']}'),
          Text('Hobi: ${item['hobi']}'),
          const SizedBox(height: 8),
          Row(
            mainAxisAlignment:
MainAxisAlignment.end,
            children: [
              IconButton(
                icon: const Icon(Icons.edit,
color: Colors.blue),
                onPressed: () {
                  Navigator.of(context).push(
                    MaterialPageRoute(
                      builder: (context) =>
AddEditPage(
                        data: item,
                        onDataSaved:
_refreshData,
                      ),
                    ),
                  );
                },
              ),
            ],
          ),
        ],
      ),
    ],
  );
}

```

```

        ),
        IconButton(
            icon: const Icon(Icons.delete,
color: Colors.red),
            onPressed: () =>
_deleteData(item['id']),
        ),
    ],
),
],
),
),
);
},
),
floatingActionButton: FloatingActionButton(
    onPressed: () {
        Navigator.of(context).push(
            MaterialPageRoute(
                builder: (context) =>
AddEditPage(onDataSaved: _refreshData),
            ),
        );
    },
    backgroundColor: Colors.grey,
    child: const Icon(Icons.add),
),
);
}
}

```

add_edit_page.dart

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



```
import 'package:laprak_10/helper/db_helper.dart';

class AddEditPage extends StatefulWidget {
  final Map<String, dynamic>? data; // Data untuk
  edit, null jika tambah

  final VoidCallback onDataSaved;

  const AddEditPage({super.key, this.data, required
  this.onDataSaved});

  @override
  State<AddEditPage> createState() =>
  _AddEditPageState();
}

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

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

    if (widget.data != null) {
      _namaController.text = widget.data!['nama'];
      _nimController.text = widget.data!['nim'];
      _alamatController.text = widget.data!['alamat'];
```

```

        _hobiController.text = widget.data!['hobi'];
    }
}

void _saveData() async {
    if (_namaController.text.isNotEmpty &&
        _nimController.text.isNotEmpty &&
        _alamatController.text.isNotEmpty &&
        _hobiController.text.isNotEmpty) {
        final data = {
            'nama': _namaController.text,
            'nim': _nimController.text,
            'alamat': _alamatController.text,
            'hobi': _hobiController.text,
        };

        if (widget.data == null) {
            // Tambah data
            await dbHelper.insert(data);
        } else {
            // Edit data
            data['id'] = widget.data!['id'];
            await dbHelper.update(data);
        }

        widget.onDataSaved(); // Memperbarui halaman
utama
        Navigator.of(context).pop();
    } else {
        ScaffoldMessenger.of(context).showSnackBar(

```

```

        const SnackBar(content: Text('Semua field
harus diisi!')),

        );
    }
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text(widget.data == null ? 'Tambah
Data' : 'Edit Data'),
            backgroundColor: Colors.grey,
        ),
        body: Padding(
            padding: const EdgeInsets.all(16.0),
            child: Column(
                children: [
                    TextField(
                        controller: _namaController,
                        decoration: const InputDecoration(
                            labelText: 'Nama',
                            prefixIcon: Icon(Icons.person),
                            border: OutlineInputBorder(),
                        ),
                    ),
                    const SizedBox(height: 16),
                    TextField(
                        controller: _nimController,
                        decoration: const InputDecoration(
                            labelText: 'NIM',

```

```
        prefixIcon: Icon(Icons.numbers),
        border: OutlineInputBorder(),
    ),
),
const SizedBox(height: 16),
TextField(
    controller: _alamatController,
    decoration: const InputDecoration(
        labelText: 'Alamat',
        prefixIcon: Icon(Icons.home),
        border: OutlineInputBorder(),
    ),
),
const SizedBox(height: 16),
TextField(
    controller: _hobiController,
    decoration: const InputDecoration(
        labelText: 'Hobi',
        prefixIcon: Icon(Icons.favorite),
        border: OutlineInputBorder(),
    ),
),
const SizedBox(height: 16),
ElevatedButton(
    onPressed: _saveData,
    style: ElevatedButton.styleFrom(
        backgroundColor: Colors.grey,
        foregroundColor: Colors.white,
    ),
    child: const Text('Simpan'),
```

```

        ),
      ],
    ),
  ),
);
}
}

```

add_data_page.dart

```

import 'package:flutter/material.dart';
import 'package:laprak_10/helper/db_helper.dart';

class AddDataPage extends StatefulWidget {
  final VoidCallback onDataAdded;

  const AddDataPage({super.key, required
this.onDataAdded});

  @override
  State<AddDataPage> createState() =>
  _AddDataPageState();
}

class _AddDataPageState extends State<AddDataPage> {
  final DatabaseHelper dbHelper = DatabaseHelper();

  final TextEditingController _namaController =
  TextEditingController();

  final TextEditingController _nimController =
  TextEditingController();

  final TextEditingController _alamatController =
  TextEditingController();

  final TextEditingController _hobiController =
  TextEditingController();
}

```

```

void _addData() async {
  if (_namaController.text.isNotEmpty &&
      _nimController.text.isNotEmpty &&
      _alamatController.text.isNotEmpty &&
      _hobiController.text.isNotEmpty) {
    await dbHelper.insert({
      'nama': _namaController.text,
      'nim': _nimController.text,
      'alamat': _alamatController.text,
      'hobi': _hobiController.text,
    });
    widget.onDataAdded();
    Navigator.of(context).pop();
  } else {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('Semua field
harus diisi!')),
    );
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Tambah Biodata Mahasiswa'),
      backgroundColor: Colors.yellow,
    ),
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(

```

```
children: [
  TextField(
    controller: _namaController,
    decoration: const InputDecoration(
      labelText: 'Nama',
      prefixIcon: Icon(Icons.person),
      border: OutlineInputBorder(),
    ),
  ),
  const SizedBox(height: 16),
  TextField(
    controller: _nimController,
    decoration: const InputDecoration(
      labelText: 'NIM',
      prefixIcon: Icon(Icons.numbers),
      border: OutlineInputBorder(),
    ),
  ),
  const SizedBox(height: 16),
  TextField(
    controller: _alamatController,
    decoration: const InputDecoration(
      labelText: 'Alamat',
      prefixIcon: Icon(Icons.home),
      border: OutlineInputBorder(),
    ),
  ),
  const SizedBox(height: 16),
  TextField(
    controller: _hobiController,
```

```

        decoration: const InputDecoration(
          labelText: 'Hobi',
          prefixIcon: Icon(Icons.favorite),
          border: OutlineInputBorder(),
        ),
      ),
      const SizedBox(height: 16),
      ElevatedButton(
        onPressed: _addData,
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.yellow,
          foregroundColor: Colors.black,
        ),
        child: const Text('Simpan'),
      ),
    ],
  ),
),
);
}
}

```

db_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: (db, version) async {
      await db.execute('''
        CREATE TABLE mahasiswa (
          id INTEGER PRIMARY KEY AUTOINCREMENT,
          nama TEXT NOT NULL,
          nim TEXT NOT NULL,
          alamat TEXT NOT NULL,
          hobi TEXT NOT NULL
        )
      ''');
    },
  );
}

Future<int> insert(Map<String, dynamic> row) async {

```

```

        final db = await database;

        return await db.insert('mahasiswa', row);
    }

    Future<List<Map<String, dynamic>>> queryAllRows()
    async {
        final db = await database;

        return await db.query('mahasiswa');
    }

    Future<int> delete(int id) async {
        final db = await database;

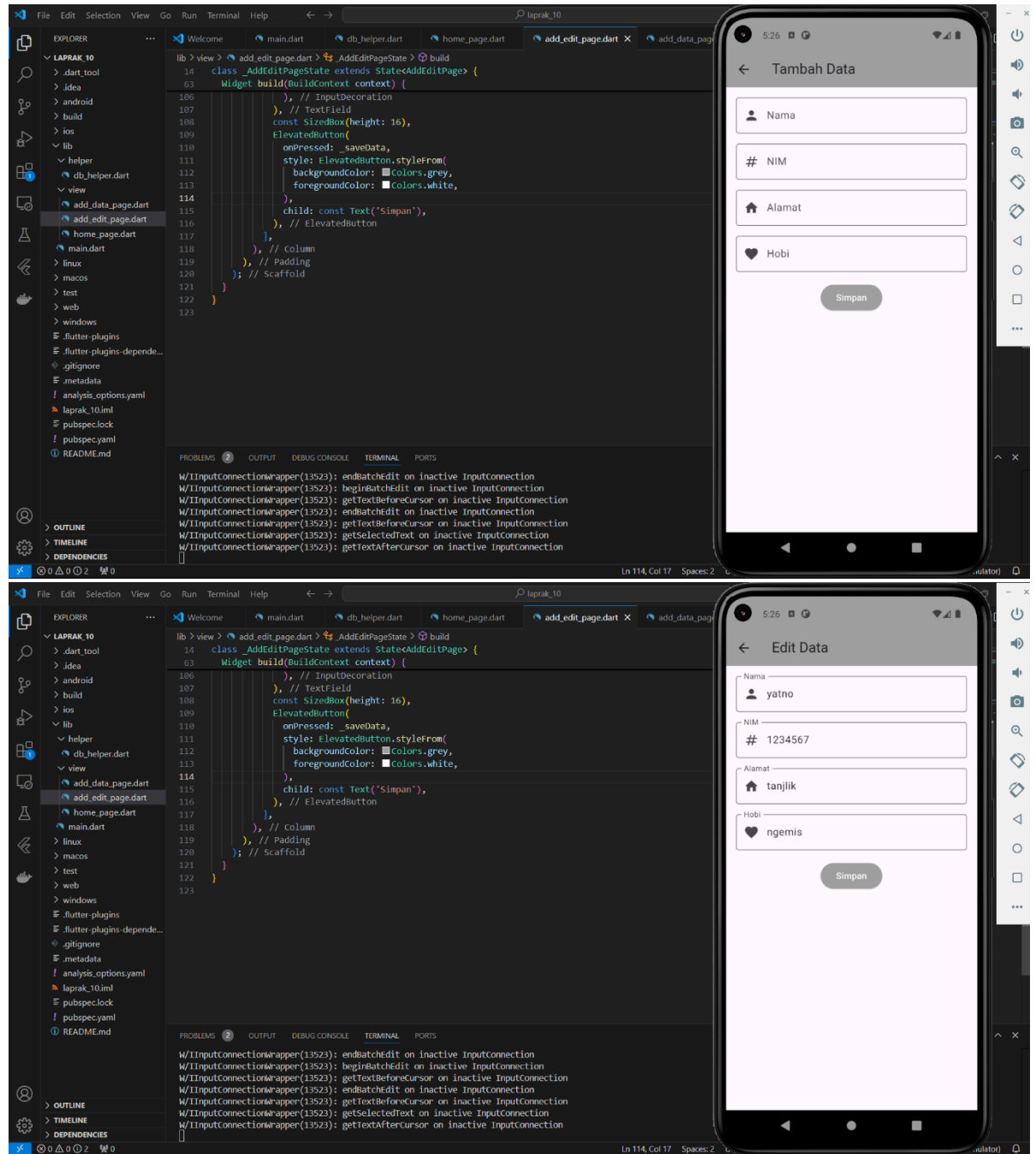
        return await db.delete(
            'mahasiswa',
            where: 'id = ?',
            whereArgs: [id],
        );
    }

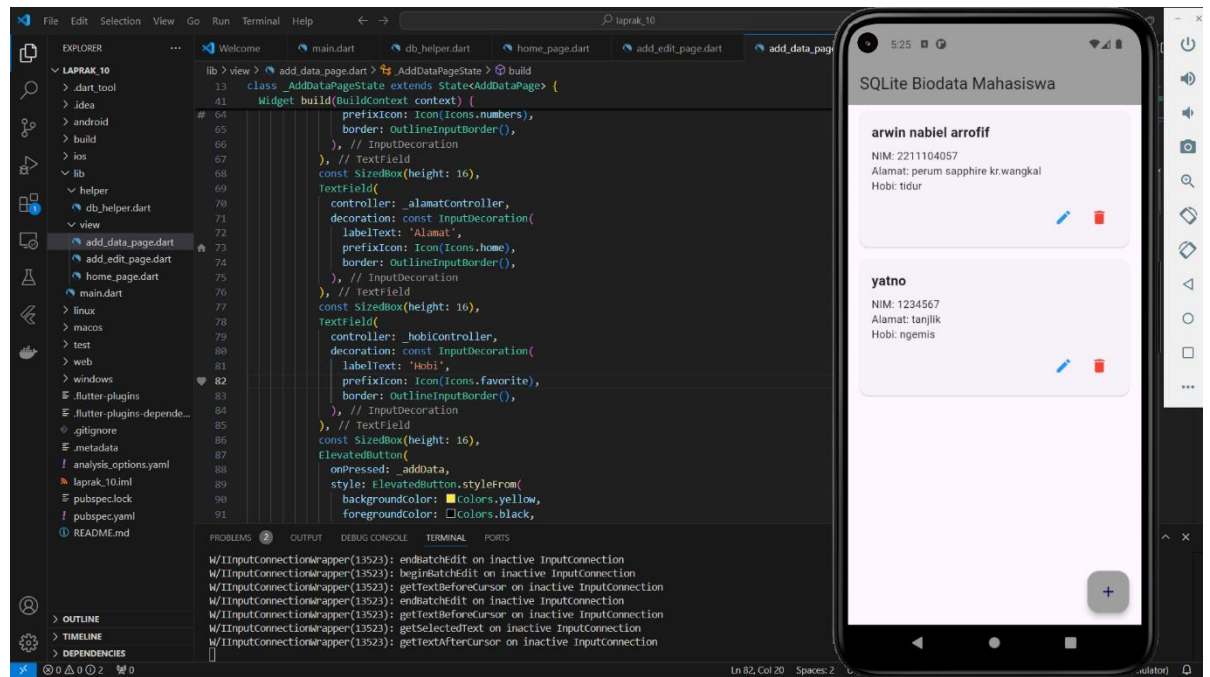
    Future<int> update(Map<String, dynamic> row) async {
        final db = await database;

        return await db.update(
            'mahasiswa',
            row,
            where: 'id = ?',
            whereArgs: [row['id']],
        );
    }
}

```

Output:





Deskripsi Program

Aplikasi SQLite Biodata Mahasiswa berbasis Flutter memungkinkan pengguna untuk menambah, melihat, mengedit, dan menghapus data mahasiswa (nama, NIM, alamat, hobi). Data disimpan secara lokal menggunakan SQLite, mendukung penggunaan tanpa koneksi internet. Halaman utama menampilkan daftar mahasiswa dalam bentuk Card, dengan tombol Edit untuk mengubah data dan Hapus untuk menghapus data. Form tambah/edit data memiliki navigasi yang intuitif dan tema abu-abu modern, menjadikan aplikasi ini efisien untuk pengelolaan data secara offline.

Teknologi yang Digunakan

1. Flutter

- Digunakan untuk membangun antarmuka pengguna yang interaktif dan modern.

2. SQLite

- Basis data lokal untuk menyimpan data mahasiswa.

3. Dart

- Bahasa pemrograman utama untuk logika aplikasi.

Fitur Utama

1. Tampilkan Daftar Mahasiswa

- Aplikasi menampilkan semua data mahasiswa yang telah disimpan dalam bentuk **Card**.
- Data yang ditampilkan mencakup nama, NIM, alamat, dan hobi.

2. Edit Data Mahasiswa

- Pengguna dapat mengedit data mahasiswa yang sudah ada.
- Saat tombol **Edit** ditekan, aplikasi akan membuka halaman form dengan data yang dapat disunting.
- Setelah disimpan, data akan diperbarui di database dan ditampilkan kembali pada halaman utama.

3. Hapus Data Mahasiswa

- Pengguna dapat menghapus data mahasiswa tertentu menggunakan tombol **Hapus** (ikon tong sampah) pada daftar mahasiswa.

4. Navigasi Halaman

Aplikasi menyediakan navigasi yang intuitif antara halaman utama (daftar mahasiswa) dan halaman tambah/edit data.