

LAPORAN GUIDED & UNGUIDED
PEMROGRAMAN PERANGKAT BERGERAK
MODUL X
DATA STORAGE (BAGIAN I)



Disusun Oleh :
Ahmad Junaidi / 2211104002
SE-06-01

Asisten Praktikum :
Ayu Susilowati
Noviana Rizki Anisa Putri

Dosen Pengampu :
Yudha Islami Sulistya, S.Kom., M.Cs.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

GUIDED

Source Code main.dart:

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false, // Menghilangkan banner
      debug
      title: 'SQLite CRUD Example',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: MyDbView(), // Menampilkan halaman utama (MyDbView)
    );
  }
}
```

Source Code db_helper.dart:

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

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

  factory DatabaseHelper() {
    return _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_prakdatabase.db');
    return await openDatabase(
        path,
        version: 1,
        onCreate: _onCreate,
    );
}

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
        )
    """);
}

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

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

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]);
}

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

```

```
}
```

Source Code my_db_view.dart

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

class MyDbView extends StatefulWidget {
  @override
  _MyDbViewState createState() => _MyDbViewState();
}

class _MyDbViewState extends State<MyDbView> {
  final dbHelper = DatabaseHelper();
  List<Map<String, dynamic>> data = [];

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

  void _fetchData() async {
    final allRows = await dbHelper.queryAllRows();
    setState(() {
      data = allRows;
    });
  }

  void _insertData(String title, String description) async {
    await dbHelper.insert({'title': title, 'description': description});
    _fetchData();
  }

  void _updateData(int id, String title, String description) async {
    await dbHelper
      .update({'id': id, 'title': title, 'description': description});
    _fetchData();
  }

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

```

}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('SQLite CRUD Example'),
    ),
    body: Column(
      children: [
        Expanded(
          child: ListView.builder(
            itemCount: data.length,
            itemBuilder: (context, index) {
              return ListTile(
                title: Text(data[index]['title']),
                subtitle: Text(data[index]['description']),
                trailing: Row(
                  mainAxisAlignment: MainAxisAlignment.min,
                  children: [
                    IconButton(
                      icon: Icon(Icons.edit),
                      onPressed: () {
                        _updateData(data[index]['id'], 'Updated Title',
                          'Updated Description');
                      },
                    ),
                    IconButton(
                      icon: Icon(Icons.delete),
                      onPressed: () {
                        _deleteData(data[index]['id']);
                      },
                    ),
                  ],
                ),
              );
            },
          ),
        Padding(
          padding: const EdgeInsets.all(8.0),
          child: ElevatedButton(
            onPressed: () {
              _insertData('New Title', 'New Description');
            },
          ),
        ),
      ],
    ),
  );
}

```

```
        child: Text('Add New Data'),
      ),
    ),
  ],
),
);
}
}
```

Output Programs GUIDED:

SQLite CRUD Example

Add New Data

UNGUIDED

Jawab:

Souce Code main.dart:

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: HomePage(),
    );
  }
}
```

Souce Code db_helper.dart:

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

class DatabaseHelper {
  // Singleton Instance
  static final DatabaseHelper _instance = DatabaseHelper._internal();
  factory DatabaseHelper() => _instance;
  DatabaseHelper._internal();

  static Database? _database;

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



```

// Inisialisasi database
Future<Database> _initDatabase() async {
  String path = join(await getDatabasesPath(), 'biodata.db'); // Path
database
  print('Database Path: $path'); // Debugging database path
  return await openDatabase(
    path,
    version: 1,
    onCreate: _onCreate, // Membuat tabel saat pertama kali database
dibuat
  );
}

// Membuat tabel SQLite
Future<void> _onCreate(Database db, int version) async {
  await db.execute("""
    CREATE TABLE biodata (
      id INTEGER PRIMARY KEY AUTOINCREMENT,
      nama TEXT,
      nim TEXT,
      alamat TEXT,
      hobi TEXT
    )
  """);
  print('Tabel biodata berhasil dibuat');
}

// Fungsi CRUD
// Insert data
Future<int> insert(Map<String, dynamic> row) async {
  Database db = await database;
  print('Menyimpan data: $row');
  return await db.insert('biodata', row);
}

// Query semua data
Future<List<Map<String, dynamic>>> queryAllRows() async {
  Database db = await database;
  List<Map<String, dynamic>> result = await db.query('biodata');
  print('Data yang dimuat: $result'); // Debugging loaded data
  return result;
}

// Update data
Future<int> update(Map<String, dynamic> data) async {

```

```

    Database db = await database;
    int id = data['id'];
    return await db.update('biodata', data, where: 'id = ?', whereArgs:
[id]);
  }

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

```

Source Code input.dart:

```

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

class InputPage extends StatefulWidget {
  @override
  _InputPageState createState() => _InputPageState();
}

class _InputPageState extends State<InputPage> {
  final _formKey = GlobalKey<FormState>();
  final _namaController = TextEditingController();
  final _nimController = TextEditingController();
  final _alamatController = TextEditingController();
  final _hobiController = TextEditingController();

  // Fungsi untuk menyimpan data
  void _saveData() async {
    if (_formKey.currentState!.validate()) {
      final dbHelper = DatabaseHelper();
      int result = await dbHelper.insert({
        'nama': _namaController.text,
        'nim': _nimController.text,
        'alamat': _alamatController.text,
        'hobi': _hobiController.text,
      });
      print('Data berhasil disimpan dengan ID: $result'); // Debug log
      Navigator.pop(context); // Kembali ke halaman utama
    }
  }
}

```

```

}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: Colors.orange,
      title: Text('Tambah Biodata Mahasiswa'),
      centerTitle: true,
    ),
    body: Padding(
      padding: EdgeInsets.all(20.0),
      child: Form(
        key: _formKey,
        child: Column(
          children: [
            TextFormField(
              controller: _namaController,
              decoration: InputDecoration(
                labelText: 'Nama',
                labelStyle: TextStyle(color: Colors.orange),
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(12),
                  borderSide: BorderSide(color: Colors.orange),
                ),
              ),
              validator: (value) =>
                value!.isEmpty ? 'Nama tidak boleh kosong' : null,
            ),
            SizedBox(height: 15),
            TextFormField(
              controller: _nimController,
              decoration: InputDecoration(
                labelText: 'NIM',
                labelStyle: TextStyle(color: Colors.orange),
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(12),
                  borderSide: BorderSide(color: Colors.orange),
                ),
              ),
              validator: (value) =>
                value!.isEmpty ? 'NIM tidak boleh kosong' : null,
            ),
            SizedBox(height: 15),
            TextFormField(

```

```

        controller: _alamatController,
        decoration: InputDecoration(
          labelText: 'Alamat',
          labelStyle: TextStyle(color: Colors.orange),
          border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(12),
            borderSide: BorderSide(color: Colors.orange),
          ),
        ),
        validator: (value) =>
          value!.isEmpty ? 'Alamat tidak boleh kosong' : null,
      ),
      SizedBox(height: 15),
      TextFormField(
        controller: _hobiController,
        decoration: InputDecoration(
          labelText: 'Hobi',
          labelStyle: TextStyle(color: Colors.orange),
          border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(12),
            borderSide: BorderSide(color: Colors.orange),
          ),
        ),
        validator: (value) =>
          value!.isEmpty ? 'Hobi tidak boleh kosong' : null,
      ),
      SizedBox(height: 20),
      ElevatedButton(
        onPressed: _saveData,
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.orange, // Button color
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(12),
          ),
        ),
        child: Text('Simpan', style: TextStyle(fontSize: 16)),
      ),
    ),
  ),
),
);
}
}

```

Source Code home.dart:

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

class HomePage extends StatefulWidget {
  @override
  _HomePageState createState() => _HomePageState();
}

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

  // Fungsi untuk memuat data dari database
  void _loadData() async {
    final data = await DatabaseHelper().queryAllRows();
    print('Data yang dimuat: $data'); // Debug log
    setState() {
      _biodata = data;
    });
  }

  @override
  void initState() {
    super.initState();
    _loadData(); // Memuat data ketika halaman pertama kali dibuka
  }


  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.orange,
        title: Text('SQLite Biodata Mahasiswa',
          style: TextStyle(color: Colors.white)),
        centerTitle: true,
      ),
      body: _biodata.isEmpty
        ? Center(
            child: Text('Tidak ada data mahasiswa',
              style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold)))
        : ListView.builder(
            itemCount: _biodata.length,
```

```

itemBuilder: (context, index) {
  final item = _biodata[index];
  return Card(
    margin: EdgeInsets.symmetric(vertical: 10, horizontal: 15),
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(15)),
    elevation: 5,
    child: ListTile(
      contentPadding: EdgeInsets.all(15),
      title: Text(item['nama'],
        style: TextStyle(
          fontSize: 18, fontWeight: FontWeight.bold)),
      subtitle: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text('NIM: ${item['nim']}',
            style: TextStyle(fontSize: 14)),
          Text('Alamat: ${item['alamat']}',
            style: TextStyle(fontSize: 14)),
          Text('Hobi: ${item['hobi']}',
            style: TextStyle(fontSize: 14)),
        ],
      ),
    ),
  );
},
),
floatingActionButton: FloatingActionButton(
  onPressed: () async {
    await Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => InputPage()),
    );
    _loadData(); // Memuat ulang data setelah menambah data
  },
  backgroundColor: Colors.orange,
  child: Icon(Icons.add, color: Colors.white),
),
);
}
}

```

Output Programs UNGUIDED

 **Tambah Biodata Mahasiswa**

Nama

Ahmad Junaidi

NIM

2211104002

Alamat

Jatimulyo

Hobi

Main bola

Simpan

SQLite Biodata Mahasiswa

Tidak ada data mahasiswa



Deskripsi Program

program ini adalah aplikasi Flutter yang menggunakan SQLite untuk menyimpan dan mengelola biodata mahasiswa secara offline. Aplikasi ini terdiri dari dua tampilan utama: halaman utama (HomePage) dan halaman input (InputPage). Pada halaman utama, pengguna dapat melihat daftar biodata mahasiswa yang tersimpan di dalam database, termasuk nama, NIM, alamat, dan hobi. Setiap entri biodata ditampilkan dalam bentuk ListView dengan desain yang menarik menggunakan widget Card, yang menambahkan border rounded dan shadow untuk memberikan tampilan yang lebih modern.

Untuk menambahkan data baru, pengguna dapat mengklik tombol floating action button yang akan mengarahkan mereka ke halaman input. Di halaman input, pengguna dapat mengisi form dengan data biodata mahasiswa seperti nama, NIM, alamat, dan hobi. Setelah form diisi dan tombol "Simpan" diklik, data akan disimpan dalam database SQLite dan halaman akan kembali ke halaman utama. Aplikasi ini juga memuat data yang ada di database setiap kali aplikasi dibuka atau setelah data baru disimpan, untuk memastikan bahwa informasi yang ditampilkan selalu terbaru.

Secara keseluruhan, aplikasi ini memungkinkan pengguna untuk mengelola data biodata mahasiswa dengan antarmuka yang sederhana namun modern, menggunakan SQLite sebagai solusi penyimpanan data lokal. Desain aplikasi ini mengutamakan pengalaman pengguna yang intuitif dan estetik, dengan elemen UI seperti tombol yang besar, form input yang mudah diisi, dan tampilan data yang rapi dan jelas.