LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK

MODUL XIII

NETWORKING

Disusun Oleh :

Devrin Anggun Saputri / 2211104001

SE0601

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

NETWORKING/STATE MANAGEMENT

State management dalam Flutter adalah proses mengelola state atau status dari aplikasi, yaitu data atau informasi yang dapat berubah sepanjang siklus hidup aplikasi. State ini mencakup segala hal yang memengaruhi tampilan antarmuka pengguna (UI), seperti input pengguna, data dari API, dan status internal widget. Ketika aplikasi semakin kompleks dibuat, maka pasti akan ada saatnya dimana harus dibagikan state aplikasi ke berbagai halaman yang ada.

Flutter adalah deklaratif, sehingga Flutter membangun user interface berdasarkan state saat ini. Dengan menggunakan state management, dapat dilakukan sentralisasi semua state dari berbagai macam UI Control untuk mengendalikan aliran data lintas aplikasi.

State management penting karena aplikasi Flutter sering kali terdiri dari banyak widget yang saling terkait. Dengan mengelola state dengan baik, kita dapat memastikan :

- Sinkronisasi UI dan data, karena selalu mencerminkan data terkini.
- Organisasi kode yang baik untuk mempermudah pengembangan dan pemeliharaan.
- Pengurangan bug, karena state yang dikelola dengan benar mengurangi kemungkinan terjadinya bug.

Jenis State dalam Flutter

1. **Ephemeral State (State Lokal)**

   State ini hanya relevan untuk widget tertentu dan tidak dibagikan ke widget lain. Contohnya adalah state untuk TextField atau Checkbox. Dan kita dapat menggunakan StatefulWidget untuk mengelola ephemeral state. Pendekatannya state menaganement-nya ada dua, yakni StatefulWidget (untuk ephemeral state) dan InheritedWidget (untuk berbagai state antar widget)

2. **App State (State Global)**

   State ini digunakan di berbagai widget dalam aplikasi. Contohnya adalah informasi pengguna yang masuk, data keranjang belanja, atau tema aplikasi. App state biasanya membutuhkan pendekatan state management yang lebih kompleks. Package/library pendukung Flutter memiliki berbagai framework atau package untuk state management, seperti :

   A. **Provider**

      Provider adalah library state management yang didukung resmi oleh tim Flutter. Provider memanfaatkan kemampuan bawaan Flutter seperti InheritedWidget, tetapi dengan cara yang lebih sederhana dan efisien.

   B. **BloC/Cubit**

      Bloc (Business Logic Component) adalah pendekatan state management berbasis pola stream. Bloc memisahkan business logic dari UI, sehingga cocok untuk aplikasi yang besar dan kompleks.

   C. **Riverpod**

      Riverpod adalah framework state management modern yang dirancang sebagai pengganti atau alternatif untuk Provider. Riverpod lebih fleksibel dan mengatasi beberapa keterbatasan Provider.

D. GetX

GetX adalah framework Flutter serbaguna yang menyediakan solusi lengkap untuk state management, routing, dan dependency injection. GetX dirancang untuk meminimalkan boilerplate code, meningkatkan efisiensi, dan mempermudah pengembangan aplikasi Flutter, terutama yang memerlukan reaktivitas tinggi.

Berikut cara instalasi GetX:

1. **Tambahkan GetX ke dalam proyek Flutter melalui pubspec.yaml :**

```
dependencies:
  flutter:
    sdk: flutter
  cupertino_icons: ^1.0.8
  get: ^4.6.6
```

2. **Konfigurasi dasar**

   Untuk menggunakan GetX, ubah root aplikasi dengan mengganti MaterialApp menjadi GetMaterialApp :

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

3. **State Management dengan GetX**

   a. **Membuat Controller**

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

class CounterController extends GetxController {
  var counter = 0.obs;

  void incrementCounter() {
    counter++;
  }
```

   b. **Menggunakan Controller di UI**

   - Tambahkan controller ke dalam widget menggunakan Get.put() untuk dependency injection.
   - Gunakan Obx untuk memantau perubahan state.

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

class CounterController extends GetxController {
  var counter = 0.obs;

  void incrementCounter() {
    counter++;
  }

  void decrementCounter() {
    counter--;
  }
  void getsnackbar() {
    Get.snackbar(
      'Get Snackbar',
      'Ini adalah getx snackbar',
      backgroundColor: Colors.green,
      colorText: Colors.white
      );
```

```dart
  }
  void getbottonsheet() {
    Get.bottomSheet(Container(
      height: 200,
      color: Colors.green,
      child: const Center(
        child: Text(
          'Ini adalah getx botton sheet',
          style: TextStyle(
            color: Colors.white,
            fontSize: 20,
          ),
        ),
      ),
    ));
  }
}
```

4. **Routing dengan GetX**
   a. **Definisikan Route**

   Gunakan GetPage pada main.dart untuk mendefinisikan rute aplikasi :

```dart
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:praktikum_13/view/detail_page.dart';
import 'package:praktikum_13/view/my_home_page.dart';

Run | Debug | Profile
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      initialRoute: '/',
      getPages: [
        GetPage(name: '/', page: () => MyHomePage(title: 'Belajar GetX')),
        GetPage(name: '/detail', page: () => DetailPage()),
      ],
    ); // GetMaterialApp
  }
}
```

   b. **Navigasi**
      - Get.to() : Navigasi ke halaman baru.
      - Get.back() : Kembali ke halaman sebelumnya.
      - Get.off() : Menghapus semua halaman sebelumnya.
      - Get.offAll() : Menghapus semua halaman dalam stack

5. **Dependency Injection dengan GetX**
   a. Injeksi Sederhana
   b. Lazy Loading
   c. Mengambil Instance
6. **Snackbar**

```dart
void getsnackbar() {
  Get.snackbar(
```

7. **Dialog**
8. **BottomSheet**

Source Code:

main.dart

```dart
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:praktikum_13/view/detail_page.dart';
import 'package:praktikum_13/view/my_home_page.dart';

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

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      initialRoute: '/',
      getPages: [
        GetPage(name: '/', page: () => MyHomePage(title: 'Belajar GetX')),
        GetPage(name: '/detail', page: () => DetailPage()),
      ],
    ); // GetMaterialApp
  }
}
```

detail_page.dart

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

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Text('Detail Page'),
      ), // Center
    ); // Scaffold
  }
}
```

my_home_page.dart

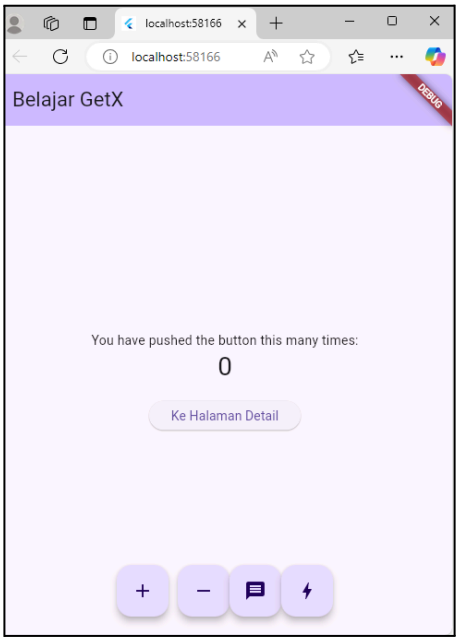
```dart
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:praktikum_13/view/detail_page.dart';
import 'package:praktikum_13/view_model/counter_controller.dart';

class MyHomePage extends StatelessWidget {
  MyHomePage({super.key, required this.title});

  final String title;

  @override
  Widget build(BuildContext context) {
    // Menginisialisasi controller
    final CounterController controller = Get.put(CounterController());

    return Scaffold(
      appBar: AppBar(
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: Text(title),
      ),
```

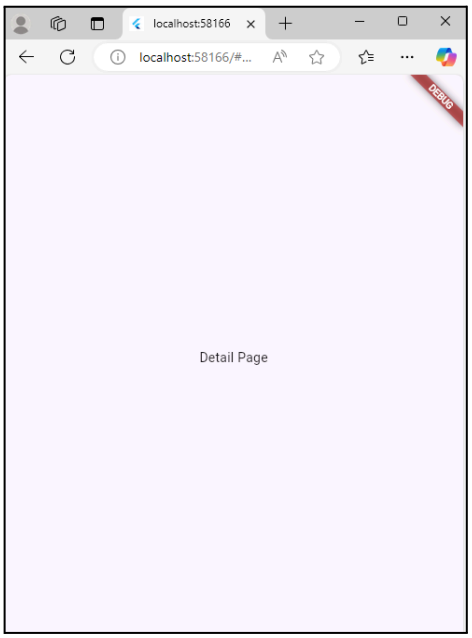
```
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            const Text(
              'You have pushed the button this many times:',
            ),
            Obx(() => Text(
                  controller.counter.value.toString(),
                  style: Theme.of(context).textTheme.headlineMedium,
                )),
            const SizedBox(height: 20),
            ElevatedButton(
              onPressed: () {
                Get.to(DetailPage());
              },
              child: const Text('Ke Halaman Detail'),
            ),
          ],
        ),
      ),
      floatingActionButton: Container(
        alignment: Alignment.bottomCenter,
        padding: const EdgeInsets.only(bottom: 20), // Jarak dari bawah layar
        child: Row(
          mainAxisSize: MainAxisSize.min, // Membatasi ukuran ke konten tombol
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            FloatingActionButton(
              onPressed: controller.incrementCounter,
              tooltip: 'Increment',
              child: const Icon(Icons.add),
            ),
            const SizedBox(width: 10),
            FloatingActionButton(
              onPressed: controller.decrementCounter,
              tooltip: 'Decrement',
              child: const Icon(Icons.remove),
            ),
            FloatingActionButton(
            onPressed: controller.getsnackbar,
            tooltip: 'Get Snackbar',
            child: const Icon(Icons.message),
          ),
            FloatingActionButton(
            onPressed: controller.getbottonsheet,
            tooltip: 'Get Bottom Sheet',
            child: const Icon(Icons.bolt_outlined),
          ),
          ],
        ),
      ),
      floatingActionButtonLocation: FloatingActionButtonLocation.centerDocked, // Tambahkan ini agar tombol berada di tengah
    );
  }
}
```

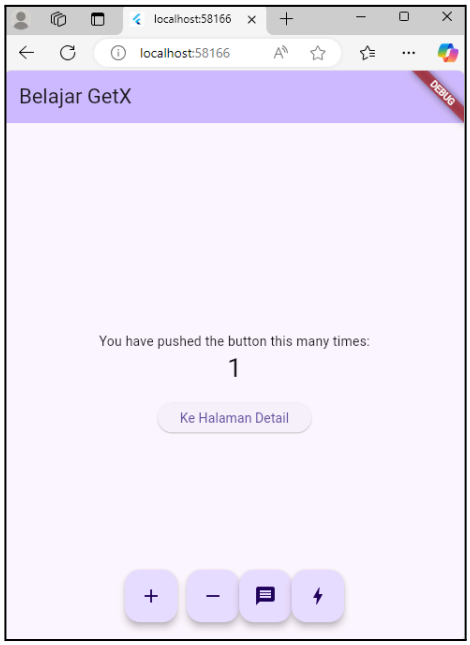
counter_controller.dart

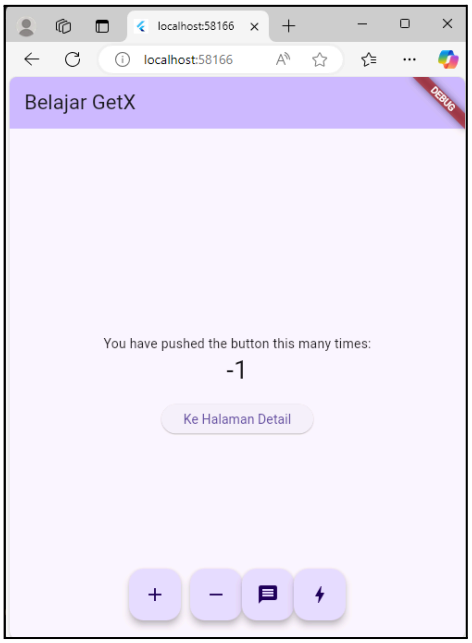
```dart
import 'package:flutter/material.dart';
import 'package:get/get.dart';

class CounterController extends GetxController {
  var counter = 0.obs;

  void incrementCounter() {
    counter++;
  }

  void decrementCounter() {
    counter--;
  }
  void getsnackbar() {
    Get.snackbar(
      'Get Snackbar',
      'Ini adalah getx snackbar',
      backgroundColor: Colors.green,
      colorText: Colors.white
      );
  }
  void getbottonsheet() {
    Get.bottomSheet(Container(
      height: 200,
      color: Colors.green,
      child: const Center(
        child: Text(
          'Ini adalah getx botton sheet',
          style: TextStyle(
            color: Colors.white,
            fontSize: 20,
          ),
        ),
      ),
    ));
  }
}

```

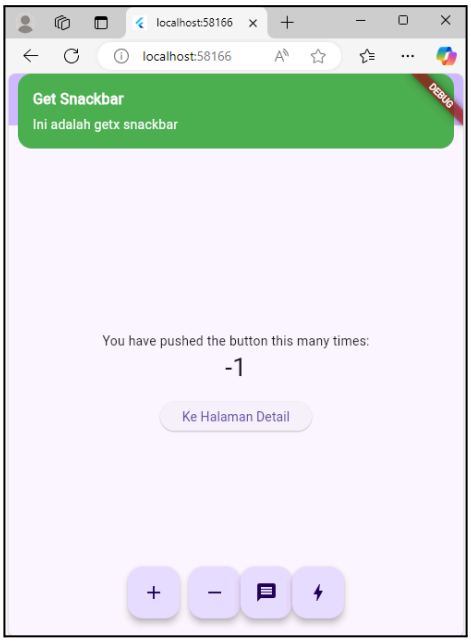
Output:

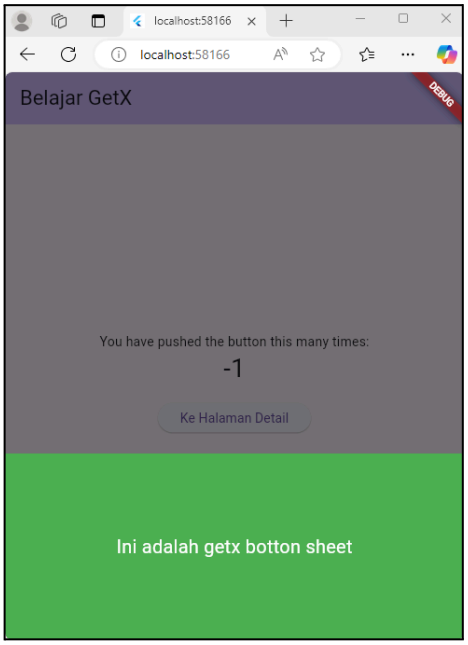

localhost:58166
Belajar GetX
DEBUG
You have pushed the button this many times:
0
Ke Halaman Detail


localhost:58166
DEBUG
Detail Page


localhost:58166
Belajar GetX
DEBUG
You have pushed the button this many times:
1
Ke Halaman Detail


localhost:58166
Belajar GetX
DEBUG
You have pushed the button this many times:
-1
Ke Halaman Detail


localhost:58166
Get Snackbar
Ini adalah getx snackbar
DEBUG
You have pushed the button this many times:
-1
Ke Halaman Detail


localhost:58166
Belajar GetX
DEBUG
You have pushed the button this many times:
-1
Ke Halaman Detail
Ini adalah getx botton sheet

Deskripsi Program:

Kode di atas merupakan contoh aplikasi Flutter yang menggunakan paket GetX untuk mengelola navigasi dan pengelolaan state. Aplikasi ini memiliki dua halaman utama, yaitu MyHomePage dan DetailPage. Pada halaman utama, terdapat sebuah *counter* yang dapat ditambah atau dikurangi nilainya melalui tombol *floating action button* (FAB), yang dikelola oleh CounterController dengan memanfaatkan *state* reaktif menggunakan RxInt (*obs*). Perubahan nilai *counter* secara otomatis akan diperbarui di antarmuka pengguna menggunakan widget Obx.

Selain itu, aplikasi ini juga dilengkapi tombol tambahan yang memanfaatkan fitur GetX, seperti menampilkan *snackbar* dan *bottom sheet*, yang diaktifkan melalui metode Get.snackbar dan Get.bottomSheet. Navigasi antar halaman dilakukan menggunakan metode Get.to() untuk berpindah ke halaman detail ketika tombol "Ke Halaman Detail" ditekan. CounterController juga mengelola berbagai aksi, termasuk pengaturan *snackbar* dan *bottom sheet*. Kode ini menunjukkan bagaimana GetX dapat mempermudah pengelolaan *state*, navigasi, dan antarmuka yang interaktif di Flutter.

UNGUIDED

SOAL

Buatlah Aplikasi Catatan Sederhana menggunakan GetX, dengan ketentuan sebagai

berikut :

1. Halaman utama atau Homepage untuk menampilkan daftar catatan yang telah ditambahkan. Setiap catatan terdiri dari judul dan deskripsi singkat, serta terdapat tombol untuk menghapus catatan dari daftar.
2. Halaman kedua untuk menambah catatan baru, berisi : form untuk memasukkan judul dan deskripsi catatan, serta tombol untuk menyimpan catatan ke daftar (Homepage).
3. Menggunakan getx controller.
4. Menggunakan getx routing untuk navigasi halaman.

Note: Jangan lupa sertakan source code, screenshoot output, dan deskripsi program.

Kreatifitas menjadi nilai tambah

Source Code:

- **main.dart**

lib > main.dart > MyApp

```dart
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:unguided_13/view/home_page.dart';
import 'package:unguided_13/view/add_note_page.dart';

Run | Debug | Profile
void main() {
  runApp(const MyApp());
}

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

  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      initialRoute: '/',
      getPages: [
        GetPage(name: '/', page: () => const HomePage()),
        GetPage(name: '/addNote', page: () => const AddNotePage()),
      ],
    ); // GetMaterialApp
  }
}
```

- **note.dart**

```dart
class Note {
  String title;
  String description;

  Note({
    required this.title,
    required this.description,
  });
}
```

- **add_note_page.dart**

lib > view > add_note_page.dart > AddNotePage > build

```dart
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:unguided_13/view_model/note_controller.dart';

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

  @override
  Widget build(BuildContext context) {
    final TextEditingController titleController = TextEditingController();
    final TextEditingController descriptionController = TextEditingController();
    final NoteController controller = Get.find();

    return Scaffold(
      appBar: AppBar(
        title: const Text('Tambah Catatan'),
        backgroundColor: const Color.fromARGB(255, 44, 156, 201), // Mengubah warna AppBar menjadi ungu
      ), // AppBar
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            const Text('Judul'),
            TextField(
              controller: titleController,
              decoration: const InputDecoration(
                hintText: 'Masukkan judul catatan',
                border: OutlineInputBorder(), // Membuat kotak
              ), // InputDecoration
            ), // TextField
            const SizedBox(height: 16),
            const Text('Deskripsi'),
            TextField(
              controller: descriptionController,
              decoration: const InputDecoration(
                hintText: 'Masukkan deskripsi catatan',
                border: OutlineInputBorder(), // Membuat kotak
              ), // InputDecoration
              maxLines: 5, // Tambahan agar lebih cocok untuk deskripsi panjang
            ), // TextField
            const SizedBox(height: 32),
            ElevatedButton(
              onPressed: () {
                if (titleController.text.isNotEmpty &&
                    descriptionController.text.isNotEmpty) {
                  controller.addNote(
                    titleController.text,
                    descriptionController.text,
                  );
                  Get.back(); // Kembali ke halaman utama
                }
              },
              child: const Text('Simpan Catatan'),
            ), // ElevatedButton
          ],
        ), // Column
      ), // Padding
    ); // Scaffold
  }
}
```

- **home_page.dart**

lib > view > home_page.dart > HomePage > build

```dart
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:unguided_13/view_model/note_controller.dart';
import 'package:unguided_13/view/add_note_page.dart';

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

  @override
  Widget build(BuildContext context) {
    final NoteController controller = Get.put(NoteController());

    return Scaffold(
      appBar: AppBar(
        title: const Text('Daftar Catatan'),
        backgroundColor: const Color.fromARGB(255, 44, 156, 201), // Warna AppBar
      ), // AppBar
      body: Obx(() {
        // Menampilkan daftar catatan
        if (controller.notes.isEmpty) {
          return const Center(
            child: Text('Tidak ada catatan'),
          ); // Center
        }
        return ListView.builder(
          itemCount: controller.notes.length,
          itemBuilder: (context, index) {
            final note = controller.notes[index];
            return Padding(
              padding: const EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0),
              child: Card(
                elevation: 4,
                shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(8),
                ), // RoundedRectangleBorder
                child: Padding(
                  padding: const EdgeInsets.all(12.0),
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      const Text(
                        'Judul:',
                        style: TextStyle(fontWeight: FontWeight.bold),
                      ), // Text
                      Text(
                        note.title,
                        style: const TextStyle(fontSize: 16),
                      ), // Text
                      const SizedBox(height: 8),
                      const Text(
                        'Deskripsi:',
                        style: TextStyle(fontWeight: FontWeight.bold),
                      ), // Text
                      Text(
                        note.description,
                        style: const TextStyle(fontSize: 14),
                      ), // Text
                      Align(
```

```
                                alignment: Alignment.centerRight,
                                child: IconButton(
                                  icon: const Icon(Icons.delete, color: Colors.red),
                                  onPressed: () {
                                    controller.deleteNote(index); // Menghapus catatan
                                  },
                                ), // IconButton
                              ), // Align
                            ],
                          ), // Column
                        ), // Padding
                      ), // Card
                    ); // Padding
                  },
                ); // ListView.builder
              }), // Obx
              floatingActionButton: FloatingActionButton(
                onPressed: () {
                  Get.to(const AddNotePage()); // Navigasi ke halaman tambah catatan
                },
                child: const Icon(Icons.add),
              ), // FloatingActionButton
            ); // Scaffold
          }
        }

```

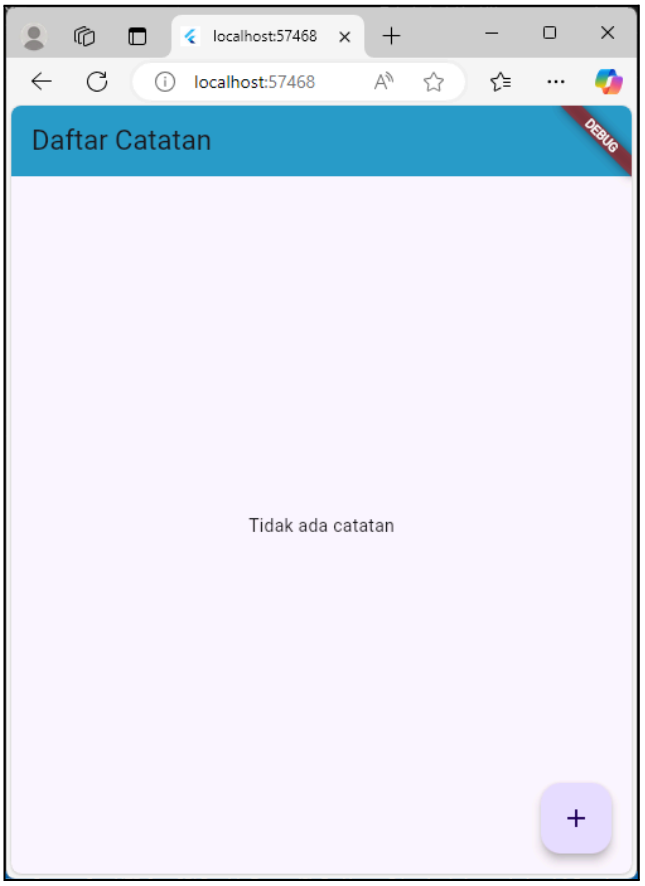
- **note_controller.dart**

```dart
import 'package:get/get.dart';
import 'package:unguided_13/models/note.dart';

class NoteController extends GetxController {
  var notes = <Note>[].obs; // Menyimpan daftar catatan

  // Fungsi untuk menambah catatan
  void addNote(String title, String description) {
    notes.add(Note(title: title, description: description));
  }

  // Fungsi untuk menghapus catatan berdasarkan index
  void deleteNote(int index) {
    notes.removeAt(index);
  }
}
```

Output:

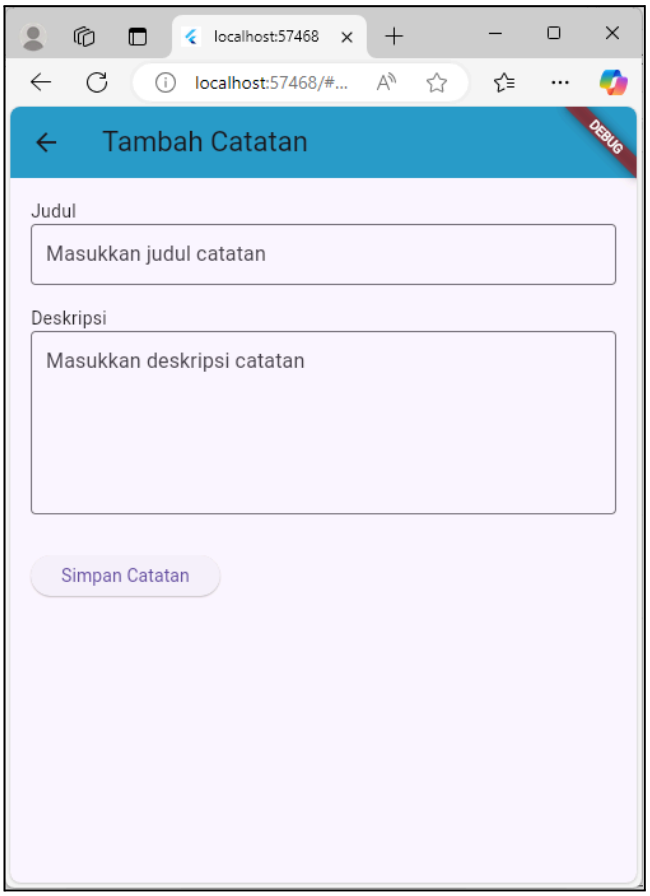
- **halaman 1**

  Menampilkan halaman awal Daftar Catatan (kosong).



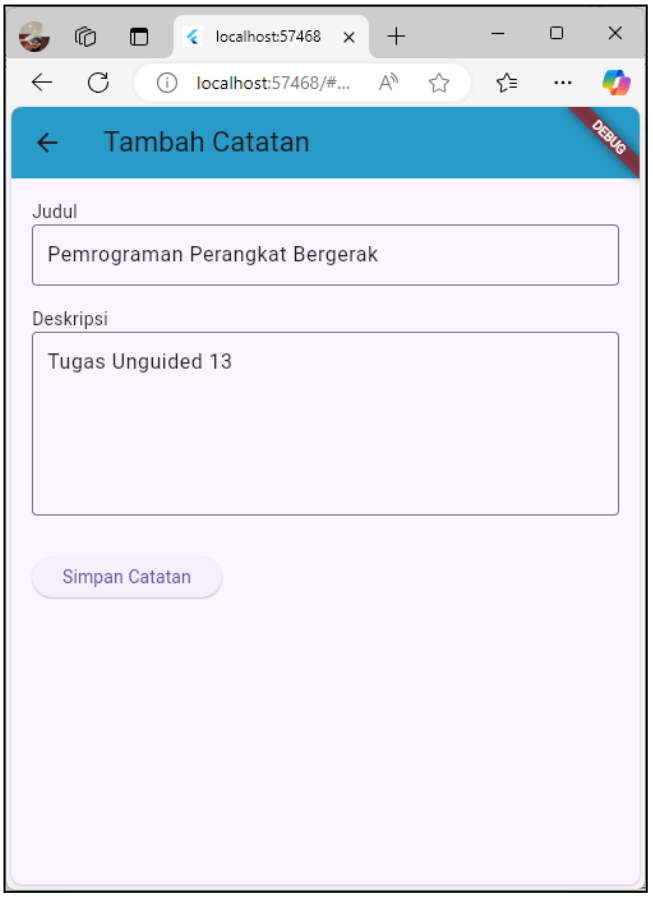
- **halaman 2**

  Menampilkan halaman tambah catatan dan simpan catatan.



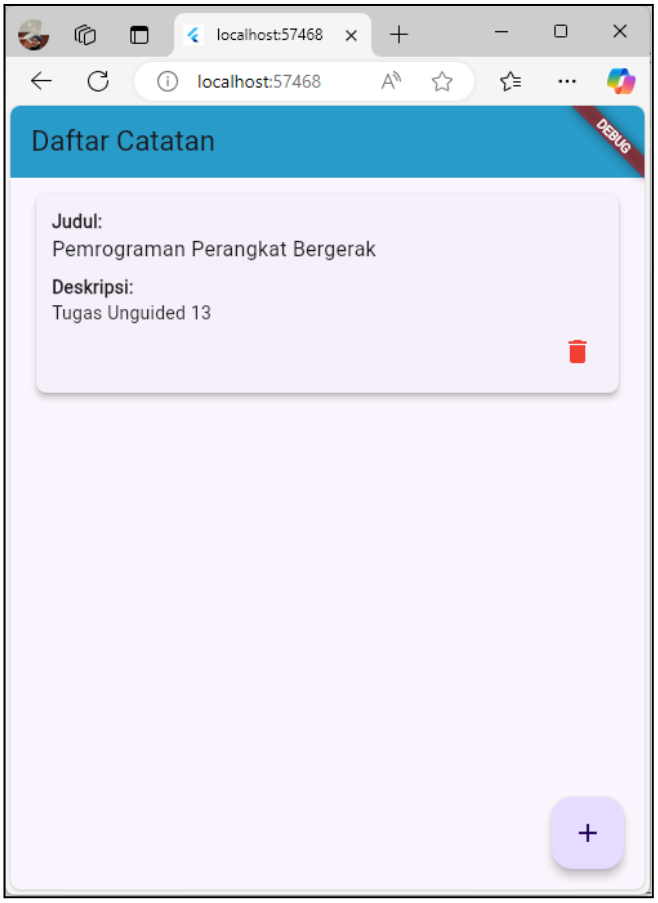
- **halaman 3**

  Menampilkan halaman tambah catatan yang sudah di isi judul dan deskripsi



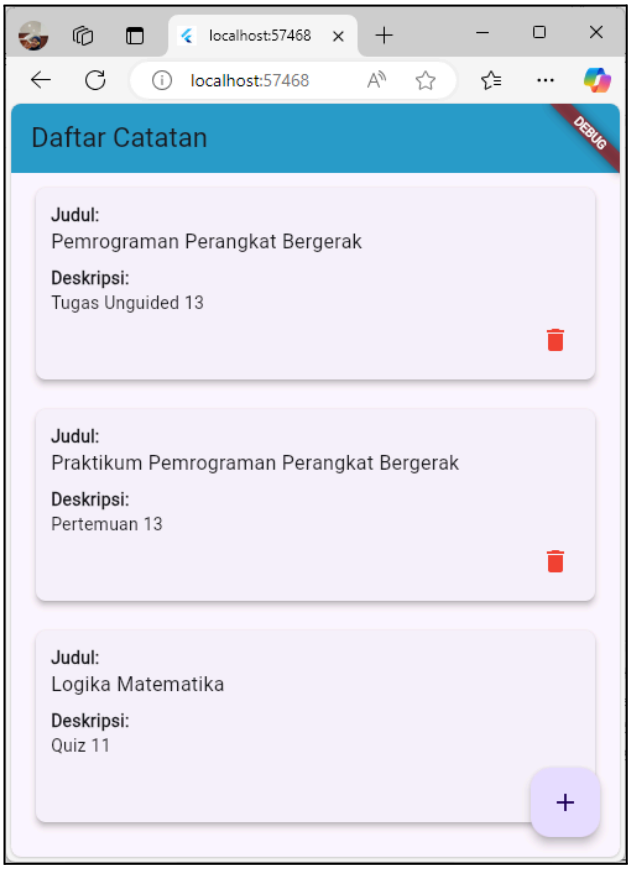
- **halaman 4**

  Menampilkan isi halaman Daftar Catatan



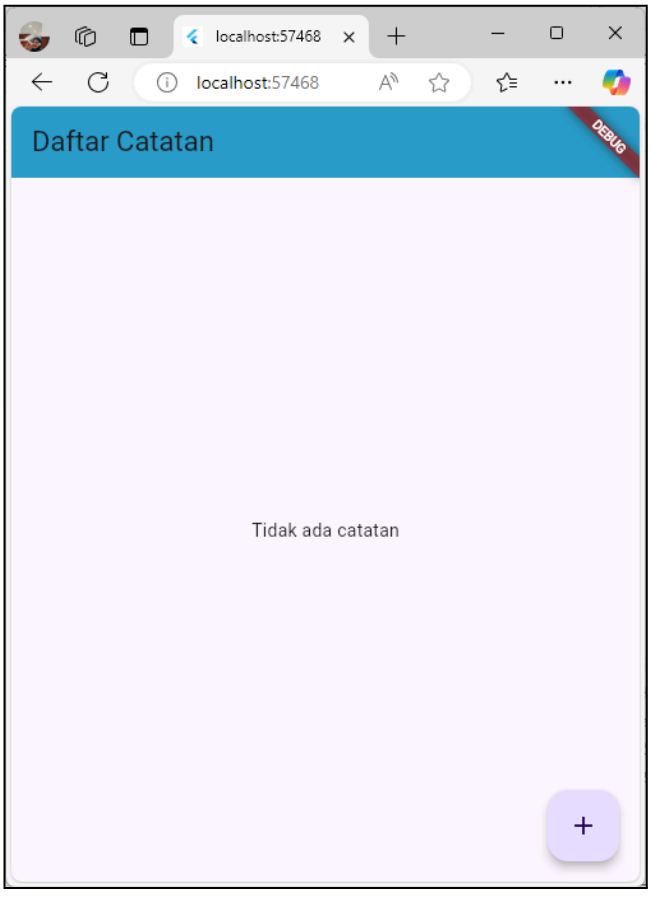
- **halaman 5**

  Menampilkan halaman Daftar Catatan yang sudah diisi beberapa judul dan deskripsi.



- **halaman 6**

  Menampilkan Daftar Catatan yang sudah dihapus dan kembali kosong seperti awal.



Deskripsi Program:

Program diatas adalah aplikasi manajemen catatan sederhana berbasis Flutter dan menggunakan *state management* dari paket GetX. Aplikasi memiliki dua fitur utama: menambah catatan baru dan melihat daftar catatan yang telah dibuat. Halaman utama menampilkan daftar catatan dalam bentuk kartu, termasuk judul dan deskripsi setiap catatan. Pengguna dapat menambahkan catatan baru melalui halaman "Tambah Catatan", di mana mereka dapat mengisi judul dan deskripsi. Selain itu, catatan yang sudah ada dapat dihapus dengan menekan ikon hapus pada masing-masing kartu. Aplikasi ini menggunakan arsitektur berbasis *controller* untuk mengelola data catatan secara reaktif, memastikan perubahan langsung terlihat di antarmuka pengguna tanpa memuat ulang halaman. Dengan antarmuka yang sederhana dan intuitif, aplikasi ini cocok untuk kebutuhan pencatatan harian.