

**LAPORAN PRAKTIKUM  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XIII  
NETWORKING**



**Disusun Oleh :**

**Dewi Atika Muthi / 2211104042**

**SE-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**

## **BAB I**

### **PENDAHULUAN**

#### **A. DASAR TEORI**

**State management** dalam Flutter merupakan konsep fundamental untuk mengelola data dan status aplikasi yang dapat berubah selama siklus hidup aplikasi. Flutter menggunakan pendekatan deklaratif, dimana UI dibangun berdasarkan state saat ini.

Terdapat **dua jenis state** dalam Flutter:

1. Ephemeral State (State Lokal)
  - Hanya relevan untuk widget tertentu
  - Tidak dibagikan ke widget lain
  - Dikelola menggunakan StatefulWidget
2. App State (State Global)
  - Digunakan di berbagai widget dalam aplikasi
  - Membutuhkan pendekatan management yang lebih kompleks
  - Contoh: informasi user login, data keranjang belanja, tema aplikasi

**GetX** merupakan salah satu solusi state management yang populer dalam Flutter karena:

- Meminimalkan boilerplate code
- Meningkatkan efisiensi pengembangan
- Menyediakan fitur routing dan dependency injection
- Memiliki performa tinggi
- Mendukung reaktivitas

#### **B. MAKSUD DAN TUJUAN**

Tujuan dalam praktikum kali ini adalah:

1. Memahami konsep state management dalam Flutter
2. Mempelajari implementasi GetX sebagai solusi state management
3. Mengembangkan aplikasi Flutter dengan fitur routing dan state management yang efisien
4. Memahami penggunaan controller untuk separasi logic dari UI

## BAB II IMPLEMENTASI

### A. PRAKTIKUM (Guided)

#### Soal Studi Case

Proyek ini mengimplementasikan aplikasi counter sederhana menggunakan GetX untuk state management.

Menyiapkan proyek Flutter dengan GetX sebagai dependensi di pubspec.yaml:

```
flutter pub add get
```

#### Sourcecode view/detail\_page.dart:

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:mdl2/view_model/counter_controller.dart';

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

  @override
  Widget build(BuildContext context) {
    final CounterController controller =
      Get.find(); // Mengambil instance CounterController

    return Scaffold(
      appBar: AppBar(
        title: Text('Detail Page'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text('Halaman Detail'),
            SizedBox(height: 20),
            ElevatedButton(
              onPressed: () {
                Get.snackbar(
                  'Info',
                  'This is a detail page.',
                  snackPosition:
                    SnackPosition.TOP, // Mengubah posisi snackbar ke atas
                  backgroundColor: Colors.green,
                  colorText: Colors.white,
                );
              },
              child: Text('Show Snackbar'),
            ),
            SizedBox(height: 20),
            Text(
              'Current Counter: ${controller.counter}', // Menampilkan counter
            ),
          ],
        ),
      ),
    );
  }
}
```

#### Sourcecode view/my\_home\_page.dart:

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:mdl2/view_model/counter_controller.dart';

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

```

final String title;
final CounterController controller = Get.put(CounterController());

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: Theme.of(context).colorScheme.inversePrimary,
      title: Text(title),
      actions: [
        IconButton(
          icon: Icon(Icons.notifications),
          onPressed: () {
            controller.getDialog(); // Memanggil fungsi getDialog
          },
        ),
      ],
    ),
    drawer: controller.getSidebar(), // Memanggil fungsi getSidebar
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          const Text(
            'You have pushed the button this many times:',
          ),
          Obx(() => Text(
            controller.counter.toString(),
            style: Theme.of(context).textTheme.headlineMedium,
          )),
          const SizedBox(height: 20), // Menambahkan jarak
          ElevatedButton(
            onPressed: () {
              Get.toNamed('/details'); // Navigasi ke DetailPage
            },
            child: const Text('View Detail'),
          ),
        ],
      ),
    ),
    floatingActionButton: Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        FloatingActionButton(
          onPressed: () {
            controller.incrementCounter(); // Memanggil increment
          },
          tooltip: 'Increment',
          child: const Icon(Icons.add),
        ),
        const SizedBox(width: 20),
        FloatingActionButton(
          onPressed: () {
            controller.decrementCounter(); // Memanggil decrement
          },
          tooltip: 'Decrement',
          child: const Icon(Icons.remove),
        ),
        const SizedBox(width: 20),
        FloatingActionButton(
          onPressed: () {
            controller
              .getBottomSheetView(); // Memanggil fungsi getBottomSheetView
          },
          tooltip: 'Show Bottom Sheet',
          child: const Icon(Icons.arrow_upward), // Ikon panah atas
        ),
      ],
    ),
  );
}

```

**Sourcecode view\_model/counter\_controller.dart:**

```

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

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

  void incrementCounter() {
    counter++;
    getSnackBar('Counter Updated', 'Counter increased to $counter');
  }

  void decrementCounter() {
    counter--;
    getSnackBar('Counter Updated', 'Counter decreased to $counter');
  }

  void getSnackBar(String title, String message) {
    Get.snackbar(
      title,
      message,
      snackPosition: SnackPosition.TOP,
      backgroundColor: Colors.green,
      colorText: Colors.white,
    );
  }

  void getDialog() {
    Get.defaultDialog(
      title: 'Dialog Title',
      middleText: 'This is a dialog message.',
      backgroundColor: Colors.white,
      titleStyle: TextStyle(color: Colors.black),
      middleTextStyle: TextStyle(color: Colors.black),
      actions: [
        TextButton(
          onPressed: () {
            Get.back(); // Menutup dialog
          },
          child: Text('Close', style: TextStyle(color: Colors.blue)),
        ),
      ],
    );
  }

  void getBottomSheetView() {
    Get.bottomSheet(
      Card(
        elevation: 4,
        child: Container(
          padding: EdgeInsets.all(20),
          width: double.infinity, // Mengisi lebar bottom sheet
          child: Column(
            mainAxisAlignment: MainAxisAlignment.min,
            children: [
              Text(
                'Bottom Sheet Title',
                style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
              ),
              SizedBox(height: 10),
              Text('This is a bottom sheet message.'),
              SizedBox(height: 20),
              ElevatedButton(
                onPressed: () {
                  Get.back(); // Menutup bottom sheet
                },
                child: Text('Close'),
              ),
            ],
          ),
        ),
      ),
      isDismissible: true,
    );
  }
}

// Fungsi untuk mendapatkan tampilan Sidebar Menu

```

```

Widget getSidebar() {
  return Drawer(
    child: ListView(
      padding: EdgeInsets.zero,
      children: [
        DrawerHeader(
          decoration: BoxDecoration(
            color: Colors.blue,
          ),
          child: Text(
            'Menu',
            style: TextStyle(
              color: Colors.white,
              fontSize: 24,
            ),
          ),
        ),
        ListTile(
          title: Text('Home'),
          onTap: () {
            Get.back(); // Menutup drawer
          },
        ),
        ListTile(
          title: Text('Detail'),
          onTap: () {
            Get.toNamed('/details'); // Navigasi ke DetailPage
          },
        ),
        ListTile(
          title: Text('Settings'),
          onTap: () {
            Get.back(); // Menutup drawer
            // Tambahkan logika untuk navigasi ke pengaturan jika diperlukan
          },
        ),
      ],
    ),
  );
}

```

### Sourcecode main.dart:

```

import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:mdl2/view/my_home_page.dart';
import 'package:mdl2/view/detail_page.dart'; // Import DetailPage

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

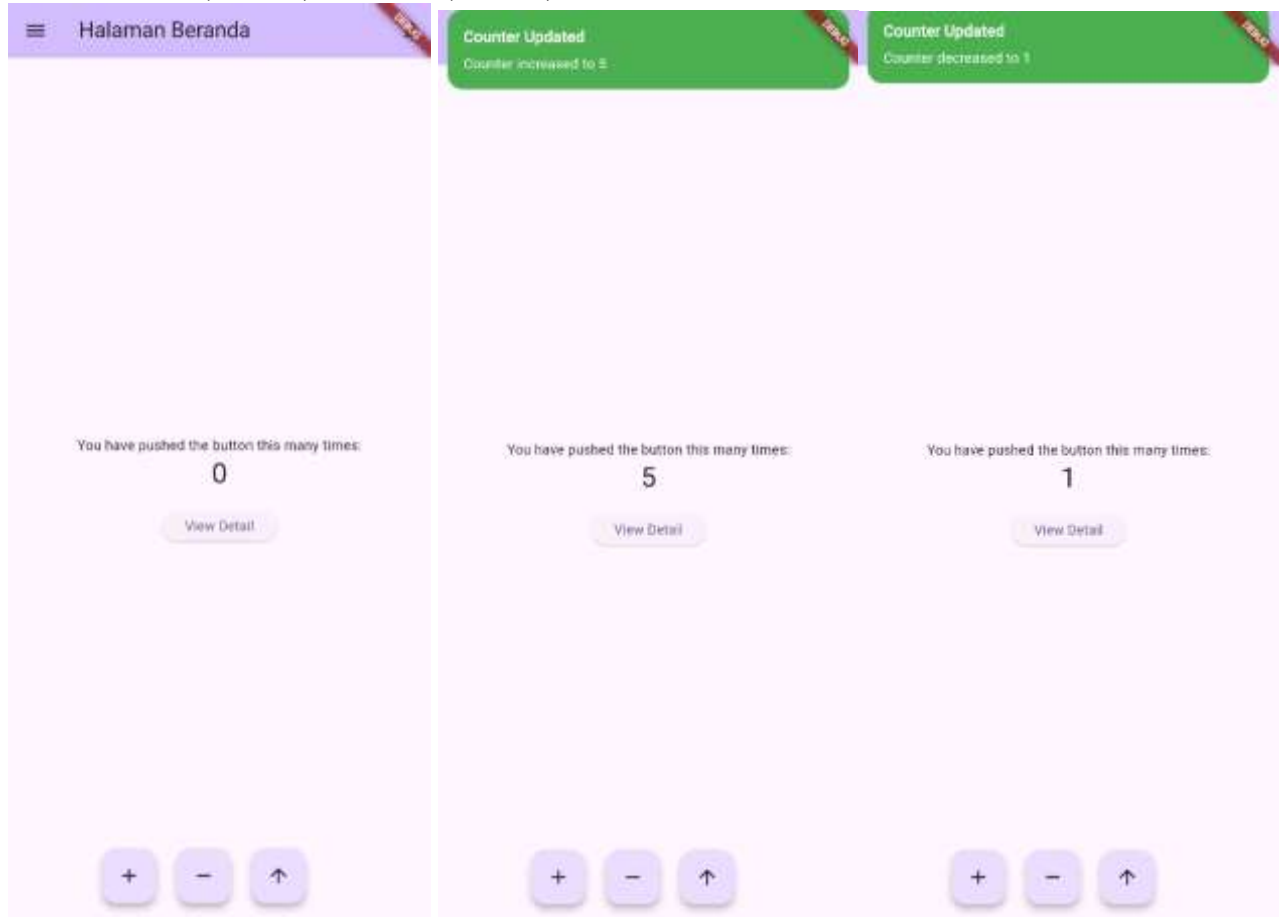
class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      title: 'Guided 13 - Networking',
      initialRoute: '/',
      getPages: [
        GetPage(name: '/', page: () => MyHomePage(title: 'Halaman Beranda')),
        GetPage(name: '/details', page: () => const DetailPage()),
      ],
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
    );
  }
}

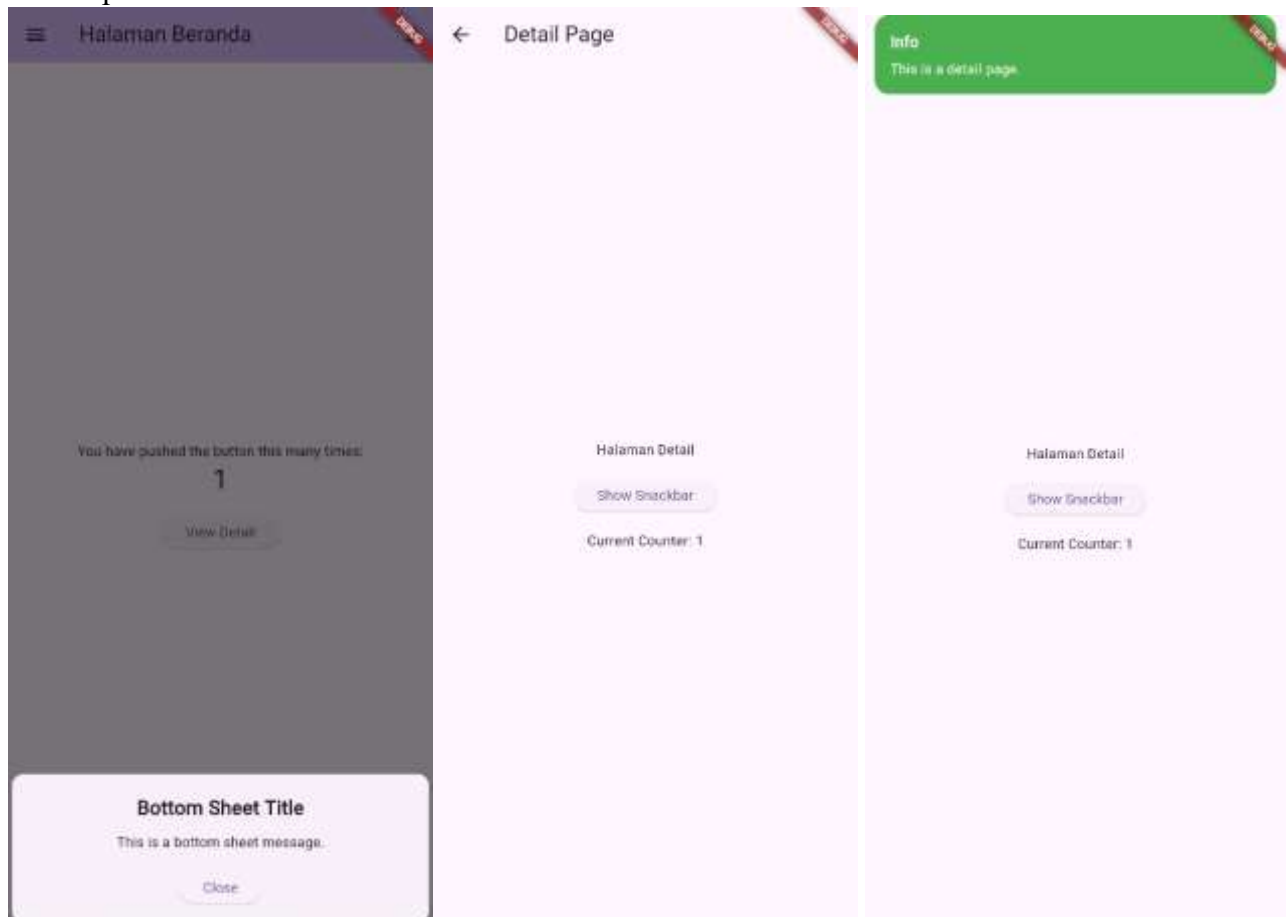
```

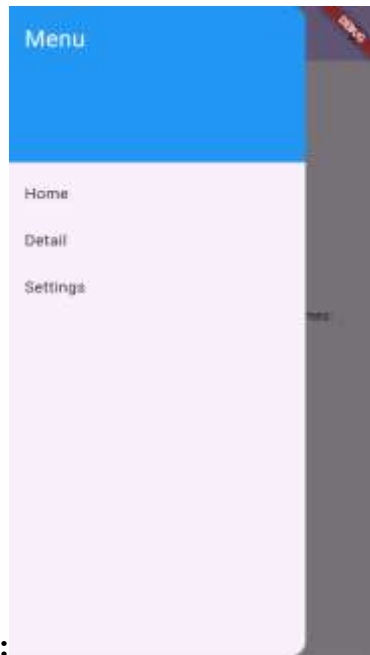
## Screenshoot Output

Start > click + (5 times) > click - (4 times)



Click up icon > click "view detail" > click "show snackbar"





**Side bar:**

### **Deskripsi Program**

Program yang dibuat adalah aplikasi penghitung (counter) sederhana yang bisa menambah dan mengurangi angka. Program ini menggunakan GetX untuk mengatur data (state management), sehingga nilai counter bisa diakses dan diubah dari berbagai halaman. Di halaman utama, terdapat tiga tombol: tombol tambah (+) untuk menambah angka, tombol kurang (-) untuk mengurangi angka, dan tombol panah ke atas untuk menampilkan pesan dari bawah layar. Setiap kali angka berubah, akan muncul notifikasi di bagian atas layar.

Aplikasi ini memiliki dua halaman yang bisa diakses melalui tombol "View Detail" atau melalui menu samping (sidebar). Halaman utama menampilkan angka counter dan tombol-tombol untuk mengubahnya, sedangkan halaman detail menampilkan nilai counter yang sama karena menggunakan data yang sama. Di bagian atas aplikasi terdapat tombol notifikasi yang bisa menampilkan pesan dialog, dan menu samping yang bisa dibuka untuk navigasi antar halaman. Semua fitur ini dikelola menggunakan GetX, yang membuat aplikasi lebih terstruktur dan mudah dikembangkan



## B. PRAKTIKUM (Tugas Mandiri)

### Soal Studi Case

#### 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.
- 

#### Sourcecode controllers/note\_controller.dart:

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

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

  void addNote(String title, String description) {
    notes.add(Note(title: title, description: description));
    Get.back();

    Get.snackbar(
      'Berhasil', // title
      'Catatan "$title" telah ditambahkan',
      backgroundColor: Colors.green,
      colorText: Colors.white,
      snackPosition: SnackPosition.TOP,
      duration: Duration(seconds: 4),
    );
  }

  void deleteNote(int index) {
    final deletedNote = notes[index];
    notes.removeAt(index);

    Get.snackbar(
      'Berhasil',
      'Catatan "${deletedNote.title}" telah dihapus',
      backgroundColor: Colors.red,
      colorText: Colors.white,
      snackPosition: SnackPosition.TOP,
      duration: Duration(seconds: 4),
    );
  }
}
```

#### Sourcecode models/note\_model.dart:

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

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

#### Sourcecode views/home\_page.dart:

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

class HomePage extends StatelessWidget {
  final NoteController controller = Get.put(NoteController());
```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Catatan Saya'),
    ),
    body: Obx(
      () => ListView.builder(
        itemCount: controller.notes.length,
        itemBuilder: (context, index) {
          final note = controller.notes[index];
          return Card(
            margin: EdgeInsets.all(8),
            child: ListTile(
              title: Text(note.title),
              subtitle: Text(note.description),
              trailing: IconButton(
                icon: Icon(Icons.delete),
                onPressed: () => controller.deleteNote(index),
              ),
            ),
          );
        },
      ),
    ),
    floatingActionButton: FloatingActionButton(
      onPressed: () => Get.toNamed('/add'),
      child: Icon(Icons.add),
    ),
  );
}

```

### Sourcecode views/add\_note\_page.dart:

```

import 'package:flutter/material.dart';
import 'package:get/get.dart';
import '../controllers/note_controller.dart';

class AddNotePage extends StatelessWidget {
  final NoteController controller = Get.find();
  final TextEditingController titleController = TextEditingController();
  final TextEditingController descriptionController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Tambah Catatan'),
      ),
      body: Padding(
        padding: EdgeInsets.all(16),
        child: Column(
          children: [
            TextField(
              controller: titleController,
              decoration: InputDecoration(
                labelText: 'Judul',
                border: OutlineInputBorder(),
              ),
            ),
            SizedBox(height: 16),
            TextField(
              controller: descriptionController,
              decoration: InputDecoration(
                labelText: 'Deskripsi',
                border: OutlineInputBorder(),
              ),
              maxLines: 3,
            ),
            SizedBox(height: 16),
            ElevatedButton(
              onPressed: () {
                if (titleController.text.isNotEmpty &&
                    descriptionController.text.isNotEmpty) {
                  controller.addNote(
                    titleController.text,
                    descriptionController.text,
                  );
                } else {

```

```

        Get.snackbar(
          'Error',
          'Judul dan deskripsi harus diisi',
          backgroundColor: Colors.orange,
          colorText: Colors.white,
          snackPosition: SnackPosition.TOP,
          duration: Duration(seconds: 2),
        );
      },
    ),
    child: Text('Simpan'),
  ),
),
),
);
}
}

```

### Sourcecode main.dart:

```

import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'views/home_page.dart';
import 'views/add_note_page.dart';

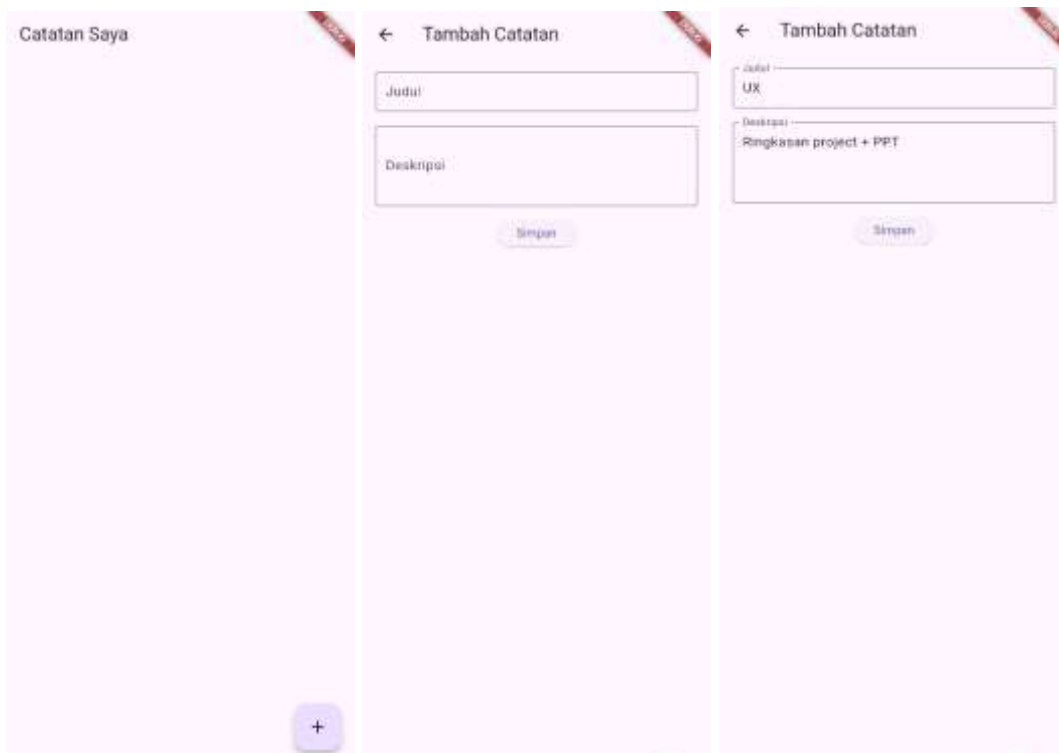
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      title: 'Aplikasi Catatan',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      initialRoute: '/',
      getPages: [
        GetPage(name: '/', page: () => HomePage()),
        GetPage(name: '/add', page: () => AddNotePage()),
      ],
    );
  }
}

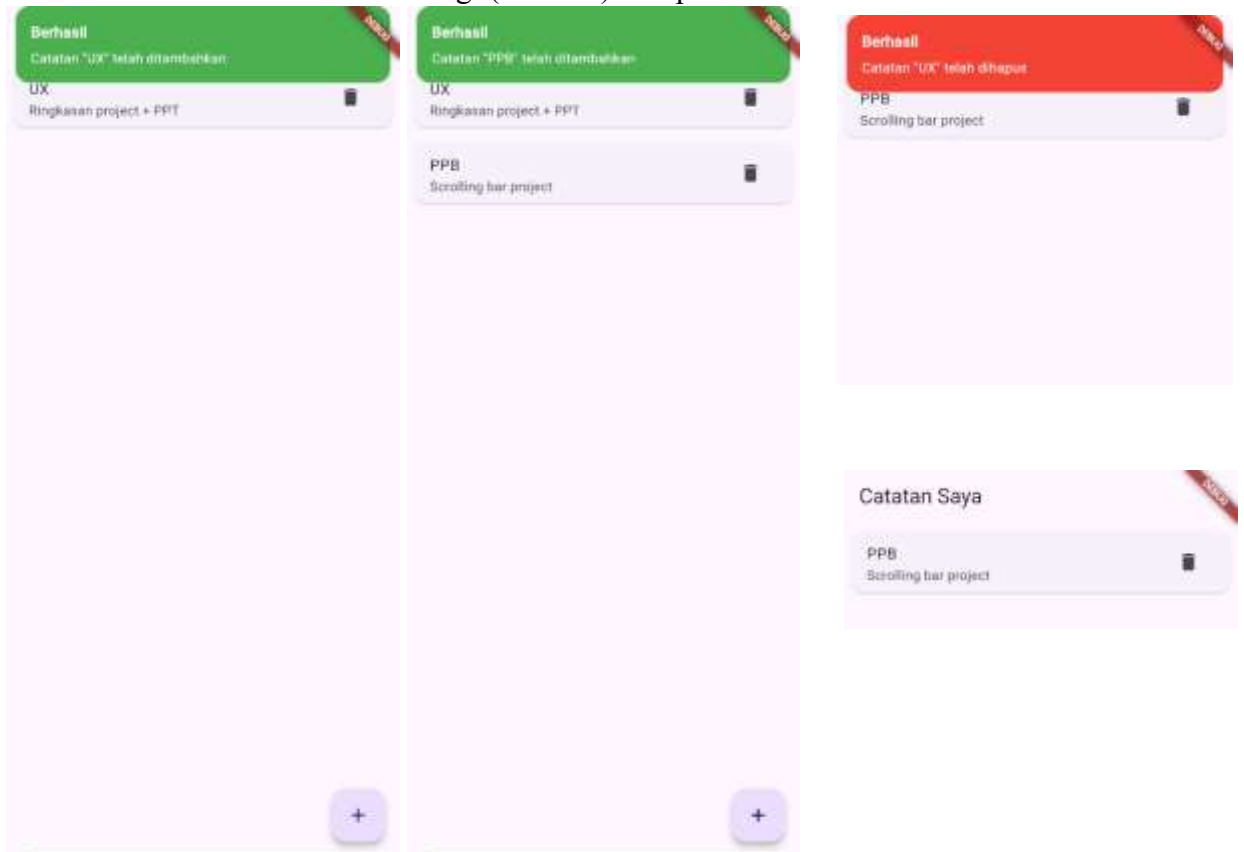
```

### Screenshoot Output

Start > klik + > isi data catatan



Snackbar berhasil > tambah data lagi (berhasil) > hapus catatan "UX" > end



## Deskripsi Program

Program ini adalah aplikasi catatan sederhana yang memungkinkan pengguna untuk menambah, melihat, dan menghapus catatan. Program ini menggunakan GetX untuk manajemen state, di mana daftar catatan disimpan dalam controller sebagai observable list (`<Note>[].obs`).

Terdiri dari dua halaman utama:

1. HomePage untuk menampilkan daftar catatan
2. AddNotePage untuk menambah catatan baru.

Setiap catatan memiliki judul dan deskripsi yang ditampilkan dalam bentuk Card dengan tombol hapus.

Navigasi antar halaman menggunakan GetX routing system dengan dua rute utama: '/' untuk HomePage dan '/add' untuk AddNotePage.

Di halaman utama, tekan tombol + untuk menambah catatan baru, catatan berisi form judul dan deskripsi, lalu tekan tombol Simpan. Saat data berhasil disimpan muncul snackbar berwarna hijau dengan deskripsi judul yang disimpan, dan catatan baru akan muncul di halaman utama. Untuk menghapus catatan, tekan icon tempat sampah di sebelah kanan catatan, saat catatan berhasil dihapus, muncul juga snackbar berwarna merah.

## **BAB III**

### **KESIMPULAN DAN SARAN**

#### **A. KESIMPULAN**

GetX menyediakan solusi state management yang efisien dan mudah diimplementasikan, pemisahan logic menggunakan controller membantu dalam maintenance code, adapun fitur routing GetX mempermudah navigasi antar halaman. Reactive state management memastikan UI selalu sinkron dengan data.

#### **B. REFERENSI**

Flutter official documentation ([flutter.dev](https://flutter.dev))  
GetX documentation ([pub.dev/packages/get](https://pub.dev/packages/get))