

**LAPORAN PRAKTIKUM  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL 13  
NETWORKING**



**Disusun Oleh :**

**Ade Fatkhul Anam / 2211104051**

**SE-06-02**

**Asisten Praktikum :**

**Muhammad Faza Zulian Gesit Al Barru**

**Aisyah Hasna Aulia**

**Dosen Pengampu :**

**Yudha Islami Sulistya**

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

**2024**

## GUIDED

### A. LOGIC

Sebelum kita pergi ke tampilan flutter project, Langkah yang pertama kita lakukan adalah mengatur file- file services untuk google maps.

#### 1. PUBSPEC.YAML

Sourcecode

```
name: pertemuan133
description: "A new Flutter project."
# The following line prevents the package from being accidentally
published to
# pub.dev using `flutter pub publish`. This is preferred for private
packages.
publish_to: 'none' # Remove this line if you wish to publish to pub.dev

# The following defines the version and build number for your application.
# A version number is three numbers separated by dots, like 1.2.43
# followed by an optional build number separated by a +.
# Both the version and the builder number may be overridden in flutter
# build by specifying --build-name and --build-number, respectively.
# In Android, build-name is used as versionName while build-number used as
versionCode.
# Read more about Android versioning at
https://developer.android.com/studio/publish/versioning
# In iOS, build-name is used as CFBundleShortVersionString while build-
number is used as CFBundleVersion.
# Read more about iOS versioning at
#
https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html
# In Windows, build-name is used as the major, minor, and patch parts
# of the product and file versions while build-number is used as the build
suffix.
version: 1.0.0+1

environment:
  sdk: ^3.5.4

# Dependencies specify other packages that your package needs in order to
work.
# To automatically upgrade your package dependencies to the latest
versions
# consider running `flutter pub upgrade --major-versions`. Alternatively,
# dependencies can be manually updated by changing the version numbers
below to
# the latest version available on pub.dev. To see which dependencies have
newer
# versions available, run `flutter pub outdated`.
dependencies:
  flutter:
    sdk: flutter
```

```
# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.
cupertino_icons: ^1.0.8
get: ^4.6.6

dev_dependencies:
  flutter_test:
    sdk: flutter

# The "flutter_lints" package below contains a set of recommended lints
to
# encourage good coding practices. The lint set provided by the package
is
# activated in the `analysis_options.yaml` file located at the root of
your
# package. See that file for information about deactivating specific
lint
# rules and activating additional ones.
flutter_lints: ^4.0.0

# For information on the generic Dart part of this file, see the
# following page: https://dart.dev/tools/pub/pubspec

# The following section is specific to Flutter packages.
flutter:

# The following line ensures that the Material Icons font is
# included with your application, so that you can use the icons in
# the material Icons class.
uses-material-design: true

# To add assets to your application, add an assets section, like this:
# assets:
#   - images/a_dot_burr.jpeg
#   - images/a_dot_ham.jpeg

# An image asset can refer to one or more resolution-specific
"variants", see
# https://flutter.dev/to/resolution-aware-images

# For details regarding adding assets from package dependencies, see
# https://flutter.dev/to/asset-from-package

# To add custom fonts to your application, add a fonts section here,
# in this "flutter" section. Each entry in this list should have a
# "family" key with the font family name, and a "fonts" key with a
# list giving the asset and other descriptors for the font. For
# example:
# fonts:
#   - family: Schyler
#     fonts:
#       - asset: fonts/Schyler-Regular.ttf
```

```
# - asset: fonts/Schyler-Italic.ttf
#   style: italic
# - family: Trajan Pro
#   fonts:
#     - asset: fonts/TrajanPro.ttf
#     - asset: fonts/TrajanPro_Bold.ttf
#       weight: 700
#
# For details regarding fonts from package dependencies,
# see https://flutter.dev/to/font-from-package
```

## Deskripsi Program

File pubspec.yaml ini menggunakan Get versi 4.6.6, sebuah library Flutter yang populer untuk state management, routing, dan dependency injection. Dengan Get, pengembang dapat mengatur UI dan logika aplikasi lebih efisien tanpa perlu menulis banyak kode boilerplate. Fitur unggulannya meliputi pengaturan controller, pembaruan state secara reaktif menggunakan GetBuilder atau Obx, serta navigasi halaman yang lebih sederhana dengan Get.to(), Get.off(), atau Get.back(). Selain itu, Get menyediakan manajemen dependensi menggunakan Get.put() atau Get.lazyPut(), menjadikannya solusi yang ringan dan fleksibel untuk pengembangan aplikasi Flutter.

## B. TAMPILAN APLIKASI

Setelah kita melakukan konfigurasi untuk Firebase Notification, tidak lupa kita melakukan coding untuk tampilan aplikasinya

### 1. VIEW\_MODEL/COUNTER\_CONTROLLER

#### Sourcecode

```
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'),
            ),
          ],
        ),
      ),
    ),
  );
}

```

```

        ],
      ),
    ),
  ),
  dismissible: 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
          },
        ),
      ],
    ),
  );
}
}

```

## Deskripsi Program

Kode di atas mendefinisikan CounterController menggunakan GetX sebagai controller untuk mengelola state aplikasi Flutter secara reaktif. Variabel counter dibuat sebagai observable dengan .obs, sehingga memungkinkan UI diperbarui secara otomatis ketika nilainya berubah. Metode incrementCounter dan decrementCounter berfungsi untuk menambah atau mengurangi nilai counter serta menampilkan notifikasi dengan Get.snackbar. Selain itu, controller ini menyediakan metode getDialog untuk menampilkan dialog sederhana menggunakan Get.defaultDialog, metode getBottomSheetView untuk menampilkan bottom sheet dengan Get.bottomSheet, serta getSidebar yang mengembalikan widget Drawer berisi menu navigasi. Navigasi antar halaman dilakukan dengan bantuan Get.toNamed. Kode ini memanfaatkan berbagai fitur GetX, seperti state management, snackbar, dialog, bottom sheet, dan navigasi, untuk membangun aplikasi Flutter yang lebih efisien, terstruktur, dan mudah dikelola.

### a. VIEW/MY\_HOME\_PAGE.DART

#### Sourcecode

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:pertemuan133/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
        ),
    ],
),
),
);
}
}

```



## Deskripsi Program

Kode di atas mendefinisikan halaman utama MyHomePage dalam aplikasi Flutter dengan menggunakan GetX untuk manajemen state dan navigasi. Halaman ini menampilkan nilai counter yang dikelola oleh CounterController, di mana perubahan nilainya diperbarui secara reaktif menggunakan Obx. Terdapat Floating Action Button untuk melakukan increment, decrement, dan menampilkan bottom sheet melalui metode yang terdapat pada controller. Selain itu, halaman ini memiliki AppBar dengan tombol notifikasi yang memunculkan dialog menggunakan Get.defaultDialog dan Drawer (sidebar menu) yang diambil dari controller untuk navigasi ke halaman lain. Pada bagian body, terdapat tombol "View Detail" yang menggunakan Get.toNamed untuk berpindah ke halaman detail. Program ini menunjukkan pemanfaatan GetX untuk state management, dialog, bottom sheet, navigasi, serta integrasi UI yang rapi dan efisien.

## b. VIEW/DETAIL\_PAGE.DART

### Sourcecode

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:pertemuan133/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
```

```

    ],
  ),
),
);
}
}

```

## Deskripsi Program

Kode di atas mendefinisikan `DetailPage`, sebuah halaman sederhana dalam aplikasi Flutter yang menggunakan `GetX` untuk mengambil instance `CounterController` melalui `Get.find()`. Halaman ini menampilkan teks "Halaman Detail" dan tombol yang, ketika ditekan, akan menampilkan `SnackBar` menggunakan `Get.snackbar` dengan pesan "This is a detail page" di bagian atas layar.

Selain itu, nilai counter yang dikelola oleh `CounterController` ditampilkan langsung di halaman ini. Hal ini menunjukkan bagaimana `GetX` digunakan untuk berbagi state antar halaman dan menyederhanakan UI serta interaksi, seperti menampilkan `snackbar` dan mengakses state melalui controller.

## c. MAIN.DART

### Sourcecode

```

import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:pertemuan133/view/my_home_page.dart';
import 'package:pertemuan133/view/detail_page.dart'; // Import DetailPage

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

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

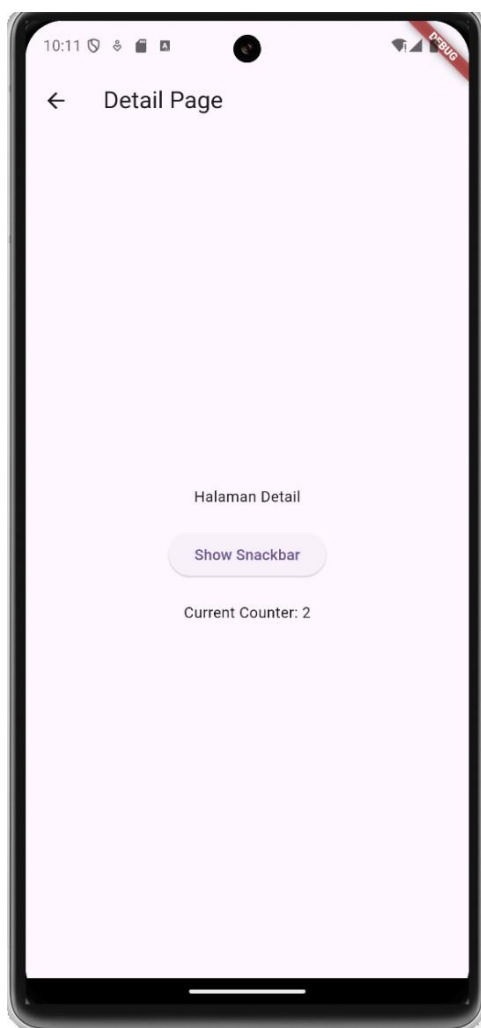
  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      title: 'Guided 13 - Networking',
      initialRoute: '/',
      getPages: [
        GetPage(name: '/', page: () => MyHomePage(title: 'Halaman Beranda')),
        GetPage(name: '/details', page: () => const DetailPage()),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

```

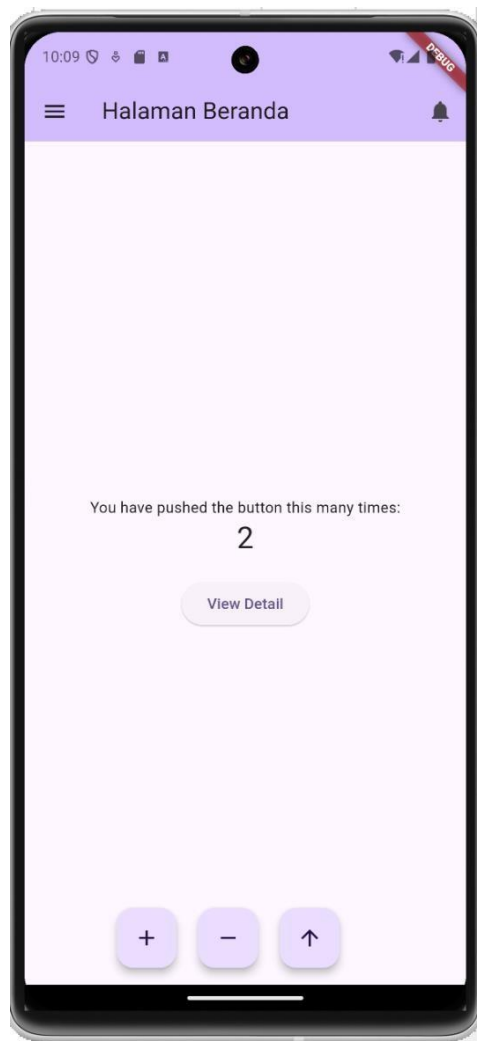
## Deskripsi Program

Kode di atas mendefinisikan aplikasi Flutter yang menggunakan GetX dengan GetMaterialApp sebagai root widget, memungkinkan navigasi yang lebih sederhana dan manajemen halaman yang efisien. Aplikasi ini memiliki initialRoute di '/' yang menampilkan halaman utama MyHomePage, serta route '/details' yang menampilkan DetailPage. Navigasi diatur menggunakan getPages untuk mendeklarasikan rute-rute aplikasi, sehingga perpindahan halaman dapat dilakukan dengan Get.toNamed(). Selain itu, aplikasi ini menerapkan Material 3 dengan tema warna utama deep purple menggunakan ColorScheme.fromSeed. Struktur ini memanfaatkan fitur GetX untuk menyederhanakan navigasi tanpa deklarasi kompleks, sehingga kode menjadi lebih clean, terstruktur, dan mudah dipelihara.

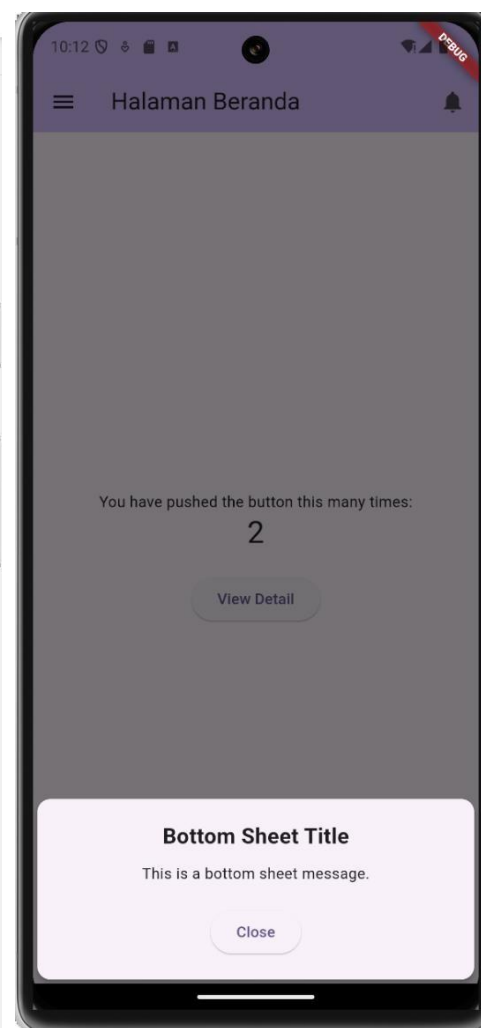
## Screenshoot Output



Detail Page



Home Page



Bottom Sheet Title

## **Deskripsi program**

Program ini adalah aplikasi Flutter yang menggunakan GetX untuk state management dan navigasi antar halaman, yang terdiri dari Halaman Beranda (Home Page), Detail Page, serta komponen tambahan seperti Snackbar, Dialog, dan Bottom Sheet. Pada Home Page, pengguna dapat melihat nilai counter yang berubah secara reaktif menggunakan Obx dan dapat meningkatkan atau menurunkan nilainya melalui tombol "+" atau "-". Selain itu, terdapat tombol navigasi "View Detail" yang mengarahkan ke Detail Page, di mana nilai counter ditampilkan bersamaan dengan tombol untuk memunculkan Snackbar sebagai notifikasi. Tombol dengan ikon panah pada Home Page digunakan untuk menampilkan Bottom Sheet dengan informasi tambahan menggunakan Get.bottomSheet. Aplikasi ini memanfaatkan fitur GetX seperti Get.snackbar untuk notifikasi, Get.bottomSheet untuk menampilkan bottom sheet, dan navigasi dinamis menggunakan Get.toNamed. Dengan pendekatan ini, pengelolaan state dan navigasi menjadi lebih efisien, bersih, dan mudah dipelihara.

## **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.

Berikut merupakan sourcecode dari program

## A. VIEW/ADD\_NOTE\_VIEW.DART

### Sourcecode

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

class AddNoteView extends StatelessWidget {
  final NotesViewModel viewModel = Get.find<NotesViewModel>();

  final TextEditingController titleController = TextEditingController();
  final TextEditingController descriptionController =
TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blueGrey[50], // Warna latar belakang
      appBar: AppBar(
        title: Text(
          'Tambah Catatan',
          style: TextStyle(fontWeight: FontWeight.bold, color:
Colors.white),
        ),
        backgroundColor: Colors.deepPurple, // Warna AppBar
        centerTitle: true,
      ),
      body: Padding(
```

```

padding: const EdgeInsets.all(16.0),
child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    Text(
      "Judul Catatan",
      style: TextStyle(
        fontSize: 18,
        fontWeight: FontWeight.w500,
        color: Colors.deepPurple,
      ),
    ),
    SizedBox(height: 8),
    TextField(
      controller: titleController,
      decoration: InputDecoration(
        hintText: 'Masukkan judul catatan',
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(12),
        ),
        filled: true,
        fillColor: Colors.white,
      ),
    ),
    SizedBox(height: 16),
    Text(
      "Deskripsi",
      style: TextStyle(
        fontSize: 18,
        fontWeight: FontWeight.w500,
        color: Colors.deepPurple,
      ),
    ),
    SizedBox(height: 8),
    TextField(
      controller: descriptionController,
      maxLines: 4,
      decoration: InputDecoration(
        hintText: 'Masukkan deskripsi catatan',
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(12),
        ),
        filled: true,
        fillColor: Colors.white,
      ),
    ),
    SizedBox(height: 20),
    Center(
      child: ElevatedButton(
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.deepPurple,
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(12),
          ),
        ),
      ),
    ),
  ],
),

```

```

padding: EdgeInsets.symmetric(horizontal: 24, vertical:
12),
),
onPressed: () {
String title = titleController.text.trim();
String description = descriptionController.text.trim();

if (title.isNotEmpty && description.isNotEmpty) {
viewModel.addNote(title, description);
Get.back();
} else {
Get.snackbar('Error', 'Judul dan Deskripsi harus
diisi',
backgroundColor: Colors.redAccent,
colorText: Colors.white);
}
},
child: Text(
'Simpan Catatan',
style: TextStyle(
fontSize: 16,
fontWeight: FontWeight.bold,
color: Colors.white),
),
),
),
],
),
),
);
}
}

```

## B. VIEW/HOME\_VIEW.DART

### Sourcecode

```

import 'package:flutter/material.dart';
import 'package:get/get.dart';
import '../view_model/notes_view_model.dart';
import '../view_model/app_routes.dart';

class HomeView extends StatelessWidget {
final NotesViewModel viewModel = Get.put(NotesViewModel());

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.blueGrey[50], // Warna latar belakang
appBar: AppBar(
title: Text(
'Catatan Sederhana',

```

```

        style: TextStyle(
          fontWeight: FontWeight.bold,
          fontSize: 22,
          color: Colors.white,
        ),
      ),
      backgroundColor: Colors.deepPurple, // Warna AppBar
      centerTitle: true,
    ),
    body: Obx(
      () {
        if (viewModel.notes.isEmpty) {
          return Center(
            child: Text(
              'Belum ada catatan.',
              style: TextStyle(
                color: Colors.black54,
                fontSize: 18,
              ),
            ),
          );
        }
        return ListView.builder(
          padding: EdgeInsets.all(10),
          itemCount: viewModel.notes.length,
          itemBuilder: (context, index) {
            final note = viewModel.notes[index];
            return Card(
              elevation: 4,
              margin: EdgeInsets.symmetric(vertical: 8, horizontal: 5),
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(15),
              ),
              child: ListTile(
                contentPadding: EdgeInsets.all(15),
                title: Text(
                  note['title'] ?? '',
                  style: TextStyle(
                    fontSize: 20,
                    fontWeight: FontWeight.w600,
                    color: Colors.deepPurple,
                  ),
                ),
                subtitle: Text(
                  note['description'] ?? '',
                  style: TextStyle(fontSize: 16, color: Colors.black87),
                ),
                trailing: IconButton(
                  icon: Icon(Icons.delete, color: Colors.redAccent,
size: 28),
                  onPressed: () => viewModel.deleteNote(index),
                ),
              ),
            );
          }
        );
      }
    );
  }
}

```



```

        },
      );
    },
  ),
  floatingActionButton: FloatingActionButton(
    onPressed: () => Get.toNamed(AppRoutes.ADD_NOTE),
    backgroundColor: Colors.deepPurple,
    child: Icon(Icons.add, color: Colors.white),
  ),
);
}
}

```

## C. VIEW\_MODEL/APP\_ROUTES.DART

### Sourcecode

```

import 'package:get/get.dart';
import '../view/home_view.dart';
import '../view/add_note_view.dart';

class AppRoutes {
  static const HOME = '/';
  static const ADD_NOTE = '/add-note';

  static final pages = [
    GetPage(name: HOME, page: () => HomeView()),
    GetPage(name: ADD_NOTE, page: () => AddNoteView()),
  ];
}

```

## D. VIEW\_MODEL/NOTES\_VIEW\_MODEL.DART

### Sourcecode

```

import 'package:get/get.dart';

class NotesViewModel extends GetxController {
  var notes = <Map<String, String>>[].obs;

  void addNote(String title, String description) {
    notes.add({"title": title, "description": description});
  }

  void deleteNote(int index) {
    notes.removeAt(index);
  }
}

```

## E. MAIN.DART

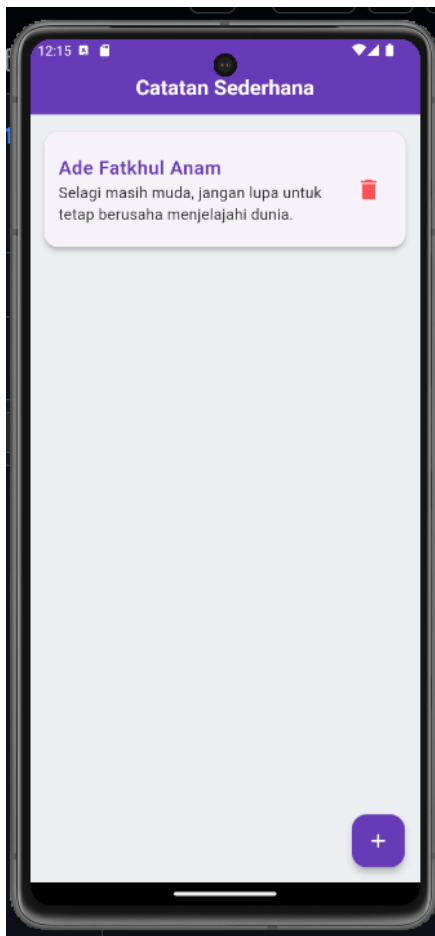
### Sourcecode

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Catatan Sederhana',
      debugShowCheckedModeBanner: false,
      initialRoute: AppRoutes.HOME,
      getPages: AppRoutes.pages,
    );
  }
}
```

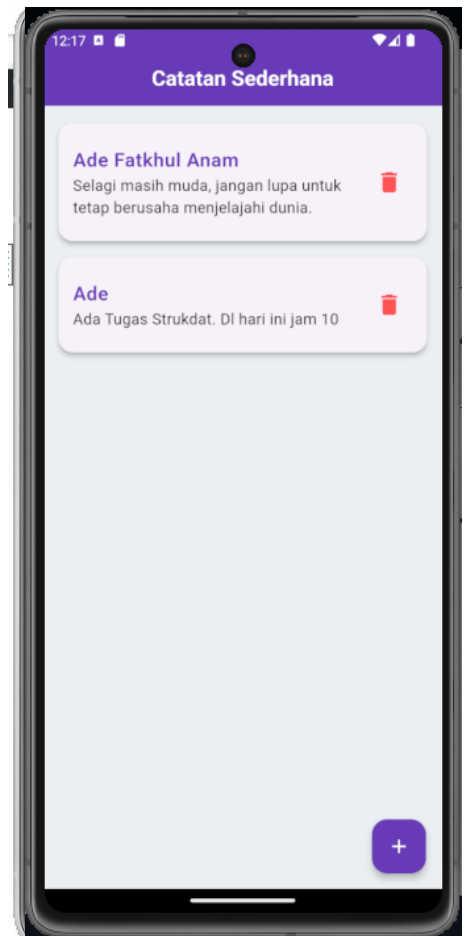
### Screenshoot output



Home Page



Add Note Page



New Note added successfully

## **Deskripsi Program**

Program ini adalah aplikasi Catatan Sederhana berbasis Flutter yang memungkinkan pengguna untuk menambahkan dan mengelola catatan. Pada Home Page, daftar catatan yang tersimpan ditampilkan, di mana setiap catatan memiliki judul, deskripsi, dan tombol untuk menghapus catatan. Di pojok kanan bawah, terdapat tombol "Tambah (+)" yang membawa pengguna ke halaman Tambah Catatan. Di halaman Tambah Catatan, pengguna dapat memasukkan judul dan deskripsi catatan, lalu menyimpannya dengan menekan tombol "Simpan Catatan". Setelah catatan berhasil disimpan, pengguna akan diarahkan kembali ke Home Page, dan catatan baru akan muncul di daftar yang sudah ada. Aplikasi ini memiliki antarmuka yang intuitif dan mudah digunakan, serta menyederhanakan pengelolaan data catatan. Kemungkinan aplikasi ini menggunakan state management seperti GetX atau Provider untuk menyimpan, memperbarui, dan menghapus catatan secara real-time, sehingga memastikan pengalaman pengguna yang lebih efisien dan responsif.