LAPORAN PRAKTIKUM
Modul 13
“Networking”

Disusun Oleh:
Ganesha Rahman Gibran -2211104058
Kelas S1SE-06-02

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

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO

2024

Tujuan

1. Mahasiswa mampu memahami state management dalam Flutter
2. Mahasiswa mampu mengimplementasikan state management dalam Flutter

Landasan Teori

State Management dalam Flutter

State management adalah proses mengelola state atau status aplikasi, yaitu data atau informasi yang dapat berubah selama siklus hidup aplikasi. Dalam Flutter, state sangat memengaruhi tampilan antarmuka pengguna (UI), mulai dari input pengguna hingga data yang diperoleh dari API. Pengelolaan state yang baik diperlukan untuk memastikan sinkronisasi antara data dan UI, pengorganisasian kode yang lebih baik, dan pengurangan bug pada aplikasi.

Flutter memiliki pendekatan deklaratif, di mana UI dibangun berdasarkan state saat ini. Dengan mengimplementasikan state management, aliran data dapat dikendalikan lintas aplikasi, sehingga mempermudah pengembangan aplikasi yang kompleks.

Jenis State dalam Flutter

1. **Ephemeral State (State Lokal)**
    - State yang hanya relevan untuk widget tertentu dan tidak dibagikan ke widget lain.
    - Contoh: State pada TextField atau Checkbox.
    - Biasanya dikelola menggunakan StatefulWidget atau InheritedWidget.
2. **App State (State Global)**
    - State yang digunakan oleh berbagai widget dalam aplikasi.
    - Contoh: Informasi pengguna, data keranjang belanja, atau tema aplikasi.
    - Membutuhkan pendekatan state management yang lebih kompleks, seperti menggunakan library atau framework tertentu.

Pendekatan State Management dalam Flutter

Beberapa framework atau library yang umum digunakan untuk state management:

1. **Provider**
    - Library yang didukung resmi oleh tim Flutter.
    - Memanfaatkan InheritedWidget dengan cara yang lebih sederhana dan efisien.
2. **BloC (Business Logic Component) / Cubit**
    - Pendekatan berbasis pola stream untuk memisahkan business logic dari UI.
    - Cocok untuk aplikasi besar dan kompleks.
3. **Riverpod**
    - Framework modern yang dirancang untuk menggantikan atau melengkapi Provider.
    - Mengatasi beberapa keterbatasan Provider dengan fleksibilitas yang lebih tinggi.
4. **GetX**
    - Framework serbaguna untuk state management, routing, dan dependency injection.

- Meminimalkan kode boilerplate, meningkatkan efisiensi, dan menyediakan solusi lengkap untuk aplikasi yang memerlukan reaktivitas tinggi.

Penerapan GetX dalam Flutter

1. **Instalasi**
   - Tambahkan get pada file pubspec.yaml untuk mengintegrasikan GetX ke dalam proyek Flutter.
2. **State Management dengan GetX**
   - Buat controller untuk mengelola state, menggunakan variabel reaktif dengan .obs.
   - Pantau perubahan state dengan widget Obx.
3. **Routing dengan GetX**
   - Definisikan rute menggunakan GetPage.
   - Navigasi dapat dilakukan dengan metode seperti Get.to() atau Get.toNamed().
4. **Dependency Injection**
   - Menggunakan Get.put() untuk menyediakan instance yang dapat diakses di mana saja.
   - Get.lazyPut() digunakan untuk lazy loading instance.
5. **Fitur Tambahan**
   - Snackbar: Get.snackbar() untuk menampilkan pesan pop-up.
   - Dialog: Get.defaultDialog() untuk menampilkan dialog.
   - Bottom Sheet: Get.bottomSheet() untuk menampilkan bottom sheet.

Guided

Input :

- main.dart

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:guided13/view%20model/counter_controller.dart';
import 'package:guided13/view/detail_page.dart';
import 'package:guided13/view/my_home_page.dart';

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

class MyApp extends StatelessWidget {
  MyApp({super.key});
  final CounterController controller = Get.put(CounterController());

  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      initialRoute: '/',
      getPages: [
        GetPage(
```

```
          name:'/',
          page: ()=> MyHomePage(title: 'Belajar GetX'),
        ),
        GetPage(
          name:'/detail',
          page: ()=> DetailPage(),
        ),
      ],
    );
  }
}
```

- view/detail_page.dart

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

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'Halaman Detail'
        ),
      ),
      body: const Center(
        child: Text(
          'Halaman Detail'
          ),
      ),
    );
  }
}
```

- view/my_home_page.dart

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:guided13/view%20model/counter_controller.dart';

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

  final String title;
  final controller = Get.find<CounterController>();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: Text(title),
      ),
```

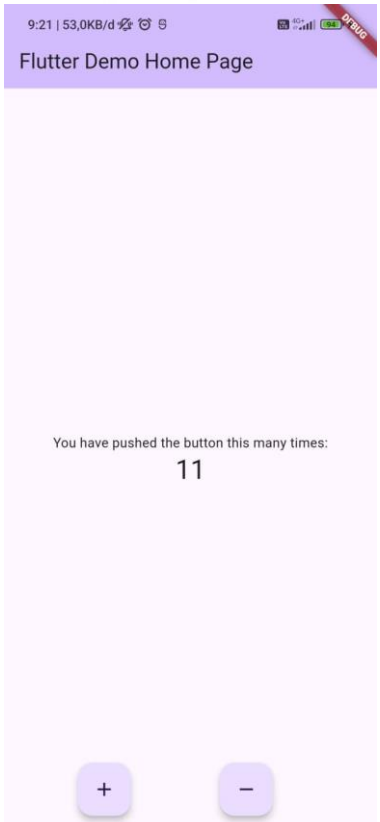
```
        body: Center(
            child: Obx(
          () => Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              const Text(
                'You have pushed the button this many times:',
              ),
              Text(
                controller.counter.toString(),
                style: Theme.of(context).textTheme.headlineMedium,
              ),
              ElevatedButton(
                  onPressed: () {
                    Get.toNamed('/detail');
                  },
                  child: Text('Go to Detail Page'))
            ],
          ),
        )),
        floatingActionButton: Row(
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          children: [
            FloatingActionButton(
              onPressed: controller.incrementCounter,
              tooltip: 'Increment',
              child: const Icon(Icons.add),
            ),
            FloatingActionButton(
              onPressed: controller.decrementCounter,
              tooltip: 'Decrement',
              child: const Icon(Icons.remove),
            ),
            FloatingActionButton(
              onPressed: controller.getsnackbar,
              tooltip: 'Snackbar',
              child: const Icon(Icons.chat),
            ),
            FloatingActionButton(
              onPressed: controller.getdialog,
              tooltip: 'Dialog',
              child: const Icon(Icons.notifications),
            ),
            FloatingActionButton(
              onPressed: controller.getbottomsheet,
              tooltip: 'Bottom Sheet',
              child: const Icon(Icons.arrow_upward),
            ),
          ],
        ),
      );
    }
}
```

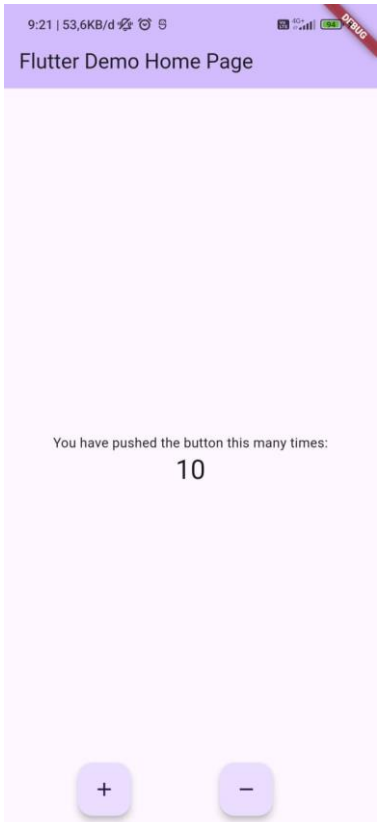
- view model/counter_controller.dart

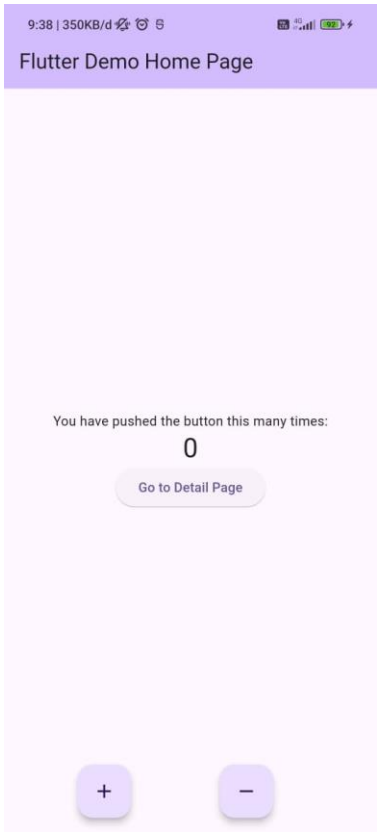
```
import 'package:flutter/material.dart';
import 'package:get/get.dart';

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

  void incrementCounter() {
    counter++;
  }

  void decrementCounter() {
    counter--;
  }

  void getsnackbar(){
    Get.snackbar(
      'GetX Snackbar',
      'Ini Snackbar',
      colorText: Colors.white,
      backgroundColor: Colors.green,
      );
  }

  void getdialog() {
    Get.defaultDialog(
      title: 'GetX Dialog',
      middleText: 'Ini dialog'
      );
  }

  void getbottomsheet(){
    Get.bottomSheet(Container(
      height: 70,
      width: double.infinity,
      color: Colors.amber,
      child: Text('Ini adalah Bottom Sheet'),
    )
    );
  }
}
```

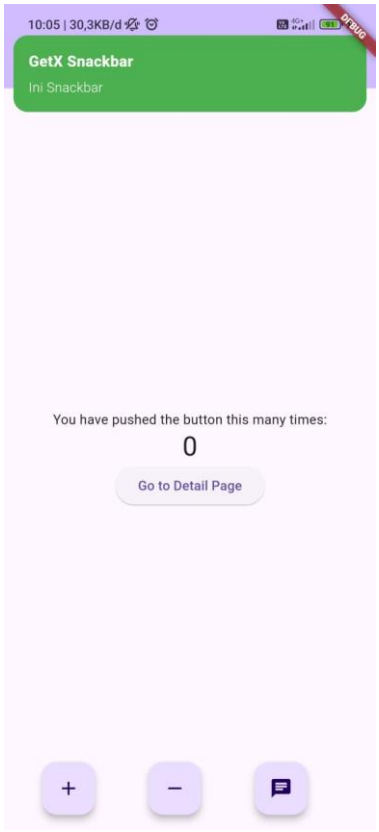
Output :

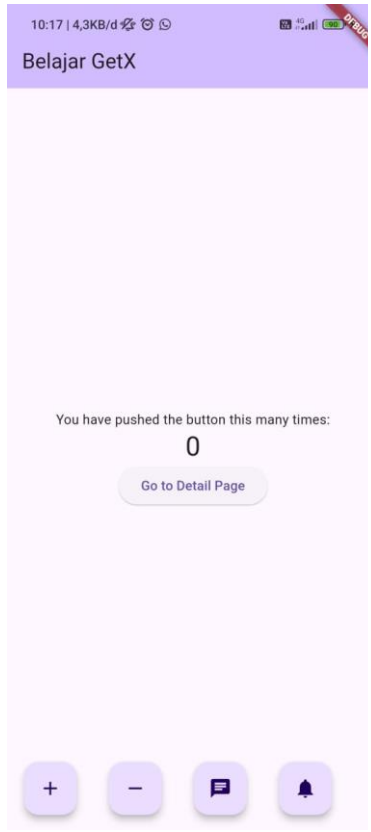
Telkom University Purwokerto

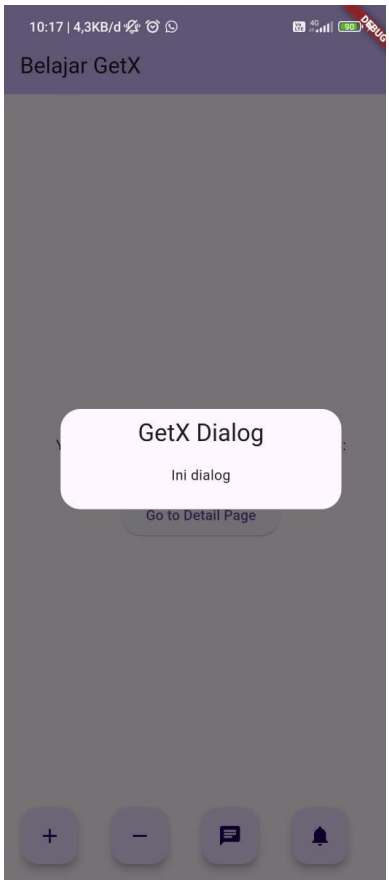


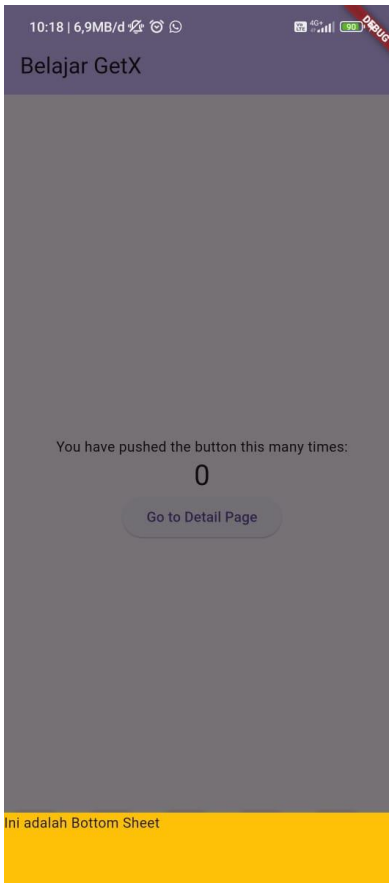












Unguided

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.

Input

- Main.dart

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'controllers/note_controller.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(
      debugShowCheckedModeBanner: false,
      title: 'Simple Notes App',
      initialRoute: '/',
      getPages: [
        GetPage(name: '/', page: () => HomePage()),
        GetPage(name: '/add', page: () => AddNotePage()),
      ],
    );
  }
}
```

- Controllers/Controller.dart

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

class Note {
  String title;
  String description;

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

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

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

  void deleteNoteAt(int index) {
    notes.removeAt(index);
  }
}
```

- 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 noteController = Get.find();
  final TextEditingController titleController = TextEditingController();
  final TextEditingController descriptionController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Tambah catatan'),
      ),
```

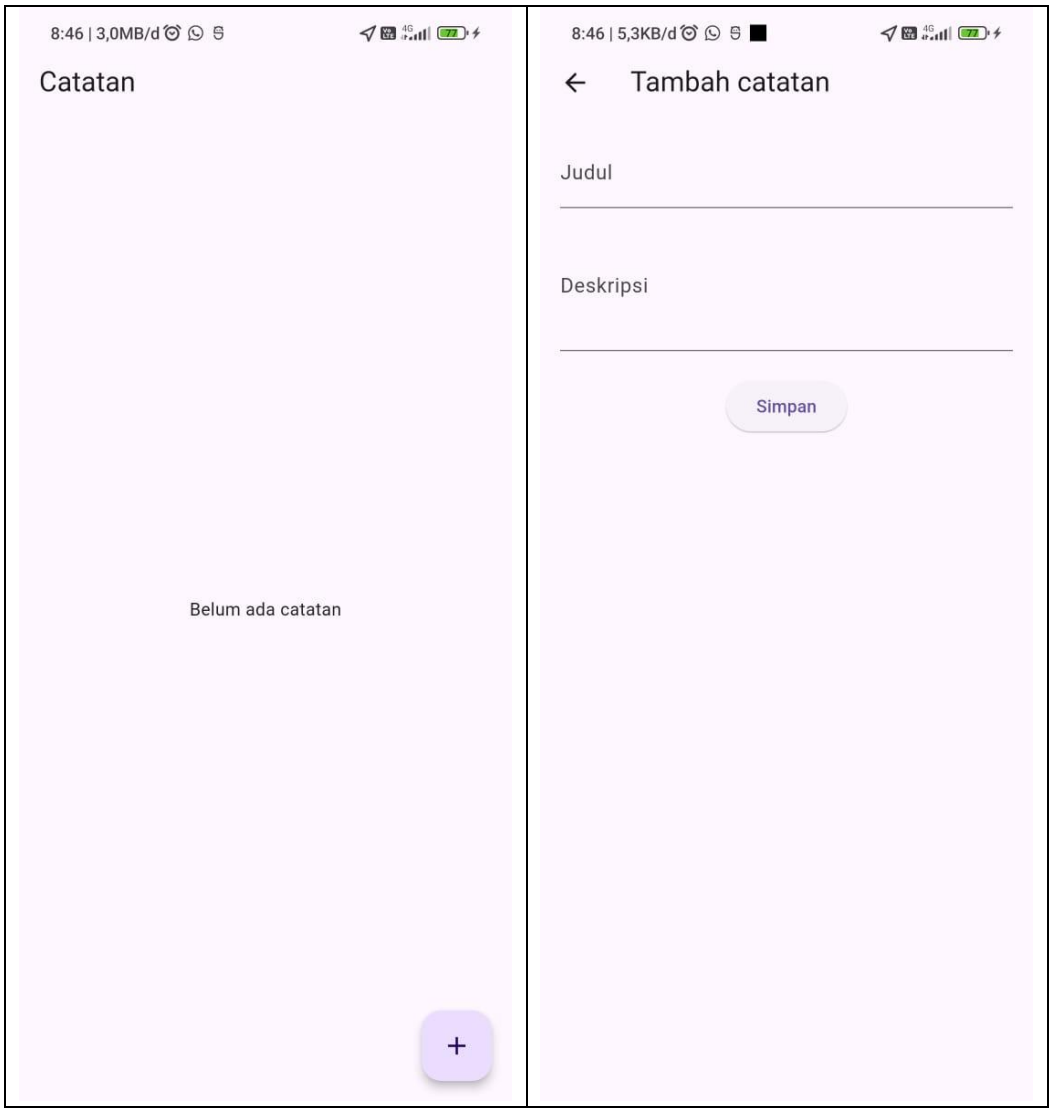
```
      body: Padding(
        padding: EdgeInsets.all(16.0),
        child: Column(
          children: [
            TextField(
              controller: titleController,
              decoration: InputDecoration(labelText: 'Judul'),
            ),
            SizedBox(height: 10),
            TextField(
              controller: descriptionController,
              decoration: InputDecoration(labelText: 'Deskripsi'),
              maxLines: 3,
            ),
            SizedBox(height: 20),
            ElevatedButton(
              onPressed: () {
                if (titleController.text.isNotEmpty &&
                    descriptionController.text.isNotEmpty) {
                  noteController.addNote(
                    titleController.text,
                    descriptionController.text,
                  );
                  Get.back();
                }
              },
              child: Text('Simpan'),
            ),
          ],
        ),
      ),
    );
  }
}
```

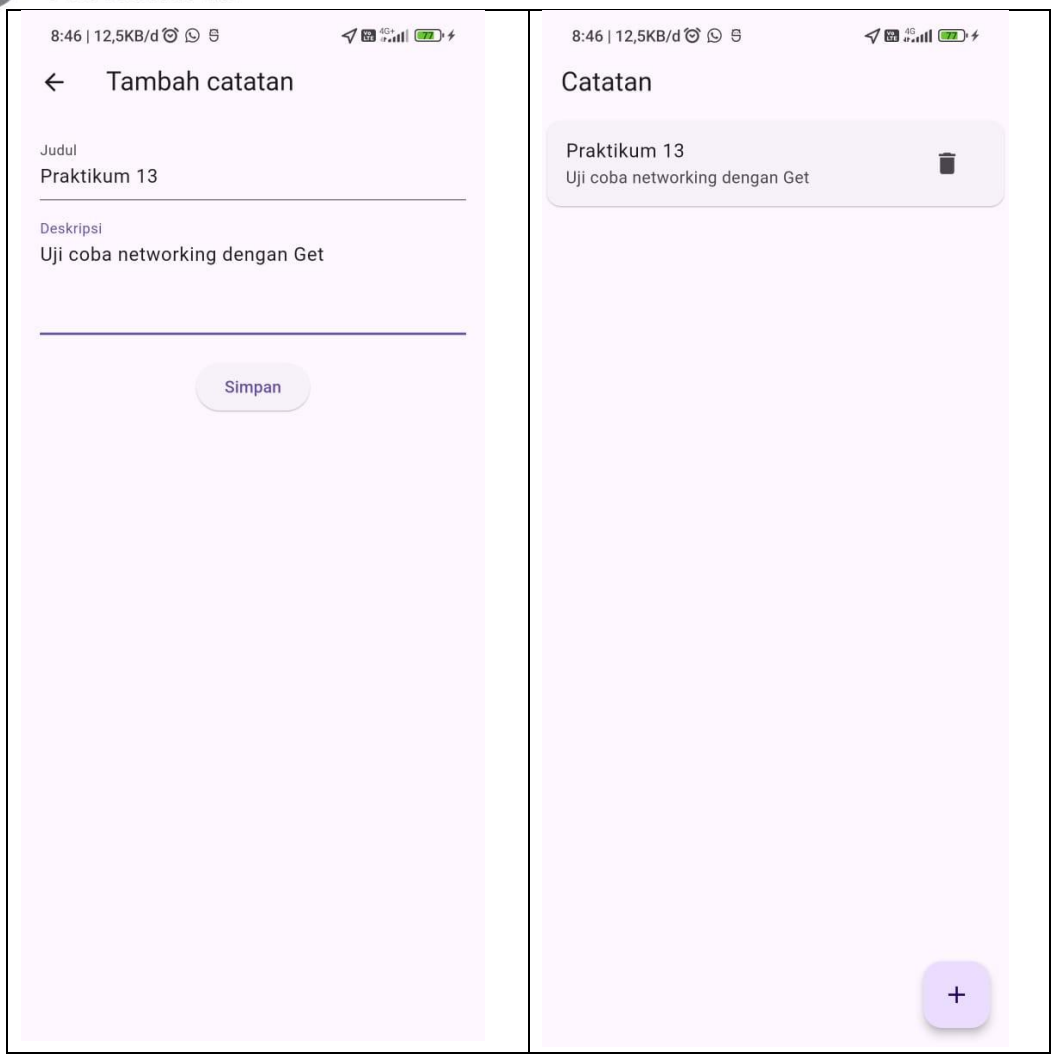
- Views/home_page.dart

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

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Catatan'),
      ),
      body: Obx(() {
        return noteController.notes.isEmpty
            ? Center(child: Text('Belum ada catatan'))
            : ListView.builder(
                itemCount: noteController.notes.length,
                itemBuilder: (context, index) {
```

```
                    final note = noteController.notes[index];
                    return Card(
                      child: ListTile(
                        title: Text(note.title),
                        subtitle: Text(note.description),
                        trailing: IconButton(
                          icon: Icon(Icons.delete),
                          onPressed: () => noteController.deleteNoteAt(index),
                        ),
                      ),
                    );
                  },
                );
        }),
        floatingActionButton: FloatingActionButton(
          onPressed: () => Get.toNamed('/add'),
          child: Icon(Icons.add),
        ),
      );
    }
}
```

Output


8:46 | 3,0MB/d
Catatan
Belum ada catatan
8:46 | 5,3KB/d
Tambah catatan
Judul
Deskripsi
Simpan



Deskripsi kode

Implementasi aplikasi catatan sederhana menggunakan Flutter dan GetX untuk manajemen state dan navigasi. Aplikasi memiliki dua halaman utama: halaman beranda untuk menampilkan daftar catatan dan halaman tambah catatan untuk menambahkan catatan baru. Data catatan dikelola secara reaktif melalui NoteController, yang memungkinkan penambahan dan penghapusan catatan secara langsung. Halaman beranda menggunakan widget Obx untuk memperbarui tampilan daftar catatan secara otomatis, sementara halaman tambah catatan menyediakan form input untuk judul dan deskripsi. Navigasi antar halaman dilakukan menggunakan Get.toNamed, menjadikan aplikasi sederhana ini responsif dan mudah dikembangkan.

Kesimpulan

Flutter menyediakan kerangka kerja UI deklaratif yang memungkinkan pengembangan antarmuka yang responsif dan efisien. GetX, sebagai pustaka manajemen state, mendukung pendekatan reaktif melalui observables (RxList) yang memperbarui tampilan secara otomatis saat data berubah. Selain itu, GetX juga menyederhanakan navigasi antar layar dengan deklarasi rute yang mudah dan metode navigasi yang ringkas seperti Get.toNamed.

Kombinasi ini menghasilkan aplikasi yang sederhana namun optimal, menonjolkan efisiensi dalam pengelolaan state dan pengalaman pengguna yang intuitif.