

GUIDED & UNGUIDED
PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK
MODUL XIII
NETWORKING



Disusun Oleh :

Maria Nathasya Desfera Pangestu / 2211104008

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 management-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  
  get: ^4.6.5
```

- 2) Konfigurasi dasar

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

```
import 'package:flutter/material.dart';  
import 'package:get/get.dart';  
  
void main() {  
  runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return GetMaterialApp(  
      title: 'Praktikum 13 - GetX',  
      home: HomePage(),  
    );  
  }  
}
```

- 3) State Management dengan GetX

a. Membuat Controller

Buat class controller untuk mengelola state. Misalnya, untuk counter sederhana:

```
import 'package:get/get.dart';  
  
class CounterController extends GetxController {  
  var count = 0.obs; // State yang reaktif  
  
  void increment() => count++;  
}
```

- 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';
import 'counter_controller.dart';

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('GetX State
Management')),
      body: Center(
        child: Obx(() => Text(
          'Counter: ${controller.count}',
          style: TextStyle(fontSize: 25),
        )),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: controller.increment,
        child: Icon(Icons.add),
      ),
    );
  }
}
```

4) Routing dengan GetX

a. Definisikan Route

Gunakan GetPage pada main.dart untuk mendefinisikan rute aplikasi :

```

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      initialRoute: '/',
      getPages: [
        GetPage(name: '/', page: () => HomePage()),
        GetPage(name: '/details', page: () =>
DetailsPage()),
      ],
    );
  }
}

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            Get.toNamed('/details'); // Navigasi ke
halaman lain
          },
          child: Text('Go to Details'),
        ),
      ),
    );
  }
}

```

```

class DetailsPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Text('Details Page'),
      ),
    );
  }
}

```

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

- Injeksi Sederhana

Gunakan `Get.put()` untuk membuat instance controller yang tersedia di mana saja :

```
final CounterController controller = Get.put(CounterController());
```

- Lazy Loading

Gunakan `Get.lazyPut()` jika ingin membuat instance hanya saat dibutuhkan:

```
Get.lazyPut(() => CounterController());
```

- Mengambil Instance

Ambil instance di mana saja dalam aplikasi :

```
final CounterController controller = Get.find();
```

6) Snackbar

```
Get.snackbar('Title', 'This is a snackbar');
```

7) Dialog

```
Get.defaultDialog(  
  title: 'Dialog Title',  
  middleText: 'This is a dialog',  
);
```

8) BottomSheet

```
Get.bottomSheet(  
  Container(  
    child: Text('This is a bottom sheet'),  
  ),  
);
```

Saat Praktikum Di Lab

1. Source code main.dart

```
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3  import 'package:guided/view/detail.dart';
4  import 'package:guided/view/homepage.dart';
5
6  void main() {
7    runApp(const MyApp());
8  }
9
10 class MyApp extends StatelessWidget {
11   const MyApp({super.key});
12
13   @override
14   Widget build(BuildContext context) {
15     return GetMaterialApp(
16       initialRoute: '/',
17       getPages: [
18         GetPage(
19           name: '/', page: () => MyHomePage(title: 'GetX Modul 13')),
20         GetPage(name: '/detail', page: () => DetailPage()),
21       ],
22     );
23   }
24 }
25
```

Source code detail.dart

```
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3
4  class DetailPage extends StatelessWidget {
5    const DetailPage({super.key});
6
7    @override
8    Widget build(BuildContext context) {
9      return Scaffold(
10        body: Center(
11          child: Text('Halaman Detail'),
12        ),
13      );
14    }
15  }
16
```

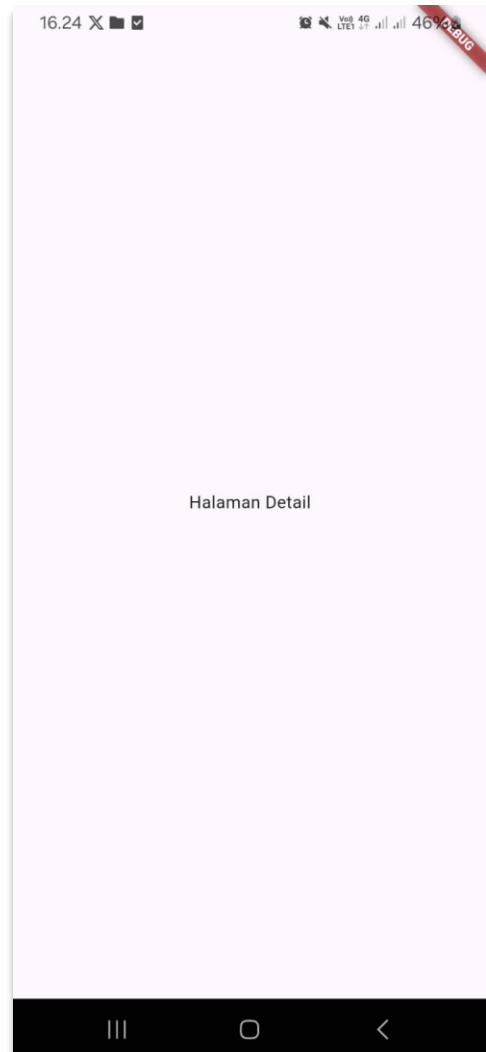
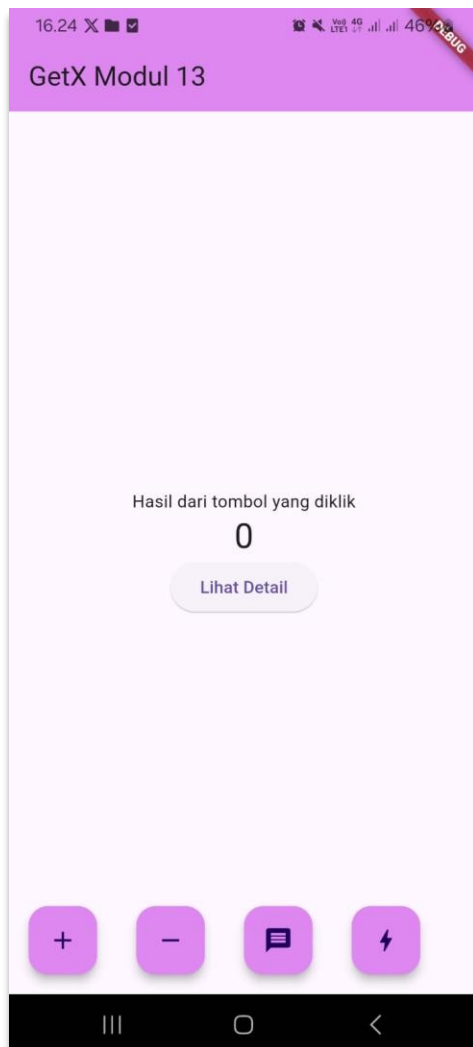
Source code homepage.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import 'package:guided/view_model/counter_controller.dart';
4
5 class MyHomePage extends StatelessWidget {
6   MyHomePage({super.key, required this.title});
7
8   final String title;
9   final CounterController controller = Get.put(CounterController());
10
11 @override
12 Widget build(BuildContext context) {
13   return Scaffold(
14     appBar: AppBar(
15       backgroundColor: const Color.fromARGB(255, 222, 135, 240),
16       title: Text(title),
17     ),
18     body: Center(
19       child: Obx(
20         () => Column(
21           mainAxisAlignment: MainAxisAlignment.center,
22           children: <Widget>[
23             const Text(
24               'Hasil dari tombol yang diklik',
25             ),
26             Text(
27               controller.counter.toString(),
28               style: Theme.of(context).textTheme.headlineMedium,
29             ),
30             ElevatedButton(
31               onPressed: () {
32                 Get.toNamed('/detail');
33               },
34               child: Text('Lihat Detail'))
35           ],
36         ),
37       ),
38     ),
39     floatingActionButton: Row(
40       mainAxisAlignment: MainAxisAlignment.spaceEvenly,
41       children: [
42         FloatingActionButton(
43           onPressed: controller.tambahCounter,
44           tooltip: 'Increment',
45           backgroundColor: const Color.fromARGB(255, 222, 135, 240),
46           child: const Icon(Icons.add),
47         ),
48         FloatingActionButton(
49           onPressed: controller.resetCounter,
50           tooltip: 'Decrement',
51           backgroundColor: const Color.fromARGB(255, 222, 135, 240),
52           child: const Icon(Icons.remove),
53         ),
54         FloatingActionButton(
55           onPressed: controller.getSnackBar,
56           tooltip: 'Get SnackBar',
57           backgroundColor: const Color.fromARGB(255, 222, 135, 240),
58           child: const Icon(Icons.message),
59         ),
60         FloatingActionButton(
61           onPressed: controller.getBottomSheet,
62           tooltip: 'Get Bottom Sheet',
63           backgroundColor: const Color.fromARGB(255, 222, 135, 240),
64           child: const Icon(Icons.bolt_outlined),
65         ),
66       ],
67     ),
68   );
69 }
70
71
```

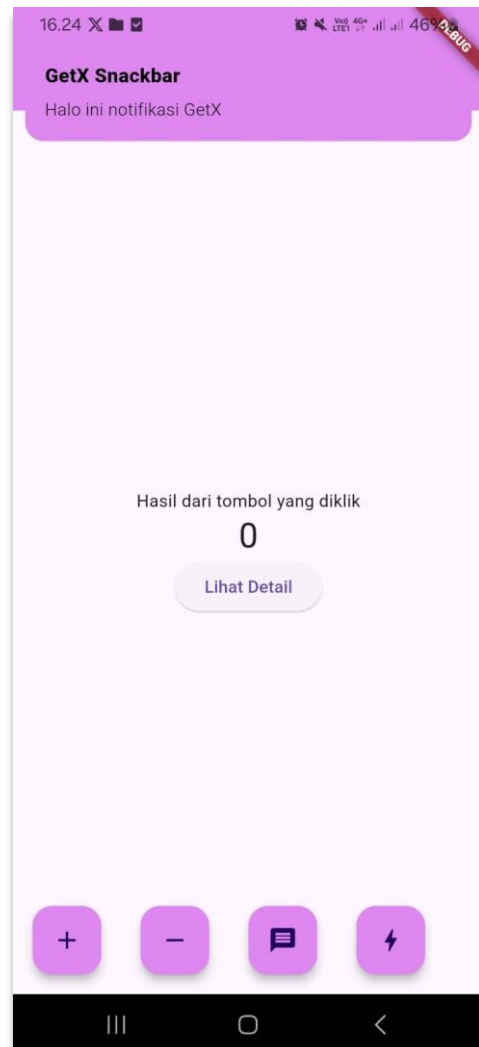
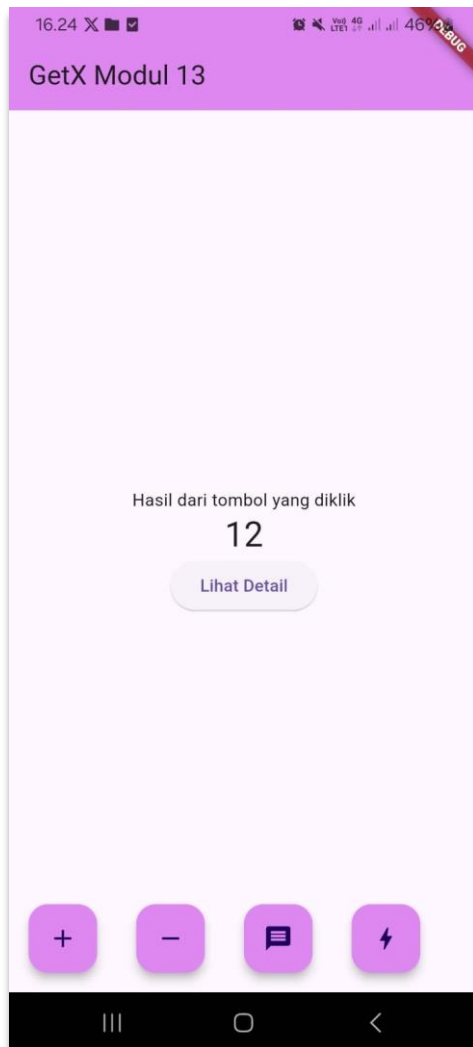

Source code counter_controller.dart

```
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3
4  class CounterController extends GetxController {
5    var counter = 0.obs;
6
7    void tambahCounter() => counter.value++;
8
9    void resetCounter() => counter.value = 0;
10   void getSnackBar() {
11     Get.snackbar(
12       'GetX Snackbar',
13       'Halo ini notifikasi GetX',
14       backgroundColor: const Color.fromARGB(255, 222, 135, 240),
15       colorText: Colors.black,
16     );
17   }
18
19   void getbottomshet() {
20     Get.bottomSheet(
21       Container(
22         height: 200,
23         color: const Color.fromARGB(255, 222, 135, 240),
24         child: const Center(
25           child: Text(
26             'Ini getX BottomSheet',
27             style: TextStyle(
28               color: Colors.black,
29               fontSize: 20,
30             ),
31           ),
32         ),
33       ),
34     );
35   }
36 }
37
```

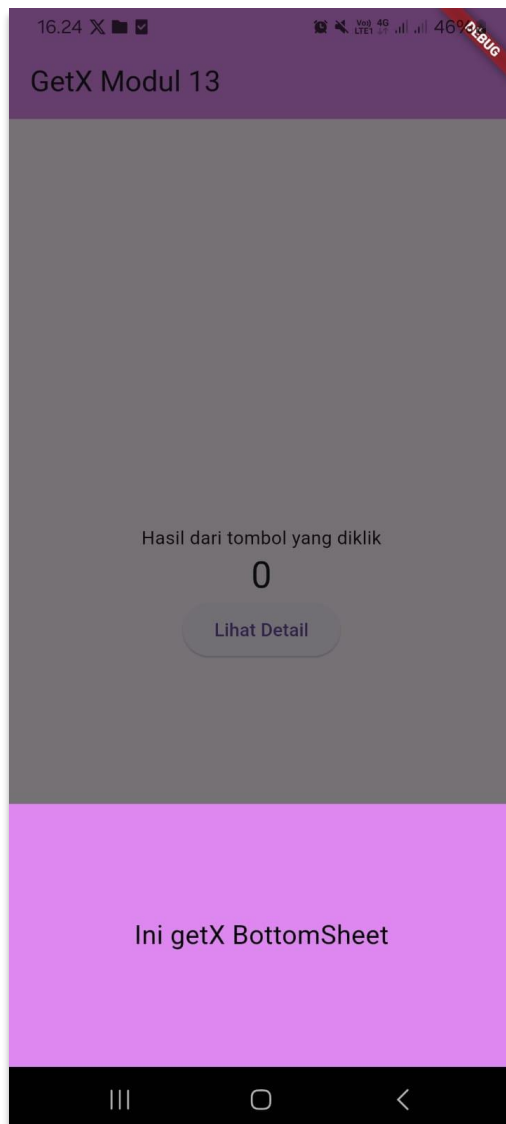
2. Screenshot tampilan output saat awal aplikasi dibuka dan saat memilih tombol lihat detail, maka tampilan halaman detail seperti gambar yang sebelah kanan.



Screenshot tampilan saat mengklik tombol “+” maka angkanya bertambah sesuai jumlah klik. Yang disebelah kanan tampilan saat mengklik tombol snackbar akan muncul notifikasi snackbar.



Screenshot tampilan saat mengklik tombol bottomsheets.



3. Penjelasan

Kode ini diimplementasikan menggunakan GetX untuk manajemen pemerintahan dan navigasi. Dengan memulai file main.dart, GetMaterialApp memungkinkan pengelolaan state dan konfigurasi rute. Kelas MyHomePage adalah sebuah StatelessWidget yang memiliki judul di AppBar, nilai counter yang dikelola secara reaktif dan beberapa tombol untuk berbagai fungsi. Sebagai hasil dari penggunaan Obx, nilai counter secara otomatis diperbarui saat terjadi perubahan. CounterController dapat diinisialisasi dengan menggunakan Get.put(CounterController()) untuk mengontrol nilai counter. Selain itu, ada empat tombol FloatingActionButton dan tombol navigasi yang dapat digunakan untuk membuka halaman detail. Tombol-tombol ini memiliki fungsi berbeda seperti menambah nilai, mereset nilai, menampilkan snackbar, dan menampilkan bottom sheet.

UNGUIDED

TUGAS MANDIRI (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.

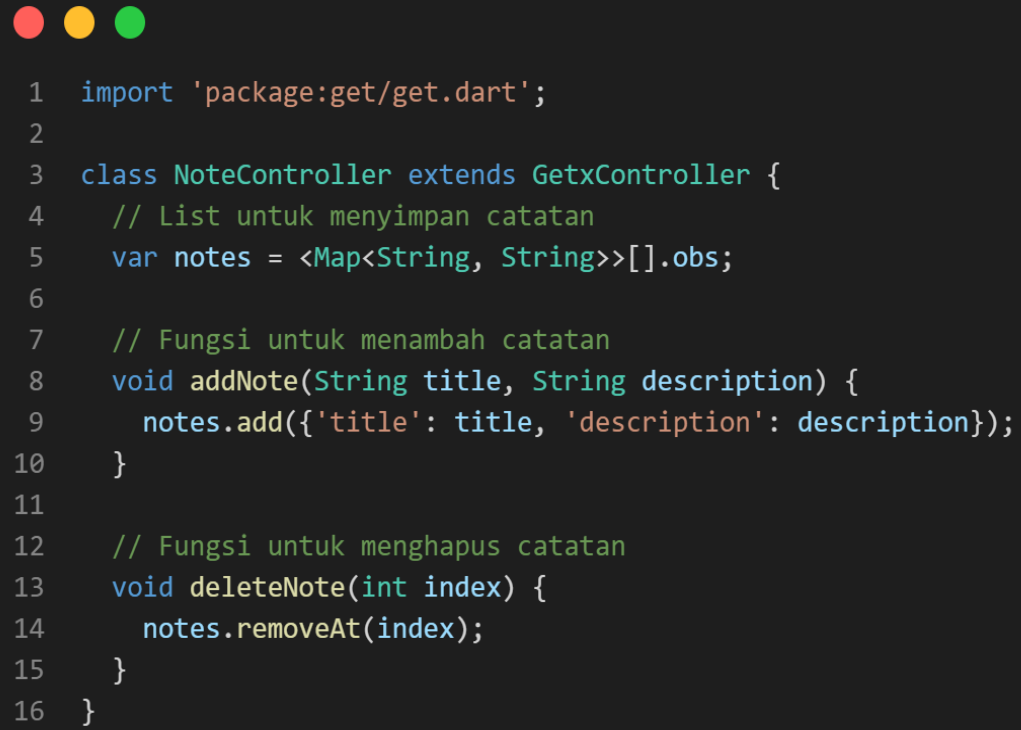
Note: Jangan lupa sertakan source code, screenshot output, dan deskripsi program. Kreativitas menjadi nilai tambah.

Jawab

1. Source code
 - Source code main.dart

```
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3  import 'package:unguided/view/homepage.dart';
4  import 'package:unguided/view/catatan.dart';
5
6  void main() {
7    runApp(const MyApp());
8  }
9
10 class MyApp extends StatelessWidget {
11   const MyApp({super.key});
12
13   @override
14   Widget build(BuildContext context) {
15     return GetMaterialApp(
16       initialRoute: '/',
17       getPages: [
18         GetPage(name: '/', page: () => HomePage()),
19         GetPage(name: '/add', page: () => TambahCatatan()),
20       ],
21     );
22   }
23 }
```

- Source code controller.dart



```
1  import 'package:get/get.dart';
2
3  class NoteController extends GetxController {
4    // List untuk menyimpan catatan
5    var notes = <Map<String, String>>[].obs;
6
7    // Fungsi untuk menambah catatan
8    void addNote(String title, String description) {
9      notes.add({'title': title, 'description': description});
10   }
11
12   // Fungsi untuk menghapus catatan
13   void deleteNote(int index) {
14     notes.removeAt(index);
15   }
16 }
```

- Source code homepage.dart

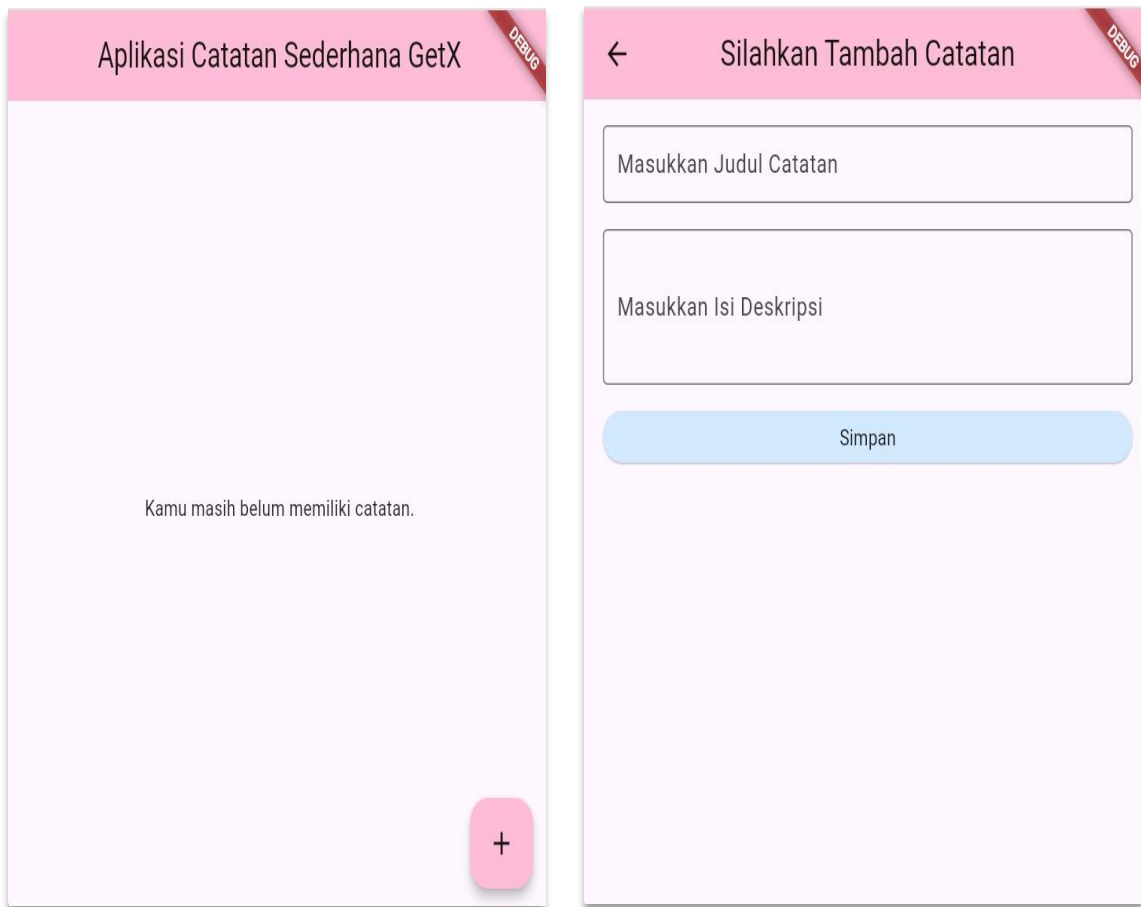
```
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import 'package:unguided/view_model/controller.dart';
4
5 class HomePage extends StatelessWidget {
6   HomePage({super.key});
7
8   final NoteController noteController = Get.put(NoteController());
9
10  @override
11  Widget build(BuildContext context) {
12    return Scaffold(
13      appBar: AppBar(
14        title: const Text('Aplikasi Catatan Sederhana GetX'),
15        centerTitle: true,
16        backgroundColor: const Color.fromARGB(255, 255, 188, 214),
17        foregroundColor: const Color.fromARGB(255, 20, 20, 20),
18      ),
19      body: Obx(
20        () => noteController.notes.isEmpty
21          ? const Center(
22            child: Text('Kamu masih belum memiliki catatan.'),
23          )
24          : ListView.builder(
25            itemCount: noteController.notes.length,
26            itemBuilder: (context, index) {
27              final note = noteController.notes[index];
28              return Card(
29                child: ListTile(
30                  title: Text(note['title']!),
31                  subtitle: Text(note['description']!),
32                  trailing: IconButton(
33                    icon: const Icon(Icons.delete),
34                    onPressed: () {
35                      noteController.deleteNote(index);
36                    },
37                  ),
38                ),
39              );
40            },
41          ),
42      ),
43      floatingActionButton: FloatingActionButton(
44        onPressed: () {
45          Get.toNamed('/add');
46        },
47        backgroundColor: const Color.fromARGB(255, 255, 188, 214),
48        foregroundColor: const Color.fromARGB(255, 20, 20, 20),
49        child: const Icon(Icons.add),
50      ),
51    );
52  }
53 }
```

- Source code catatan.dart

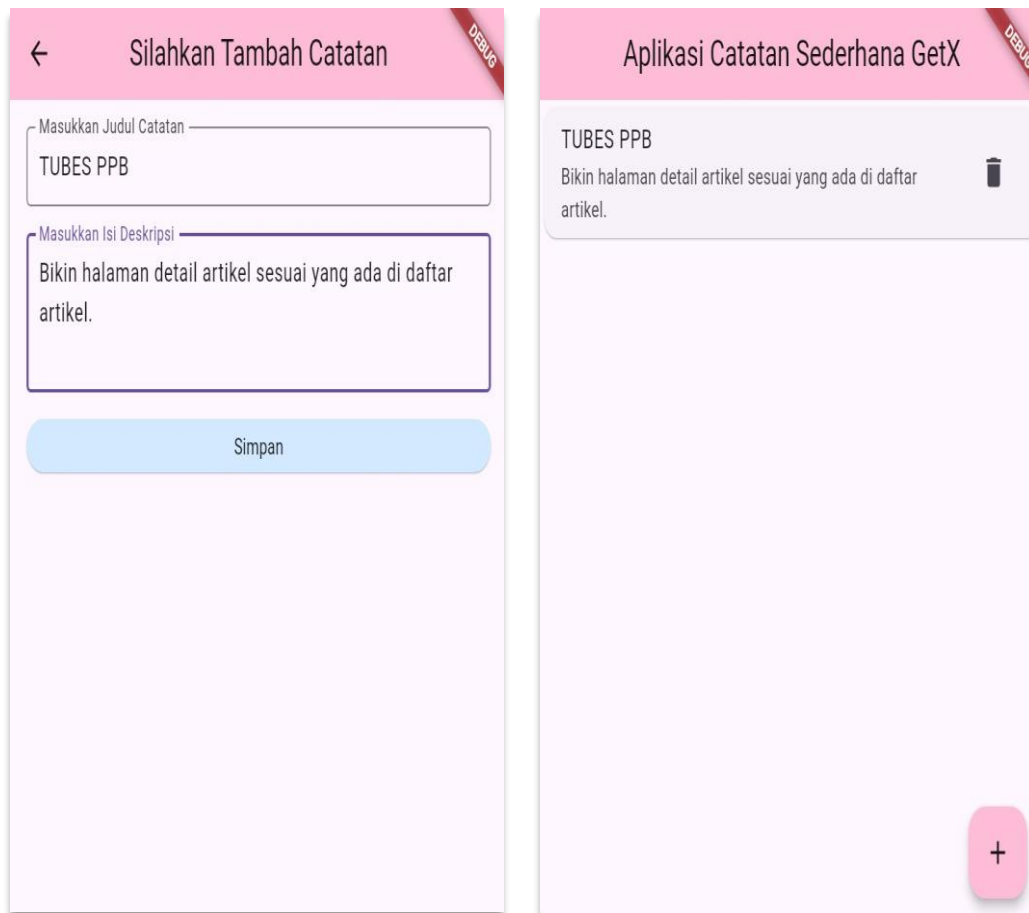
```
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import 'package:unguided/view_model/controller.dart';
4
5 class TambahCatatan extends StatelessWidget {
6   TambahCatatan({super.key});
7
8   final NoteController noteController = Get.find<NoteController>();
9   final TextEditingController titleController = TextEditingController();
10  final TextEditingController descriptionController = TextEditingController();
11
12  @override
13  Widget build(BuildContext context) {
14    return Scaffold(
15      appBar: AppBar(
16        title: const Text('Silahkan Tambah Catatan'),
17        centerTitle: true,
18        backgroundColor: const Color.fromARGB(255, 255, 188, 214),
19        foregroundColor: const Color.fromARGB(255, 20, 20, 20),
20      ),
21      body: Padding(
22        padding: const EdgeInsets.all(16.0),
23        child: Column(
24          crossAxisAlignment: CrossAxisAlignment.stretch,
25          children: [
26            TextField(
27              controller: titleController,
28              decoration: const InputDecoration(
29                labelText: 'Masukkan Judul Catatan',
30                border: OutlineInputBorder(),
31              ),
32            ),
33            const SizedBox(height: 16),
34            TextField(
35              controller: descriptionController,
36              decoration: const InputDecoration(
37                labelText: 'Masukkan Isi Deskripsi',
38                border: OutlineInputBorder(),
39              ),
40              maxLines: 3,
41            ),
42            const SizedBox(height: 16),
43            ElevatedButton(
44              onPressed: () {
45                if (titleController.text.isNotEmpty &&
46                  descriptionController.text.isNotEmpty) {
47                  noteController.addNote(
48                    titleController.text,
49                    descriptionController.text,
50                  );
51                  Get.back(); // Kembali ke halaman utama
52                } else {
53                  Get.snackbar(
54                    'Ada Kesalahan',
55                    'Judul dan Deskripsi Catatan tidak boleh kosong!',
56                    backgroundColor: const Color.fromARGB(255, 255, 104, 74),
57                    colorText: const Color.fromARGB(255, 20, 20, 20),
58                  );
59                }
60              },
61              style: ElevatedButton.styleFrom(
62                backgroundColor: const Color.fromARGB(255, 210, 233, 255), // Warna biru
63                foregroundColor: const Color.fromARGB(255, 20, 20, 20), // Warna teks hitam
64              ),
65              child: const Text('Simpan'),
66            ),
67          ],
68        ),
69      );
70  }
71 }
72
73
```


2. Screenshot output

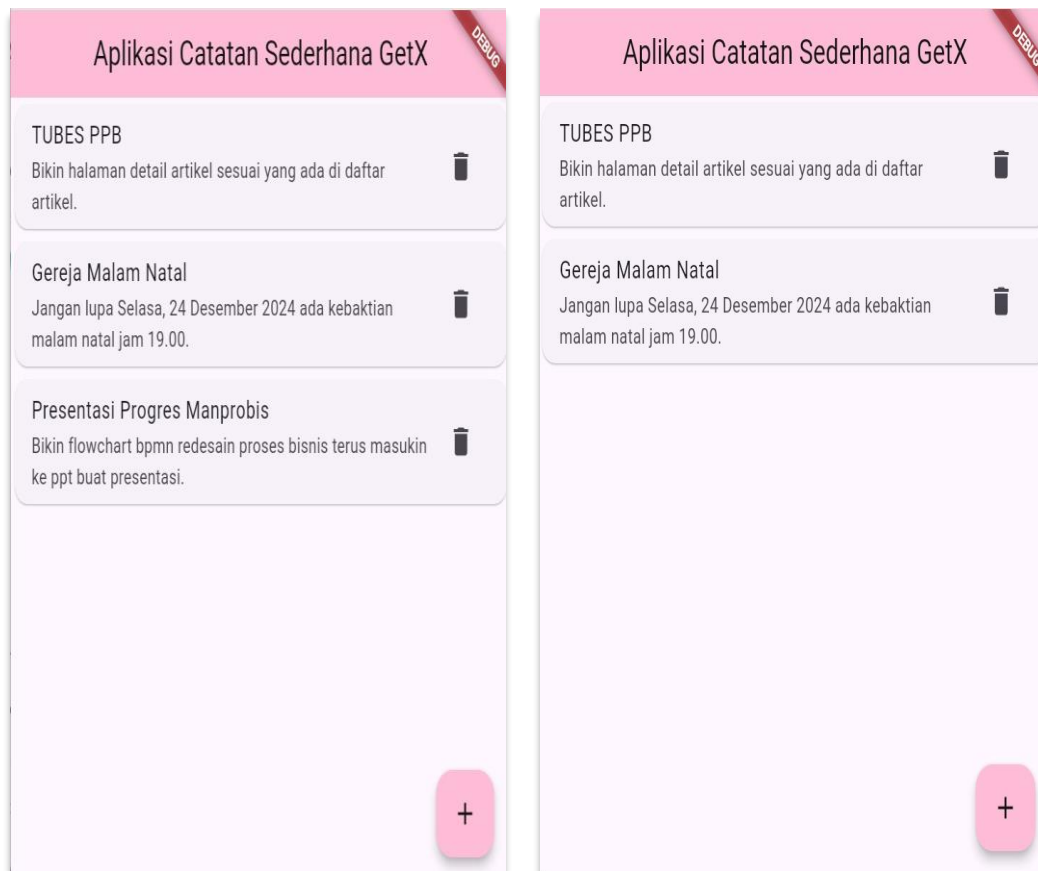
- Screenshot output awal dan tampilan saat memilih tombol “+” untuk menambahkan catatan.



- Screenshot output saat mengisi/menambahkan catatan dan saat catatan berhasil disimpan.



- Screenshot output saat sudah menambahkan beberapa catatan dan tampilan saat salah satu catatan dihapus.



3. Penjelasan

Kode program tersebut adalah halaman "Tambah Catatan" yang memungkinkan pengguna menambahkan catatan baru dengan mengisi judul dan deskripsi. Struktur utama menggunakan widget Scaffold yang mencakup AppBar dan body. Memiliki AppBar dengan latar belakang merah muda dan teks hitam. Serta terdiri dari dua TextField untuk input dan sebuah tombol "Simpan" berwarna biru muda dengan teks hitam. Tombol ini memvalidasi input, jika valid data dikirim ke NoteController untuk disimpan dan pengguna otomatis kembali ke halaman sebelumnya. Jika input kosong, snackbar dengan pesan kesalahan akan muncul di atas layar. Program ini menggunakan GetX untuk manajemen state dan navigasi yang efisien.