

LAPORAN PRAKTIKUM
Modul 13
“Networking”



Disusun Oleh:
Muhammad Ralfi - 2211104054
Kelas SE-06-2

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

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

1. Tujuan

- a. Mahasiswa mampu memahami state management dalam Flutter
- b. Mahasiswa mampu mengimplementasikan state management dalam Flutter

2. Landasan Teori

State management dalam flutter adalah proses mengelola data yang memengaruhi tampilan dan logika aplikasi, state management membantu pengembang dan desainer membuat aplikasi yang lebih dinamis sesuai dengan kebutuhan pengguna. Pada flutter ada dua tipe state yaitu stateless dan stateful, penggunaan keduanya bergantung pada kebutuhan.

a. Stateless Widget

Ketika menggunakan beberapa widget flutter yang memiliki data tidak berubah-ubah atau immutable, maka stateless lebih tepat digunakan, contohnya halaman contact us, pricing dan lain sebagainya.

b. Statefull Widget

Tidak semua aplikasi yang akan dibangun hanya menggunakan data statis, terkadang juga membutuhkan data dinamis, maka stateful widget lebih tepat penggunaannya.

3. Guided

a. File main.dart

```
import 'package:flutter/material.dart';
import 'package:gd13/view/detail_page.dart';
import 'package:gd13/view/my_home_page.dart';
import 'package:gd13/view_model/counter_controller.dart';
import 'package:get/get.dart';

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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      // navigasi halaman menggunakan get initial route
      initialRoute: '/',
      getPages: [
        GetPage(
          name: '/',
          page: () => MyHomePage(title: 'Flutter Demo Home Page'),
        ),
        GetPage(
          name: '/detail',
          page: () => const MyDetail(),
        ),
      ],
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
    );
  }
}
```

```

    theme: ThemeData(
      colorScheme: ColorScheme.fromSeed(seedColor:
Colors.deepPurple),
      useMaterial3: true,
    ),
    home: MyHomePage(title: 'Flutter Demo Home Page'),
  );
}
}

```

b. File homepage

```

import 'package:flutter/material.dart';
import 'package:gd13/view/detail_page.dart';
import 'package:gd13/view_model/counter_controller.dart';
import 'package:get/get.dart';

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

  final String title;
  // menagmbil insane dari get put didalam file main.dart
  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');
                Get.to(MyDetail());
              },
              child: const 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.getSnackBar,
        tooltip: 'SnackBar',
        child: const Icon(Icons.notifications),
      ),
      FloatingActionButton(
        onPressed: controller.getDialog,
        tooltip: 'Dialog',
        child: const Icon(Icons.notifications_active),
      ),
      FloatingActionButton(
        onPressed: controller.decrementCounter,
        tooltip: 'Decrement',
        child: const Icon(Icons.remove),
      ),
      FloatingActionButton(
        onPressed: controller.getBottomSheet,
        tooltip: 'bottom sheet',
        child: const Icon(Icons.arrow_upward),
      ),
    ],
  ),
);
}
}

```

c. File 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('PROMO 12.12', 'Guncang desember dengan diskon serba 12K', colorText: Colors.white, backgroundColor: Colors.green, snackPosition: SnackPosition.TOP, duration: const Duration(seconds: 3));
  }

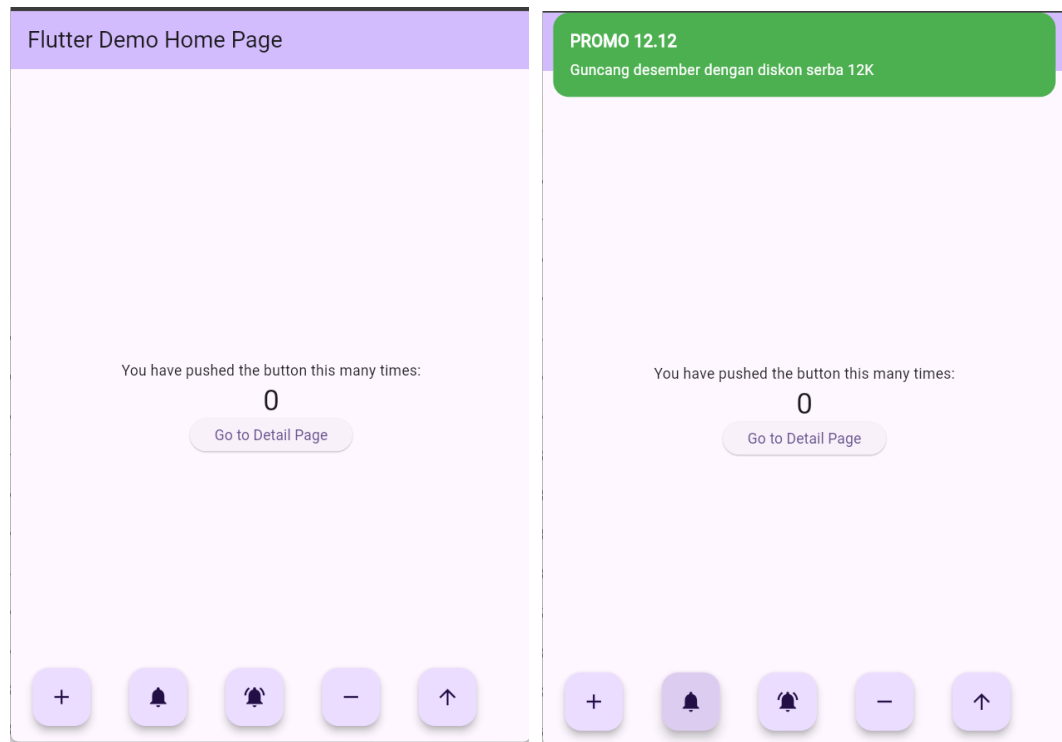
  void getDialog(){
    Get.defaultDialog(
      title: 'PROMO 12.12',
      middleText: 'Guncang desember dengan diskon serba 12K',
    );
  }

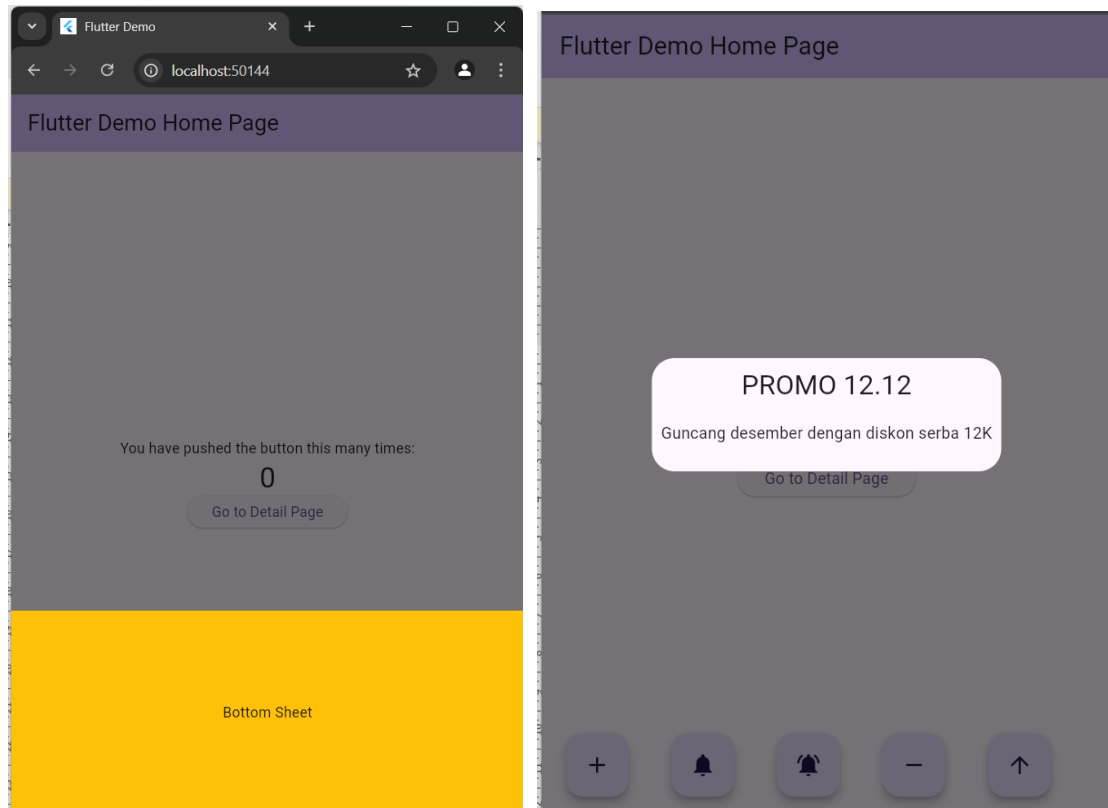
  void getBottomSheet(){

```

```
Get.bottomSheet(  
  Container(  
    height: 200,  
    color: Colors.amber,  
    child: Center(  
      child: Text('Bottom Sheet'),  
    ),  
  ),  
);  
}
```

Output





4. Unguided

a. File main

```
import 'package:flutter/material.dart';
import 'package:noteapps/homepage.dart';
import 'package:get/get.dart';
import 'controller/note_controller.dart';

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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor:
Colors.deepPurple),
        useMaterial3: true,
      ),
      home: HomePages(),
      initialBinding: BindingsBuilder((){
        Get.put(NoteController());
      }),
    );
  }
}
```

```

    );
  }
}

```

b. File home

```

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

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.purpleAccent,
        title: Text('Simple Notes App'),
      ),
      body: Obx(() => ListView.builder(
        itemCount: noteController.notes.length,
        itemBuilder: (context, index) {
          final note = noteController.notes[index];
          return ListTile(
            title: Text(note.title),
            subtitle: Text(note.description),
            trailing: IconButton(
              icon: Icon(Icons.delete),
              onPressed: () => noteController.deleteNote(index),
            ),
          );
        },
      )),
      floatingActionButton: FloatingActionButton(
        child: Icon(Icons.add),
        onPressed: () => Get.to(AddNotePage()),
      ),
    );
  }
}

```

c. File controller

```

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

```

d. File Addnote

```

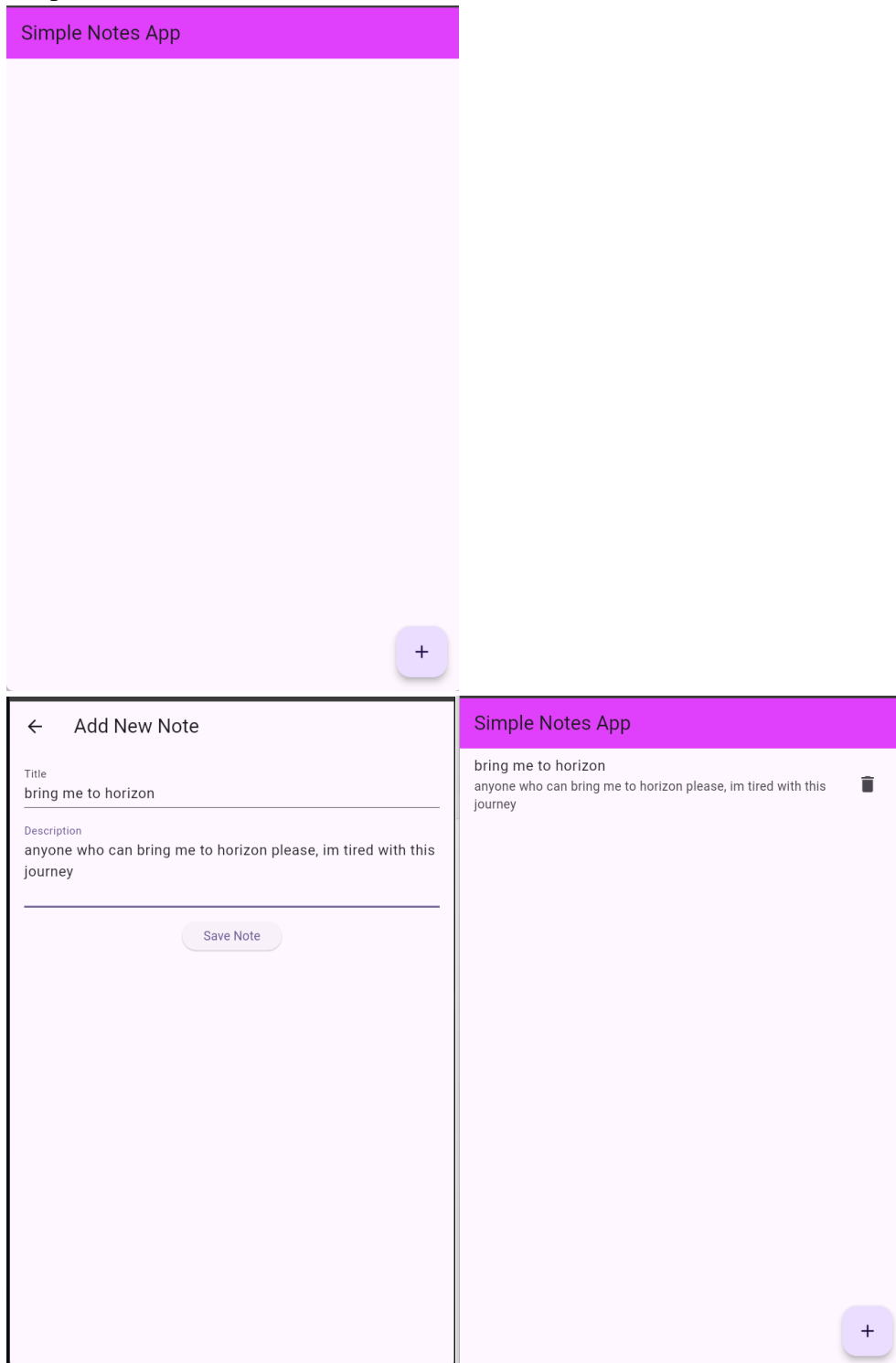
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'controller/note_controller.dart';

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('Add New Note'),
            ),
            body: Padding(
                padding: const EdgeInsets.all(16.0),
                child: Column(
                    children: [
                        TextField(
                            controller: titleController,
                            decoration: InputDecoration(labelText: 'Title'),
                        ),
                        SizedBox(height: 16),
                        TextField(
                            controller: descriptionController,
                            decoration: InputDecoration(labelText: 'Description'),
                            maxLines: 3,
                        ),
                        SizedBox(height: 16),
                        ElevatedButton(
                            child: Text('Save Note'),
                            onPressed: () {
                                noteController.addNote(
                                    titleController.text,
                                    descriptionController.text,
                                );
                                Get.back();
                            },
                        ),
                    ],
                ),
            ),
        );
    }
}

```


Output



5. Kesimpulan

Berdasarkan hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa State Management dan GetX pada Flutter merupakan komponen penting dalam pengembangan aplikasi yang dinamis dan efisien. State Management memungkinkan pengelolaan data aplikasi secara terstruktur, sementara GetX menyediakan solusi yang ringan dan praktis untuk menangani state, route, dan dependency injection.

Implementasi kedua komponen ini membantu dalam menciptakan kode yang lebih terorganisir dan mudah dipelihara.

PenggunaanGetX terbukti memberikan keuntungan signifikan dalam pengembangan aplikasi Flutter, terutama dalam hal performa dan kemudahan penggunaan. Framework ini menawarkan sintaks yang sederhana namun powerful untuk manajemen state, navigasi, dan pengelolaan dependensi, yang menghasilkan pengembangan aplikasi yang lebih cepat dan efisien. Selain itu, GetX juga membantu dalam memisahkan logika bisnis dari tampilan UI, sehingga memudahkan dalam melakukan testing dan maintenance aplikasi di masa mendatang.