

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XIII
NETWORKING**



Disusun Oleh :

Dwi Candra Pratama / 2211104035

SE06-02

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**

GUIDED

TUJUAN PRAKTIKUM

- Mahasiswa mampu memahami state management dalam Flutter
- Mahasiswa mampu mengimplementasikan state management dalam Flutter

NETWORKING/STATE MANAGEMENT

State management merupakan proses mengelola state atau status suatu aplikasi, yaitu data atau informasi yang dapat berubah selama siklus hidup aplikasi. State mencakup semua hal yang memengaruhi antarmuka pengguna, seperti input pengguna, data dari API, dan status internal widget. Ketika aplikasi menjadi semakin kompleks, akan tiba saatnya state harus dibagikan ke berbagai halaman dalam aplikasi. Flutter bersifat deklaratif, sehingga membangun antarmuka pengguna berdasarkan state yang ada saat itu. Dengan state management, semua state dari berbagai UI Control dapat dipusatkan untuk mengatur aliran data di seluruh aplikasi.

State management sangat penting karena aplikasi Flutter memiliki banyak widget yang saling berhubungan. Dengan pengelolaan state yang tepat, dapat dipastikan bahwa:

- UI dan data selalu tersinkronisasi, mencerminkan data yang terbaru.
- Struktur kode menjadi lebih terorganisir, sehingga mempermudah pengembangan dan pemeliharaan.
- Mengurangi potensi bug, karena state yang terkelola dengan baik meminimalkan kemungkinan kesalahan.

JENIS STATE DALAM FLUTTER

1. Ephemeral State (State Lokal)

Ephemeral state hanya relevan untuk widget tertentu dan tidak dibagikan ke widget lain. Contoh penggunaannya adalah untuk mengelola state pada TextField atau Checkbox. Untuk mengelola ephemeral state, biasanya digunakan StatefulWidget. Ada dua pendekatan dalam pengelolaan state ini, yaitu StatefulWidget untuk state lokal dan InheritedWidget untuk berbagi state antar widget.

2. App State (State Global)

App state digunakan di berbagai widget dalam aplikasi. Contohnya adalah data pengguna yang masuk, keranjang belanja, atau tema aplikasi. Karena cakupan yang lebih luas, pengelolaan app state memerlukan pendekatan state management yang lebih kompleks, yang didukung oleh berbagai library Flutter, seperti:

a. Provider

Library resmi yang didukung oleh tim Flutter, memanfaatkan InheritedWidget dengan cara yang lebih sederhana dan efisien.

- b. Bloc/Cubit
Pendekatan berbasis pola stream untuk memisahkan business logic dari UI, cocok untuk aplikasi besar dan kompleks.
- c. Riverpod
Framework modern yang dirancang untuk menggantikan atau menjadi alternatif dari Provider, menawarkan fleksibilitas lebih tinggi.
- d. GetX
Framework serbaguna untuk state management, routing, dan dependency injection. GetX meminimalkan boilerplate code dan mendukung aplikasi dengan reaktivitas tinggi, menjadikannya solusi yang efisien.

Berikut cara instalasi GetX :

- a. Tambahkan Package GetX ke pubspec.yaml

```
30 dependencies:
31 flutter:
32   sdk: flutter
33   cupertino_icons: ^1.0.8
34   get: ^4.6.6
```

- b. Konfigurasi dasar dan State Management dengan GetX

Berikut codenya:

Main.dart

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import
'package:pertemuan_13/view%20model/counter_controller.dart';
import 'package:pertemuan_13/view/detail_page.dart';
import 'package:pertemuan_13/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(
```

```

        debugShowCheckedModeBanner: false,
        initialRoute: '/',
        getPages: [
          GetPage(
            name: '/',
            page: () => MyHomePage(title: 'Belajar GetX'),
          ),
          GetPage(
            name: '/Detail',
            page: () => DetailPage(),
          ),
        ],
      );
    }
  }
}

```

- Lalu membuat Controller dan menggunakannya di UI

```

import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:pertemuan_13/view%20model/counter_controller.dart';

class MyHomePage extends StatelessWidget {
  MyHomePage({super.key, required this.title});
  final controller = Get.find<CounterController>();

  final String title;

  @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'),
      )
    ],
  ),
)),
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_active),
    ),
    FloatingActionButton(
      onPressed: controller.getbottomsheet,
      tooltip: 'bottom sheet',
      child: const Icon(Icons.arrow_upward),
    ),
  ],
), // This trailing comma makes auto-formatting nicer for build
methods.
);
}
}

```

c. Routing dengan GetX

- Definisikan Route

Gunakan GetPage pada main.dart untuk mendefinisikan rute aplikasi :

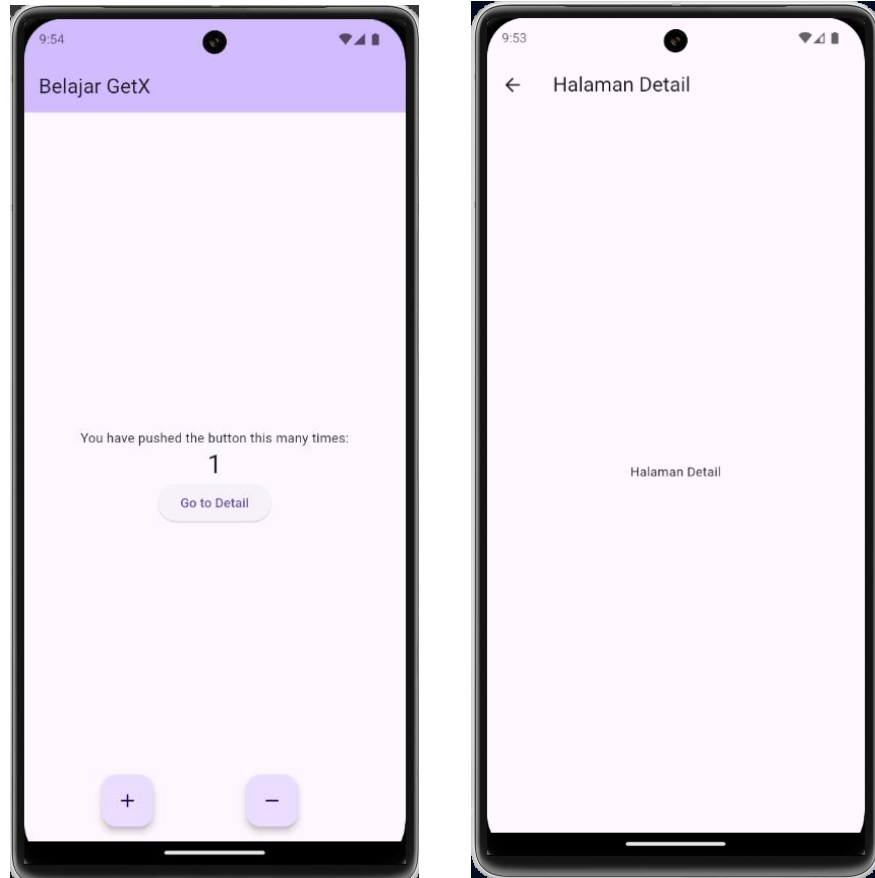
```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import
'package:pertemuan_13/view%20model/counter_controller.dart';

class MyHomePage extends StatelessWidget {
  MyHomePage({super.key, required this.title});
  final controller = Get.find<CounterController>();

  final String title;

  @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'),
              )
            ],
          ),
        ),
      ),
    ),
  ),
);
```

- 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.



d. Dependency Injection dengan GetX

1. Injeksi Sederhana Gunakan Get.put() untuk membuat instance controller yang tersedia di mana saja :
2. Lazy Loading Gunakan Get.lazyPut() jika ingin membuat instance hanya saat dibutuhkan :
3. Mengambil Instance Ambil instance di mana saja dalam aplikasi:

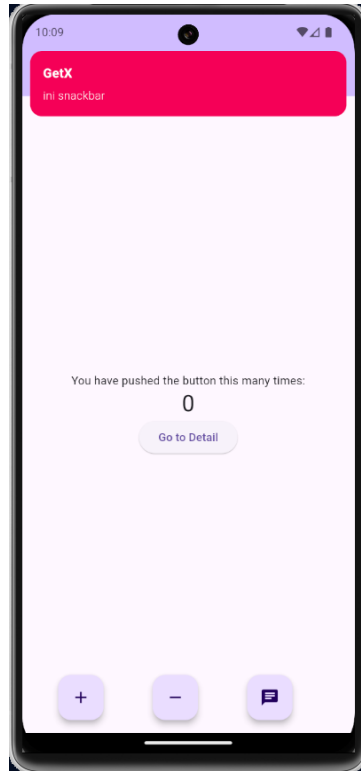
e. Snackbar

```
void getsnackbar() {
  Get.snackbar(
    'GetX',
    'ini snackbar',
    colorText: Colors.white,
    backgroundColor: Colors.pinkAccent[400],
  );
}
```

```

FloatingActionButton(
  onPressed: controller.getsnackbar,
  tooltip: 'Snackbar',
  child: const Icon(Icons.chat),
),

```



f. Dialog

```

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

```

```

FloatingActionButton(
  onPressed: controller.getdialog,
  tooltip: 'dialog',
  child: const Icon(Icons.notifications_active),
),

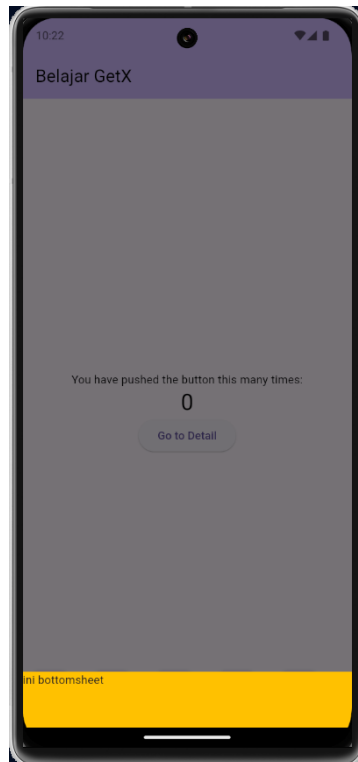
```




g. BottomSheet

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

```
FloatingActionButton(  
  onPressed: controller.getbottomsheet,  
  tooltip: 'bottom sheet',  
  child: const Icon(Icons.arrow_upward),  
),
```



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.

Isi package seperti ini:

```
37   cupertino_icons: ^1.0.8
38   get: ^4.6.6
```

Source Code nya:

Folder **Controller** isi file **note_controller.dart**

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

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

```

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

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

```

Folder **Routes** isi file **app_pages.dart**

```

import 'package:get/get.dart';
import 'package:pertemuan_13/views/home_page.dart';
import 'package:pertemuan_13/views/add_note_page.dart';

class AppPages {
  static final routes = [
    GetPage(
      name: '/home',
      page: () => HomePage(),
    ),
    GetPage(
      name: '/add-note',
      page: () => AddNotePage(),
    ),
  ];
}

```

Folder **views** isi folder **add_note_page.dart** & **home_page.dart**
Add_note_page.dart

```

import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:pertemuan_13/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: const Text('Tambah Catatan'),

```

```
        backgroundColor: Colors.pinkAccent[400],
      ),
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        children: [
          TextField(
            controller: titleController,
            decoration: InputDecoration(
              labelText: 'Judul',
              border: OutlineInputBorder(
                borderRadius: BorderRadius.circular(12.0),
              ),
            ),
          ),
          const SizedBox(height: 16),
          TextField(
            controller: descriptionController,
            maxLines: 3,
            decoration: InputDecoration(
              labelText: 'Deskripsi',
              border: OutlineInputBorder(
                borderRadius: BorderRadius.circular(12.0),
              ),
            ),
          ),
          const SizedBox(height: 20),
          ElevatedButton(
            onPressed: () {
              final title = titleController.text;
              final description = descriptionController.text;
              if (title.isNotEmpty && description.isNotEmpty) {
                noteController.addNote(title, description);
                Get.back();
              }
            },
            style: ElevatedButton.styleFrom(
              backgroundColor: Colors.pinkAccent[400],
              padding: const EdgeInsets.symmetric(
                horizontal: 24.0, vertical: 12.0),
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(12.0),
```

```

        ),
      ),
      child: const Text(
        'Simpan', // Teks pada tombol
        style: TextStyle(fontSize: 16),
      ),
    ),
  ],
),
),
);
}
}

```

home_page.dart

```

import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:pertemuan_13/controller/note_controller.dart';

class HomePage extends StatelessWidget {

  final NoteController noteController = Get.put(NoteController());

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Catatan'),
        backgroundColor: Colors.pinkAccent[400],
      ),
      body: Obx(() => noteController.notes.isEmpty
        ? Center(

          child: Text(
            'Belum ada catatan!',
            style: TextStyle(fontSize: 18, color: Colors.grey[600]),
          ),
        )
        : ListView.builder(

          padding: const EdgeInsets.all(16.0),
          itemCount: noteController.notes.length,
          itemBuilder: (context, index) {
            final note = noteController.notes[index];
            return Card(

```

```

        elevation: 4,
        margin: const EdgeInsets.symmetric(vertical: 8),
        child: ListTile(
          title: Text(
            note['title']!,
            style: const TextStyle(
              fontWeight: FontWeight.bold, fontSize: 16),
          ),
          subtitle: Text(note['description']!),
          trailing: IconButton(
            icon: const Icon(Icons.delete, color: Colors.red),
            onPressed: () {
              noteController.removeNoteAt(index);
            },
          ),
        ),
      );
    },
  )),
  floatingActionButton: FloatingActionButton(
    backgroundColor: Colors.pinkAccent[400],
    onPressed: () => Get.toNamed('/add-note'),
    child: const Icon(Icons.add), // Ikon tambah
  ),
);
}
}

```

Main.dart

```

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

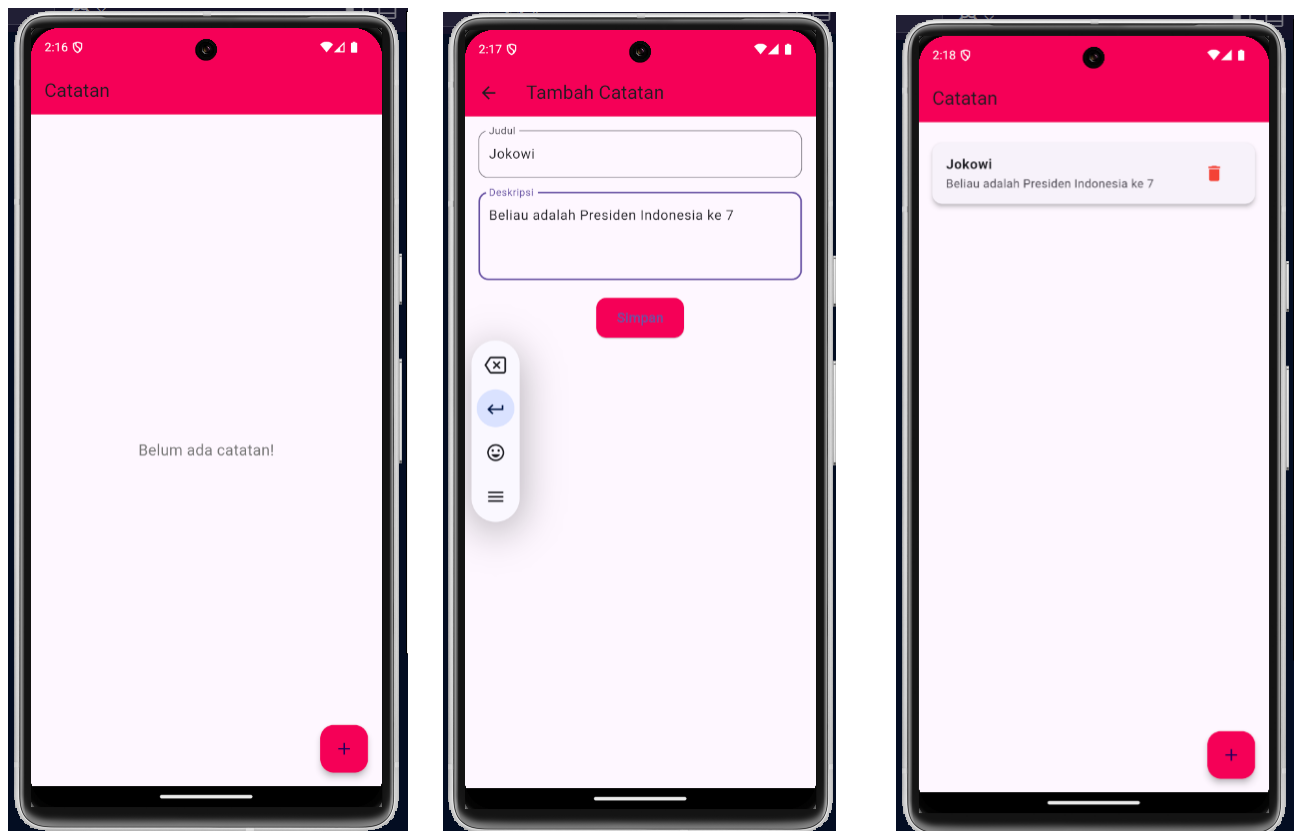
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      debugShowCheckedModeBanner: false,
      initialRoute: '/home',
      getPages: AppPages.routes,
    );
  }
}

```

}

OutPutnya



Deskripsi Program:

Aplikasi ini merupakan program pencatatan sederhana yang dibangun menggunakan Flutter dan memanfaatkan state management dari GetX. Aplikasi ini terdiri dari dua halaman utama: halaman beranda untuk menampilkan daftar catatan dan halaman tambah catatan untuk membuat catatan baru. Di halaman beranda, pengguna dapat melihat daftar catatan dengan judul dan deskripsinya, serta terdapat opsi untuk menghapus catatan. Untuk menambahkan catatan baru, pengguna cukup menekan tombol tambah (Floating Action Button) yang ada di bagian kanan bawah layar.

Pengelolaan data catatan dilakukan secara reaktif menggunakan GetX Controller, sehingga setiap perubahan seperti menambah atau menghapus catatan akan langsung terlihat tanpa memuat ulang halaman. Navigasi antar halaman dirancang menggunakan GetX Routing untuk memastikan pengalaman pengguna yang lebih lancar.