

**TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK**

MODUL XIII NETWORKING



**Disusun Oleh :
Yogi Hafidh Maulana / 2211104061
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**

TUGAS PENDAHULUAN

SOAL

1. Apa yang dimaksud dengan state management pada Flutter?

State management dalam Flutter merujuk pada cara mengelola dan mempertahankan data atau keadaan (state) dari aplikasi agar dapat digunakan secara konsisten di seluruh antarmuka pengguna (UI). State adalah data yang berubah selama aplikasi berjalan, seperti input pengguna, status login, atau data yang diambil dari server. Dalam konteks Flutter, state management sangat penting karena Flutter menggunakan pendekatan deklaratif untuk membangun UI, yang berarti UI dirender ulang setiap kali state berubah. Oleh karena itu, pengelolaan state yang efisien memastikan aplikasi tetap responsif dan data selalu diperbarui dengan benar.

Ada beberapa pendekatan state management yang dapat digunakan di Flutter, tergantung pada kebutuhan aplikasi. Pendekatan sederhana menggunakan `StatefulWidget` untuk menangani state lokal dalam widget tertentu. Namun, untuk aplikasi yang lebih kompleks, diperlukan solusi yang lebih terstruktur seperti Provider, Riverpod, Redux, atau Bloc. Pendekatan ini memungkinkan pengelolaan state global yang dapat diakses di berbagai bagian aplikasi, memisahkan logika bisnis dari UI, dan membuat kode lebih mudah dipelihara. Dengan manajemen state yang baik, pengembang dapat memastikan pengalaman pengguna yang lancar dan efisien di seluruh aplikasi.

2. Sebut dan jelaskan komponen-komponen yang ada di dalam GetX.

GetX adalah framework manajemen state yang populer di Flutter karena kemudahan penggunaan dan performanya. GetX terdiri dari beberapa komponen utama yang memudahkan pengelolaan state, dependensi, dan navigasi dalam aplikasi Flutter. Berikut adalah penjelasan deskriptif tentang komponen-komponen utama di dalam GetX:

- a. State Management

Komponen ini digunakan untuk mengelola state atau data yang berubah selama aplikasi berjalan. GetX menyediakan reaktif state management menggunakan Rx (Reactive Extensions) dan Obx. Dengan pendekatan ini, data dapat didefinisikan sebagai variabel reaktif (misalnya, `RxInt`, `RxString`), dan perubahan data otomatis memperbarui UI yang terkait tanpa memerlukan `setState()`.

- b. Dependency Injection

GetX menyediakan mekanisme untuk mengelola dependensi melalui `Get.put()` dan `Get.lazyPut()`. Ini memungkinkan Anda untuk membuat dan mengakses objek secara global dengan mudah tanpa harus melewati objek secara manual melalui konstruktor. Komponen ini mempermudah pengelolaan controller, service, atau kelas lain yang digunakan di berbagai bagian aplikasi.

c. Routing & Navigation

GetX menyediakan sistem navigasi tanpa memerlukan BuildContext, yang menyederhanakan proses berpindah antar halaman atau rute. Anda dapat menggunakan metode seperti Get.to(), Get.back(), atau Get.offAll() untuk navigasi. Selain itu, GetX mendukung pengelolaan rute dinamis dengan parameter, middleware, dan nested routing.

d. Controller

Controller dalam GetX adalah kelas yang digunakan untuk mengelola logika bisnis dan state. Anda dapat membuat controller dengan mewarisi GetxController atau GetxService (jika ingin menjaga instance tetap hidup di seluruh aplikasi). Controller juga memiliki lifecycle hooks seperti onInit(), onClose(), dan onReady() untuk mengatur logika tertentu pada tahap tertentu.

3. Lengkapilah code di bawah ini, dan tampilkan hasil outputnya serta jelaskan.

```
lib > state.dart > HomePage
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3
4  /// Controller untuk mengelola state counter
5  class CounterController extends GetxController {
6    // Variabel untuk menyimpan nilai counter
7    var counter = 0.obs;
8
9    // Fungsi untuk menambah nilai counter
10   void increment() {
11     counter++;
12   }
13
14   // Fungsi untuk mereset nilai counter
15   void reset() {
16     counter.value = 0;
17   }
18 }
19
20 class HomePage extends StatelessWidget { Constructors for public widgets
21   final CounterController controller = Get.put(CounterController());
22
23   @override
24   Widget build(BuildContext context) {
25     return Scaffold(
26       appBar: AppBar(title: const Text("Counter App")),
```

```

27     body: Center(
28       child: Obx(() {
29         // Menampilkan nilai counter
30         return Text(
31           "${controller.counter}",
32           style: const TextStyle(fontSize: 48),
33         ); // Text
34       }), // Obx
35     ), // Center
36     floatingActionButton: Column(
37       mainAxisAlignment: MainAxisAlignment.end,
38       children: [
39         FloatingActionButton(

```

```

40           onPressed: () {
41             // Menambah nilai counter
42             controller.increment();
43           },
44         child: const Icon(Icons.add),
45       ), // FloatingActionButton
46       const SizedBox(height: 10),
47       FloatingActionButton(
48         onPressed: () {
49           // Mereset nilai counter
50           controller.reset();
51         },
52         child: const Icon(Icons.refresh),
53       ), // FloatingActionButton

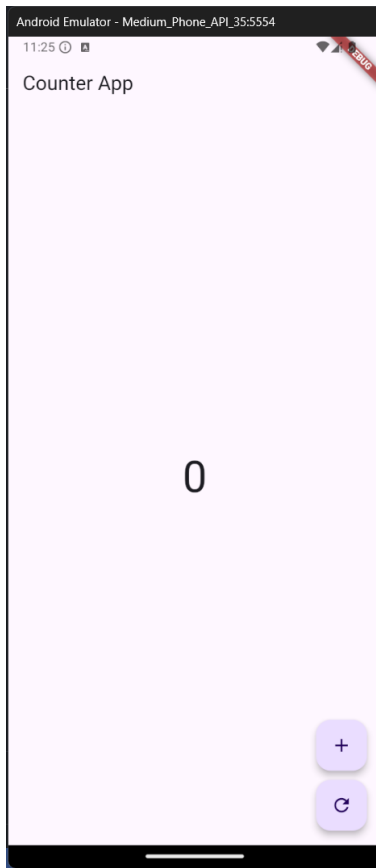
```

```

54     ],
55   ), // Column
56 ); // Scaffold
57 }
58 }
59
60
61

```

Screenshoot Output



Deskripsi Program

Program ini adalah aplikasi Flutter sederhana yang menggunakan GetX untuk mengelola state dari sebuah counter. Aplikasi terdiri dari dua komponen utama: Controller dan UI. Controller (`CounterController`) bertugas mengelola logika bisnis, yaitu menyimpan nilai counter, menambah nilai counter, dan meresetnya. Nilai counter disimpan dalam variabel counter yang bersifat reaktif (`.obs`), sehingga UI akan otomatis diperbarui ketika nilai ini berubah.

Di dalam UI, nilai counter ditampilkan menggunakan widget `Text` yang dibungkus oleh `Obx`. Widget ini memantau perubahan pada counter di controller. Ada dua tombol aksi (`FloatingActionButton`): satu untuk menambah nilai counter dengan memanggil fungsi `increment()`, dan satu lagi untuk mereset nilai counter ke 0 dengan fungsi `reset()`. Dengan menggunakan GetX, navigasi state menjadi lebih mudah tanpa perlu menyentuh `setState()`.