

**TUGAS PENDAHULUAN  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XIII  
NETWORKING**



**Disusun Oleh :  
Nita Fitrotul Ma'ah  
2211104005  
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**

## **TUGAS PENDAHULUAN**

### **SOAL**

1. Apa yang dimaksud dengan state management pada Flutter?

Jawab:

State Management adalah pengelolaan data atau status aplikasi, seperti nilai variabel, yang berubah selama siklus hidup aplikasi. Dalam Flutter, state management memastikan bahwa perubahan data (state) yang terjadi dapat memicu pembaruan pada tampilan aplikasi (UI) secara efisien. State management digunakan untuk mengelola komunikasi antara komponen UI dan data, baik secara lokal (pada satu widget) maupun global (lintas widget atau halaman).

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

Jawab:

GetX adalah library state management yang sederhana, cepat, dan efisien di Flutter. Komponen utamanya meliputi:

- a. State Management

Mengelola state aplikasi dengan cara reaktif, menggunakan Rx untuk membuat variabel menjadi observabel dan secara otomatis memperbarui UI saat nilainya berubah.

- b. Dependency Injection

Mengatur dependensi atau objek di seluruh aplikasi dengan cara yang efisien menggunakan Get.put, Get.lazyPut, atau Get.find.

- c. Routing Management

Mempermudah navigasi antar halaman tanpa memerlukan BuildContext, menggunakan Get.to, Get.off, dan Get.back.

- d. Utilities

Menyediakan fitur tambahan seperti snackbar (Get.snackbar), dialog (Get.dialog), dan bottom sheet (Get.bottomSheet).

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

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

// Controller untuk mengelola state counter
class CounterController extends GetxController {
  // Variabel untuk menyimpan nilai counter
  var counter = 0.obs;

  // Fungsi untuk menambah nilai counter
  void increment() {
    counter++;
  }

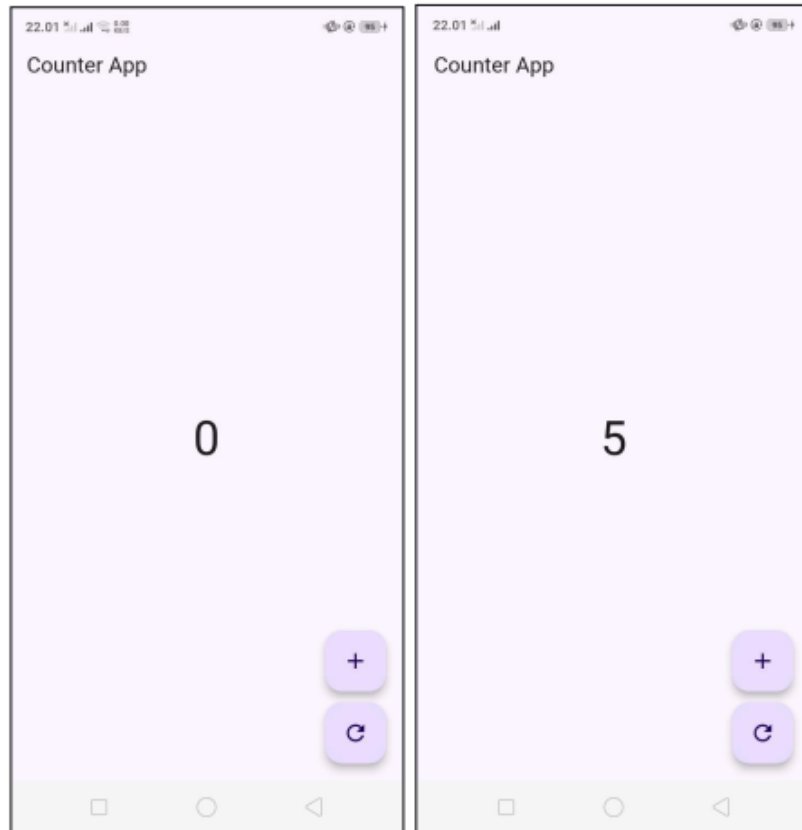
  // Fungsi untuk mereset nilai counter
  void reset() {
    counter.value = 0;
  }
}

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Counter App")),
      body: Center(
        child: Obx(() {
          // Menampilkan nilai counter secara reaktif
          return Text(
            "${controller.counter}",
            style: TextStyle(fontSize: 48),
          );
        })
      ),
      floatingActionButton: Column(
        mainAxisAlignment: MainAxisAlignment.end,
        children: [
          FloatingActionButton(
            onPressed: () {
              // Menambah nilai counter
              controller.increment();
            },
            child: Icon(Icons.add),
          ),
          SizedBox(height: 10),
          FloatingActionButton(
            onPressed: () {
              // Mereset nilai counter
              controller.reset();
            },
            child: Icon(Icons.refresh),
          ),
        ],
      ),
    );
  }
}

void main() {
  runApp(MaterialApp(
    debugShowCheckedModeBanner: false,
    home: HomePage(),
  ));
}
```

## Screenshoot Output



## Deskripsi Program

Program ini adalah aplikasi Flutter sederhana yang menggunakan GetX untuk mengelola state. Program ini memiliki satu layar utama yang menampilkan nilai counter secara dinamis.

1. Nilai counter dikelola oleh CounterController dengan variabel reaktif counter.
2. Saat tombol tambah ditekan, fungsi increment akan menambah nilai counter dan memperbarui UI secara otomatis.
3. Saat tombol reset ditekan, fungsi reset akan mengembalikan nilai counter ke 0 dan memperbarui UI.
4. Fungsi reaktif dari GetX (Obx) memungkinkan pembaruan UI secara langsung saat nilai counter berubah tanpa harus memanggil setState.
5. Dengan pendekatan ini, program menjadi lebih sederhana dan efisien dalam hal state management.