

**TUGAS PENDAHULUAN  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XIII  
NETWORKING**



**Disusun Oleh :  
Muhammad Agam Nasywaan / 2211104085  
SE-06-1**

**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 proses mengelola data yang dinamis (state) dalam aplikasi mobile. State adalah data yang dapat berubah-ubah, seperti informasi pengguna, data dari API, atau hasil operasi pengguna.

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

**Jawab :**

1. State Management untuk mengelola dan memantau state aplikasi menggunakan reaktivitas
2. Dependency Injection untuk mengelola dependensi secara otomatis sehingga tidak perlu membuat instance secara manual
3. Route Management, menyediakan navigasi yang lebih sederhana tanpa memerlukan atau bergantung pada konteks widget
4. Controller sebagai penghubung antara model dan view serta bertanggung jawab untuk menangani logika bisnis dan mengelola state aplikasi
5. GetBuilder digunakan untuk membangun UI berdasarkan perubahan state dengan cara yang lebih sederhana dibandingkan dengan penggunaan StreamBuilder atau ChangeNotifier.

3. Berikut code lengkap yang sudah ditambahkan.

```
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 count = 0.obs;

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

  // Fungsi untuk mereset nilai counter
  void reset() {
    count.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
          return Text(
            "${controller.count}", // Mengganti dengan
            nilai counter
            style: TextStyle(fontSize: 48),
          );
        }),
      ),
      floatingActionButton: Column(
        mainAxisAlignment: MainAxisAlignment.end,
        children: [
```

```

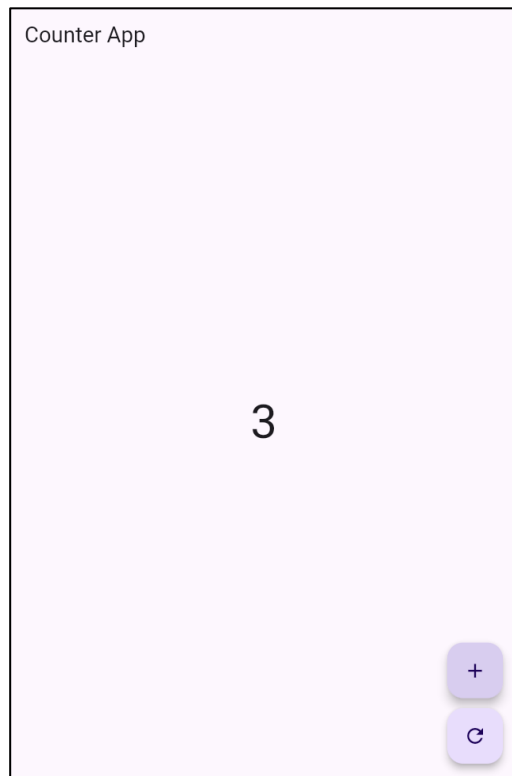
        children: [
          FloatingActionButton(
            onPressed: () {
              // Menambah nilai counter
              controller.increment();
            },
            child: Icon(Icons.add),
          ),
          SizedBox(height: 10),
          FloatingActionButton(
            onPressed: () {
              // Mereset nilai counter
              controller.reset();
            },
            child: Icon(Icons.refresh),
          ),
        ],
      ),
    );
  }

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

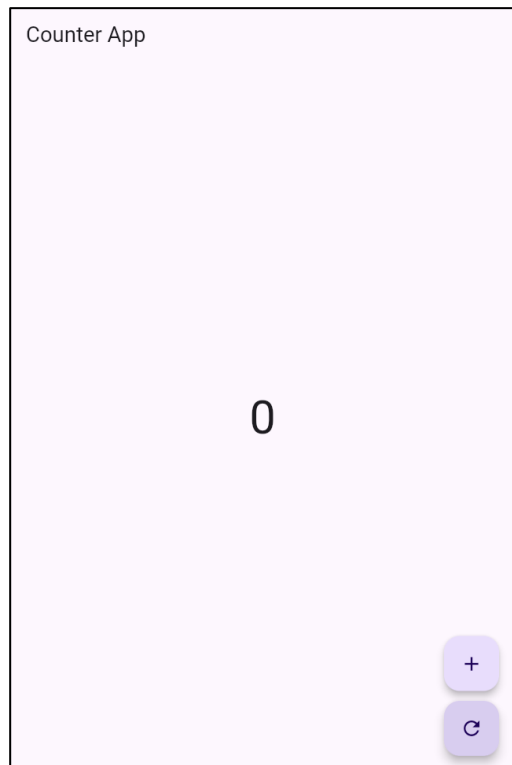
```

## Screenshoot Output

Ketika melakukan increment counter



Ketika melakukan reset counter



### **Deskripsi Program**

Kode di atas adalah implementasi sederhana dari aplikasi Counter menggunakan Flutter dan GetX untuk mengelola state. Bagian pertama adalah kelas CounterController, yang bertugas mengelola logika aplikasi, termasuk variabel reaktif count yang akan otomatis memperbarui tampilan setiap kali nilainya berubah. Terdapat dua fungsi di dalamnya, yaitu increment() untuk menambah nilai count dan reset() untuk mengatur ulang nilai count ke nol. Pada kelas MyApp, kontroler CounterController diinisialisasi dengan Get.put() agar bisa digunakan di seluruh aplikasi tanpa perlu menginisialisasi ulang. Antarmuka aplikasi terdiri dari sebuah Scaffold dengan AppBar yang menampilkan judul, body berupa teks nilai counter yang dibungkus dalam Obx agar nilai selalu sinkron dengan perubahan. Terdapat dua tombol aksi di floatingActionButton, yaitu tombol tambah untuk memanggil fungsi increment() dan tombol refresh untuk memanggil fungsi reset().