

**TUGAS PENDAHULUAN  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XIII  
NETWORKING**



**Disusun Oleh :  
Dimas Cahyo Margono / 2211104060  
SE-06-02**

**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?
2. Sebut dan jelaskan komponen-komponen yang ada di dalam GetX.
3. Lengkapilah 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 {
  // TODO: Tambahkan variabel untuk menyimpan nilai
  counter
  // TODO: Buat fungsi untuk menambah nilai counter
  // TODO: Buat fungsi untuk mereset nilai counter
}
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(() {
          // TODO: Lengkapi logika untuk menampilkan nilai
          counter
          return Text(
            "0", // Ganti ini dengan nilai counter
            style: TextStyle(fontSize: 48),
          );
        }),
      ),
    ),
```

```

floatingActionButton: Column(
  mainAxisAlignment: MainAxisAlignment.end,
  children: [
    FloatingActionButton(
      onPressed: () {
        // TODO: Tambahkan logika untuk menambah nilai counter
      },
      child: Icon(Icons.add),
    ),
    SizedBox(height: 10),
    FloatingActionButton(
      onPressed: () {
        // TODO: Tambahkan logika untuk mereset nilai counter
      },
      child: Icon(Icons.refresh),
    ),
  ],
);
}

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

```

## JAWABAN

1. State management dalam Flutter adalah cara untuk mengelola dan mengontrol data atau state dalam aplikasi sehingga dapat memperbarui antarmuka pengguna secara efisien saat data berubah. Dalam Flutter, state mengacu pada data yang dapat berubah

selama masa hidup aplikasi, seperti input pengguna, respons API, atau data dinamis lainnya. Dengan state management, aplikasi dapat memastikan bahwa perubahan data akan tercermin langsung pada UI tanpa harus membuat ulang seluruh struktur widget. Ada berbagai metode state management di Flutter, seperti bawaan (setState), serta menggunakan paket pihak ketiga seperti Provider, GetX, Bloc, atau Riverpod, yang membantu mengelola state dengan lebih terorganisir, terutama untuk aplikasi kompleks.

2. GetX adalah framework Flutter untuk state management, dependency injection, dan route management. Berikut komponen utamanya:

**1. State Management:**

- Mengelola perubahan data secara reaktif dengan tipe data reaktif seperti Rx<T>.
- Contoh: `var counter = 0.obs`; yang memungkinkan UI diperbarui secara otomatis saat nilai berubah.

**2. Dependency Injection (DI):**

- Mempermudah pembuatan dan pengelolaan instance class dengan cara menginjeksikannya menggunakan `Get.put()` atau `Get.lazyPut()`.
- Contoh: `final controller = Get.put(MyController());` untuk menggunakan instance controller.

**3. Route Management:**

- Mengelola navigasi antara halaman tanpa memerlukan konteks menggunakan `Get.to()`, `Get.back()`, dan `Get.offAll()`.
- Contoh: `Get.to(NextPage());` untuk berpindah ke halaman lain.

**4. Snackbar, Dialog, BottomSheet:**

- Mempermudah menampilkan notifikasi atau UI interaktif seperti dialog dan bottom sheet dengan metode bawaan `Get.snackbar()`, `Get.defaultDialog()`, dan `Get.bottomSheet()`.

**5. Reaktivitas:**

- GetX menyediakan widget reaktif seperti `Obx` dan `GetBuilder` untuk memperbarui UI hanya pada bagian yang relevan saat state berubah.

3.

```
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(
```

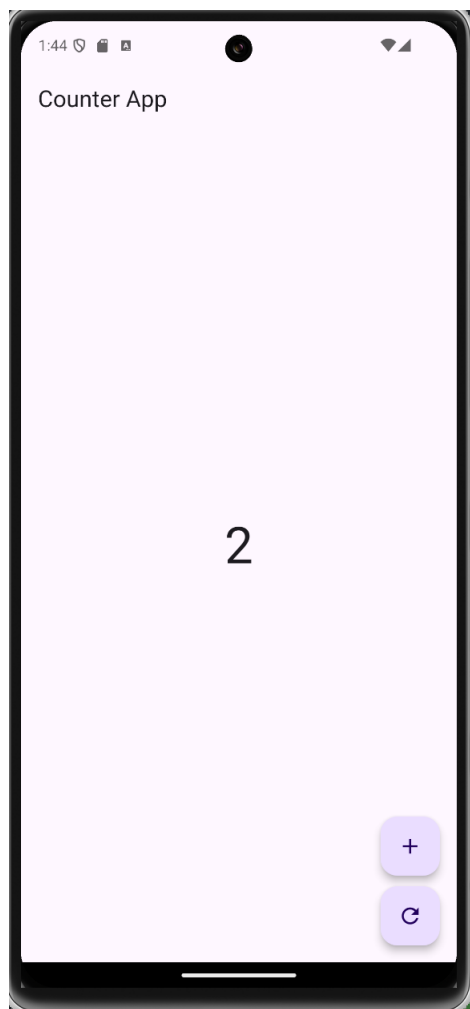
```

        child: Obx(() {
          // Menampilkan nilai counter
          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(),  
  ));  
}
```

### Screenshot Output



## **Deskripsi Program**

Program ini adalah aplikasi sederhana berbasis Flutter menggunakan GetX untuk state management. Aplikasi terdiri dari CounterController, yang mengelola nilai counter menggunakan variabel counter bertipe RxInt untuk mendukung reaktivitas. Tombol "+" digunakan untuk meningkatkan nilai counter melalui fungsi increment, sedangkan tombol refresh mereset nilai counter ke nol dengan fungsi reset. Komponen Obx pada HomePage digunakan untuk mendeteksi perubahan nilai counter secara otomatis dan memperbarui tampilan tanpa memanggil setState. Aplikasi ini menggunakan widget FloatingActionButton untuk kontrol interaktif, dan nilai counter ditampilkan secara dinamis di tengah layar dalam format teks besar. Program ini memperlihatkan penggunaan GetX yang efisien untuk mengelola perubahan state secara real-time.