

**TUGAS PENDAHULUAN  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XIII  
NETWORKING**



**Disusun Oleh :**

**Ganesha Rahman Gibran / 221110458**

**S1SE-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?

Cara untuk mengelola dan menjaga konsistensi data atau *state* aplikasi, terutama ketika data berubah dan perlu diperbarui di UI. Dalam Flutter, state management sangat penting karena aplikasi berbasis widget membutuhkan cara yang efisien untuk menghubungkan data dengan tampilan.

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

- a. State Management

- Reactive State : Menggunakan Obx dan variabel reaktif (Rx) untuk mendengarkan perubahan data secara otomatis.
- Simple State : Menggunakan GetBuilder untuk memicu pembaruan UI secara manual.

- b. Dependency Injection (DI)

- Memungkinkan pengelolaan instans kelas secara efisien dengan Get.put(), Get.lazyPut(), dan Get.find().

- c. Route Management

- Menyediakan cara mudah untuk navigasi tanpa BuildContext dengan Get.to(), Get.off(), dll

- d. Controller

- Merupakan kelas yang mengelola logika bisnis dan state, biasanya diperluas dari GetxController.

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(),
  ));
}

```

### Perbaiki code

```

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

// Controller untuk mengelola state counter
class CounterController extends GetxController {
  // Variabel reaktif 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 {
  @override
  Widget build(BuildContext context) {

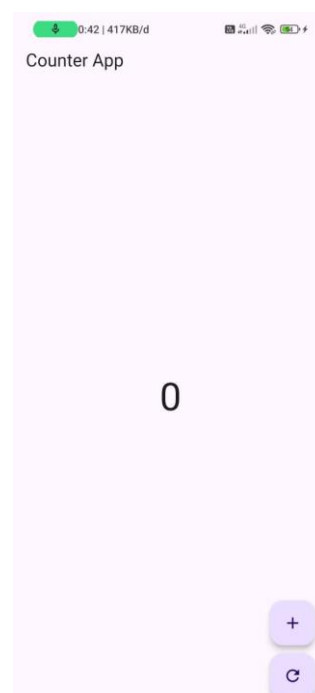
```

```
// Inisialisasi controller menggunakan Get.put() di dalam build
final CounterController controller = Get.put(CounterController());

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

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

## Screenshoot Output



## **Deskripsi Program**

Aplikasi penghitung sederhana berbasis Flutter dengan menggunakan framework GetX untuk manajemen state. Aplikasi ini memungkinkan pengguna untuk meningkatkan nilai penghitung (counter) dengan menekan tombol tambah (+) dan mengatur ulang nilai penghitung ke nol dengan menekan tombol reset. Algoritma utama program menggunakan konsep reaktivitas dari GetX, di mana nilai variabel counter dibungkus dalam objek reaktif (obs). Setiap kali nilai counter berubah, tampilan akan secara otomatis diperbarui tanpa perlu menyegarkan halaman secara manual. Output aplikasi ditampilkan dalam bentuk angka besar di layar, yang menunjukkan nilai penghitung saat ini, serta dua tombol aksi di bagian bawah untuk mengontrol penghitung.