

**TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XIII
NETWORKING**



Disusun Oleh :

Akmal AdiCandra /

2211104059 SE06B

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 pada Flutter adalah cara untuk mengelola dan mengontrol data yang berubah dalam aplikasi. Flutter menyediakan berbagai metode dan pustaka untuk menangani perubahan data (state) di aplikasi, sehingga tampilan UI dapat disesuaikan secara otomatis ketika data berubah. Dengan menggunakan state management, pengelolaan state menjadi lebih terstruktur, efisien, dan mudah dikelola, terutama pada aplikasi yang kompleks dengan banyak interaksi pengguna. Beberapa pustaka yang umum digunakan untuk state management di Flutter adalah Provider, Riverpod, Bloc, dan GetX.

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

GetX adalah pustaka state management yang ringan, cepat, dan mudah digunakan di Flutter. Komponen-komponen utama dalam GetX adalah:

- **Controller:** Komponen ini digunakan untuk mengelola state dan logika bisnis. Controller akan menyimpan data dan menyediakan fungsi-fungsi untuk mengubah state, misalnya penambahan atau penghapusan nilai.
- **Rx (Reactive Programming):** GetX menggunakan konsep reaktif untuk memantau perubahan nilai variabel. Ketika nilai variabel berubah, UI yang terkait dengan variabel tersebut akan otomatis terupdate tanpa perlu dipanggil ulang secara manual.
- **Obx:** Widget ini digunakan untuk membungkus widget lain yang bergantung pada nilai reaktif. Ketika nilai yang dipantau berubah, widget yang dibungkus dengan Obx akan di-rebuild otomatis.
- **Get.put():** Fungsi ini digunakan untuk mendaftarkan controller atau objek yang ingin dikelola oleh GetX dan memungkinkan akses ke controller tersebut di berbagai tempat dalam aplikasi.
- **GetX:** Merupakan widget yang menyediakan cara alternatif untuk mengelola state tanpa membutuhkan StatefulWidget dan memberikan pengelolaan state secara otomatis.

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(),
  ));
}

```

KODE YANG DI PERBAIKI :

```

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; // .obs membuat counter menjadi reaktif

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

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

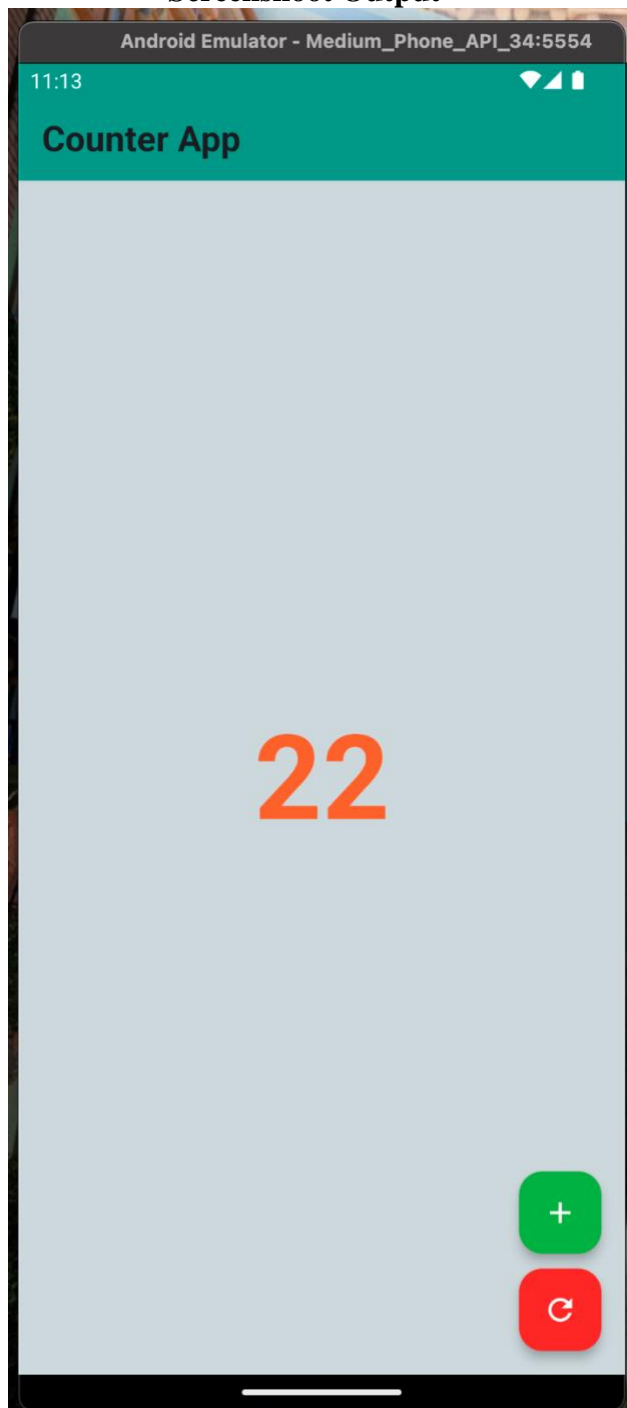
class HomePage extends StatelessWidget {
  final CounterController controller = Get.put(CounterController());

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blueGrey.shade100, // Latar belakang yang
cerah appBar: AppBar(
      title: Text("Counter App",
        style: TextStyle(fontWeight: FontWeight.bold, fontSize: 24)),
      backgroundColor: Colors.teal, // Warna app bar yang menarik
    ),
    body: Center(
      child: Obx(() {
        // Animasi dengan perubahan nilai counter
        return AnimatedSwitcher(
          duration: Duration(milliseconds: 300),
          child: Text(
            "${controller.counter}",
            key: ValueKey<int>({
              controller.counter.value, // Key untuk animasi perubahan
            }),
            style: TextStyle(
              fontSize: 80,
              fontWeight: FontWeight.bold,
              color: Colors.deepOrangeAccent,
            ),
          ),
        );
      })),
    floatingActionButton: Column(
      mainAxisAlignment: MainAxisAlignment.end,
      children: [
        FloatingActionButton(
          onPressed: () {
            controller.increment(); // Menambah nilai counter
          },
          child: Icon(Icons.add, color: Colors.white),
          backgroundColor: Colors.green, // Warna tombol tambah
        ),
        SizedBox(height: 10),
        FloatingActionButton(
          onPressed: () {
            controller.reset(); // Mereset nilai counter
          },
          child: Icon(Icons.refresh, color: Colors.white),
          backgroundColor: Colors.red, // Warna tombol reset
        ),
      ],
    ),
  );
}

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

```

Screenshoot Output



Deskripsi Program

Program ini adalah aplikasi Flutter sederhana yang berfungsi untuk menghitung nilai counter. Aplikasi menggunakan GetX untuk manajemen state yang memungkinkan nilai counter berubah secara otomatis di UI. Ketika tombol "+" ditekan, nilai counter akan bertambah satu, dan ketika tombol refresh ditekan, nilai counter akan kembali ke 0. Fungsi `increment()` digunakan untuk menambah nilai counter, sementara `reset()` digunakan untuk mengatur nilai counter kembali ke 0. State counter dikelola oleh `CounterController`, dan nilai

counter yang terbaru ditampilkan menggunakan widget Obx, yang memungkinkan UI memperbarui tampilan secara otomatis setiap kali nilai counter berubah.