

**TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XIII
NETWORKING**



Disusun Oleh :

Muhammad Ralfi – 2211104054

SE-06-2

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 dalam flutter adalah proses mengelola data yang memengaruhi tampilan dan logika aplikasi, state management membantu pengembang dan desainer membuat aplikasi yang lebih dinamis sesuai dengan kebutuhan pengguna. Pada flutter ada dua tipe state yaitu stateless dan stateful, penggunaan keduanya bergantung pada kebutuhan.

Stateless Widget

Ketika menggunakan beberapa widget flutter yang memiliki data tidak berubah-ubah atau immutable, maka stateless lebih tepat digunakan, contohnya halaman contact us, pricing dan lain sebagainya.

Statefull Widget

Tidak semua aplikasi yang akan dibangun hanya menggunakan data statis, terkadang juga membutuhkan data dinamis, maka stateful widget lebih tepat penggunaannya.

2. Sebut dan jelaskan komponen-komponen yang ada di dalam GetX.
 - a. Reactive state, menggunakan class untuk membuat variabel reaktif
 - b. Simple state management bisa menggunakan GetBuilder
 - c. Mixin state management bisa menggunakan GetxController
 - d. Routing menggunakan Get.to, Get.off, Get.offAll
 - e. Dependency, seperti lazy put
 - f. Snackbar, menggunakan Get.snackbar
 - g. Dialog, Get.defaultDialog
 - h. BottomSheet, Get.bottomSheet
 - i. Bindings digunakan untuk mengatur dependency pada setiap route.

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 {
  // TODO: Tambahkan variabel untuk menyimpan nilai counter
  final count = 0.obs;
  Int get counter => count.value;

  // TODO: Buat fungsi untuk menambah nilai counter
  void increment() {
    count.value++;
  }

  // TODO: Buat 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(() {
          // TODO: Lengkapi logika untuk menampilkan nilai
          counter

          return Text(
            "${controller.counter}", // Ganti ini
            dengan nilai counter style:
            TextStyle(fontSize: 48),
          );
        })
      )
    );
  }
}
```

```

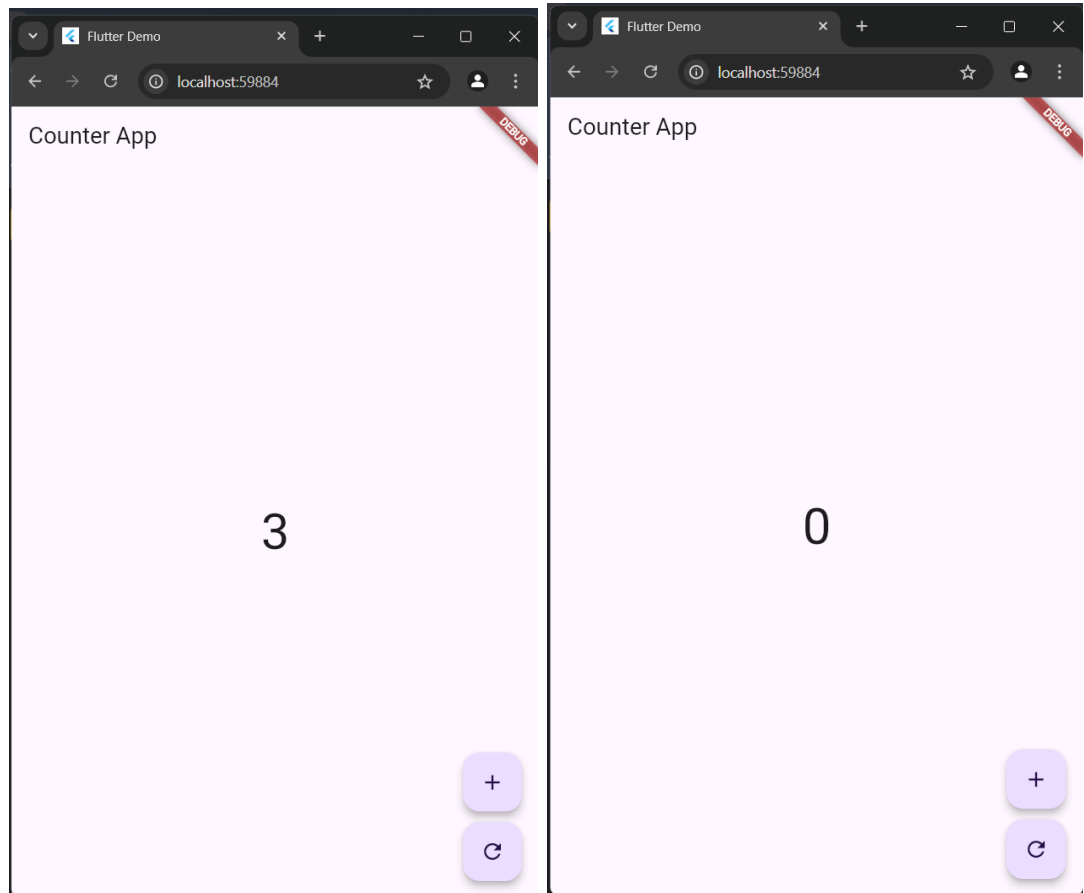
        mainAxisAlignment: MainAxisAlignment.end,
        children: [
          FloatingActionButton(
            onPressed: () {
              controller.increment(); // TODO: Tambahkan
              logika untuk menambah nilai
counter
            },
            child: Icon(Icons.add),
          ),
       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

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

```

Screenshoot Output

(lampirkan bukti screenshoot output dari sourcecode)



Deskripsi Program

Program ini adalah sebuah aplikasi Flutter yang menggunakan **GetX** untuk mengelola state management pada fitur counter. Di dalam program, terdapat sebuah class bernama **CounterController** yang berfungsi untuk mengelola logika terkait nilai counter, termasuk menyimpan nilai counter, menambah nilai, dan meresetnya kembali ke nol. Class ini memanfaatkan reaktivitas dari **GetX** dengan menggunakan variabel **obs** agar setiap perubahan nilai counter otomatis tercermin di UI.

Halaman utama aplikasi, yang diimplementasikan dalam class **HomePage**, memiliki sebuah UI sederhana dengan teks yang menampilkan nilai counter dan dua tombol aksi. Tombol pertama, dengan ikon tambah, digunakan untuk menambahkan nilai counter, sedangkan tombol kedua, dengan ikon refresh, digunakan untuk mereset nilai counter ke nol. Komponen **Obx** memastikan bahwa perubahan pada nilai counter yang dikelola oleh controller langsung di-update di layar tanpa memerlukan kode tambahan.