

**TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XIII
NETWORKING**



Disusun Oleh :

Aji Prasetyo Nugroho / 2211104049

S1SE-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 adalah cara untuk mengelola data atau "state" dari aplikasi Flutter agar aplikasi dapat merespons perubahan data secara dinamis. Dalam Flutter, "state" mengacu pada data atau informasi yang memengaruhi bagaimana UI ditampilkan. State management memungkinkan pengembang memisahkan logika aplikasi dari tampilan, sehingga aplikasi lebih terorganisasi dan mudah untuk dikelola. Ada berbagai pendekatan state management, seperti menggunakan Provider, Bloc, Riverpod, atau GetX.

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

GetX adalah framework state management dan dependency injection yang sederhana, namun sangat powerful. Komponen utama GetX adalah:

- **State Management:** GetX menawarkan cara reaktif untuk mengelola state menggunakan Obx dan Rx (reactive variables). Dengan ini, UI akan diperbarui secara otomatis ketika state berubah.
- **Route Management:** GetX menyediakan navigasi yang mudah tanpa memerlukan context. Fungsi seperti `Get.to()`, `Get.back()`, dan `Get.off()` mempermudah navigasi antar halaman.
- **Dependency Injection:** GetX memungkinkan pengelolaan dependensi menggunakan metode seperti `Get.put()` untuk singleton dan `Get.lazyPut()` untuk lazy loading objek.
- **Internationalization:** Mendukung penerjemahan aplikasi ke berbagai bahasa secara langsung dengan GetX.

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
  var counter = 0.obs;

  // TODO: Buat fungsi untuk menambah nilai counter
  void increment() => counter++;

  // TODO: Buat 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(() {
          // TODO: Lengkapi logika untuk menampilkan nilai
counter
          return Text(
            "${controller.counter.value}",
            style: TextStyle(fontSize: 48),
          );
        }),
      ),
      floatingActionButton: Column(
```

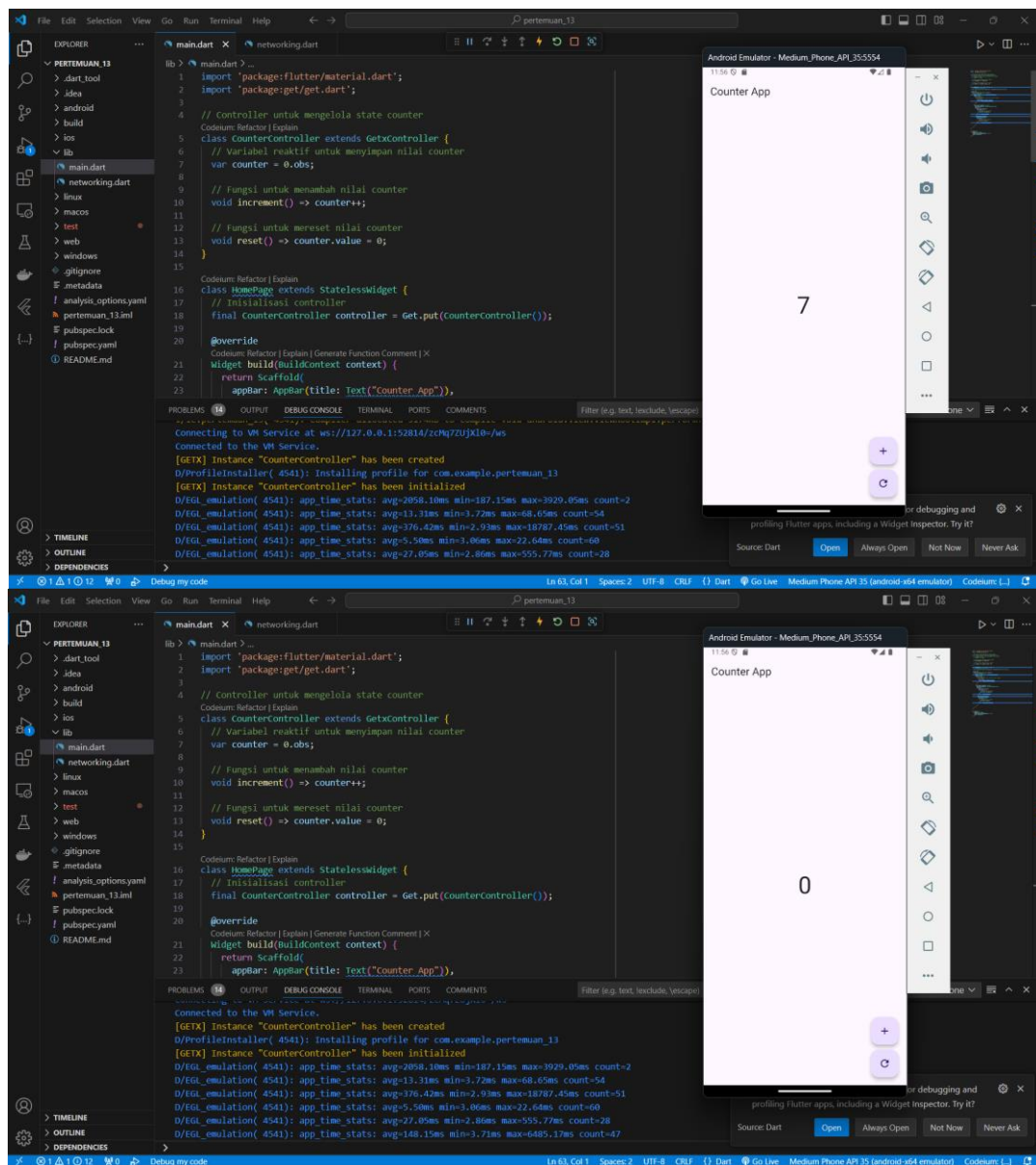
```

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

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

```

Screenshoot Output



Deskripsi Program

Program di atas adalah aplikasi Flutter sederhana bernama "Counter App" yang menggunakan GetX sebagai framework state management. Aplikasi ini memiliki fitur untuk mengelola nilai counter (penghitung) secara reaktif. Dengan bantuan CounterController, nilai counter disimpan dalam variabel reaktif (RxInt) yang memungkinkan UI diperbarui secara otomatis saat nilai berubah. Pada tampilan utama, nilai counter ditampilkan di tengah layar, dan pengguna dapat menambah nilai counter menggunakan tombol "+" atau mereset nilai counter ke 0 menggunakan tombol refresh. Semua interaksi ini ditangani secara reaktif menggunakan widget Obx, yang memastikan bahwa perubahan state pada controller langsung terlihat di antarmuka pengguna tanpa kode tambahan untuk pembaruan UI manual.