

**TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XIII
NETWORKING**



Disusun Oleh :

Dwi Candra Pratama / 2211104035

SE06-02

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?

State management adalah proses pengelolaan data atau status aplikasi yang dapat berubah selama siklus hidup aplikasi. Pada Flutter, state management bertujuan untuk memastikan bahwa UI tetap sinkron dengan data (state) yang diwakilinya.

Flutter menyediakan berbagai metode untuk mengelola state, baik bawaan (seperti `setState`) maupun menggunakan paket pihak ketiga seperti Provider, GetX, Bloc, dan Riverpod.

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

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

1. State Management:

Digunakan untuk mengelola data (state) secara reaktif menggunakan Rx dan widget Obx.

Contoh: `RxInt`, `RxString`, `RxList`.

2. Dependency Injection:

- Mengelola dependency aplikasi dengan cara yang efisien menggunakan `Get.put()`, `Get.lazyPut()`, atau `Get.find()`.
- Dependency Injection memungkinkan pengelolaan objek atau controller secara global tanpa boilerplate code.

3. Route Management:

- Mengelola navigasi antar halaman tanpa memerlukan konteks.
- Menggunakan metode seperti `Get.to()`, `Get.back()`, atau `Get.off()`.

4. Reactive Programming:

Menggunakan pendekatan reaktif untuk memantau perubahan pada variabel yang dibungkus dalam Rx.

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),
    ),
  ],
),
);
}
}

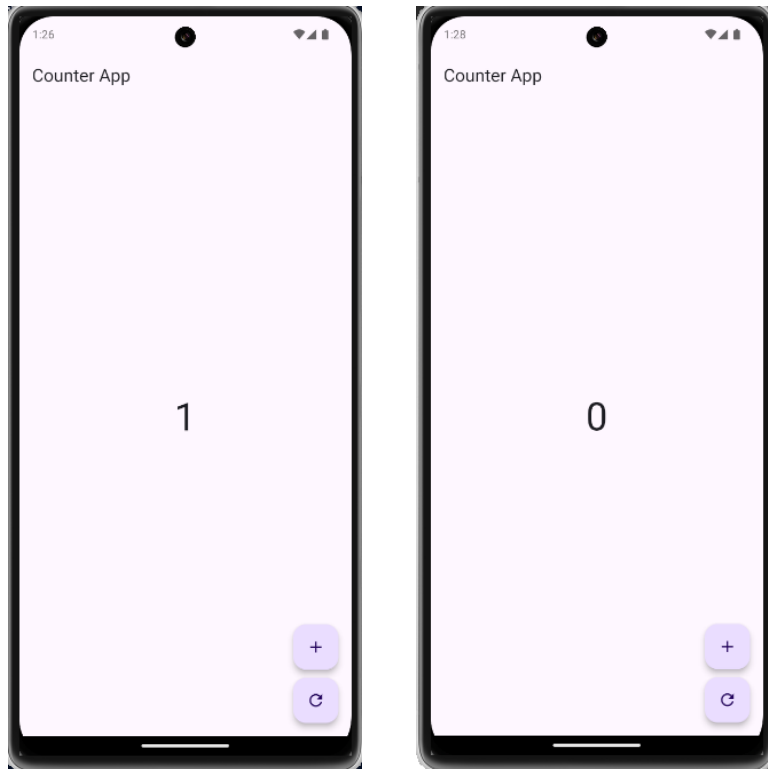
void main() {
runApp(MaterialApp(
    debugShowCheckedModeBanner: false,
home: HomePage(),
));
}

```

Screenshoot Output

(lampirkan bukti screenshot output dari sourcecode)

Gambar ketika klik + dan setelah refresh



Deskripsi Program

Aplikasi **Counter App** ini dibuat menggunakan Flutter dengan bantuan framework **GetX** untuk mengelola state secara reaktif. Fitur utamanya adalah menambah dan mereset nilai angka (counter) yang ditampilkan di layar. State counter disimpan dalam bentuk **RxInt** di dalam sebuah controller, sehingga perubahan nilainya secara otomatis diperbarui di UI menggunakan widget **Obx**. Tombol tambah (+) memanggil fungsi **increment()** untuk meningkatkan nilai counter, sementara tombol reset menggunakan fungsi **reset()** untuk mengembalikan nilainya ke nol. Dengan pendekatan reaktif ini, setiap perubahan pada counter langsung terlihat tanpa perlu proses pemanggilan ulang secara manual.