

**TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK
MODUL XIII
NETWORKING**



**Disusun Oleh :
Ahmad Junaidi / 2211104002
SE-06-01**

**Asisten Praktikum :
Ayu Susilowati
Noviana Rizki Anisa Putri**

**Dosen Pengampu :
Yudha Islami Sulistya, S.Kom., M.Cs.**

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan state management pada Flutter?

Jawab:

State Management pada Flutter adalah sebuah pendekatan atau teknik yang digunakan untuk mengelola state (data atau informasi yang dapat berubah selama aplikasi berjalan) pada aplikasi Flutter. State ini dapat berupa data sederhana seperti nilai variabel atau data kompleks seperti daftar item. Pada Flutter, state management sangat penting karena memastikan bahwa perubahan data ditampilkan dengan benar pada antarmuka pengguna (UI) tanpa harus membangun ulang seluruh aplikasi.

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

Jawab:

Komponen-komponen dalam GetX:

1. State Management: Mengelola state aplikasi secara reaktif menggunakan Obx, GetBuilder, atau GetX. Komponen ini memudahkan sinkronisasi antara data dan tampilan tanpa boilerplate yang rumit.
 2. Navigation: Mengelola navigasi halaman tanpa memerlukan Navigator bawaan Flutter. Fitur ini mendukung navigasi tanpa konteks menggunakan fungsi `Get.to()`, `Get.off()`, atau `Get.back()`.
 3. Dependency Injection: Mengelola dependensi dan membuat controller atau objek secara instan menggunakan `Get.put()` atau `Get.lazyPut()`. Ini membantu menjaga codebase lebih modular dan efisien.
 4. Snackbar/Dialog/BottomSheet: Komponen bawaan untuk menampilkan pesan (snackbar), dialog, atau bottom sheet tanpa harus menggunakan `BuildContext`.
3. Lengkapilah code di bawah ini, dan tampilkan hasil outputnya serta jelaskan.

Jawab:

```

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;

  // 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());

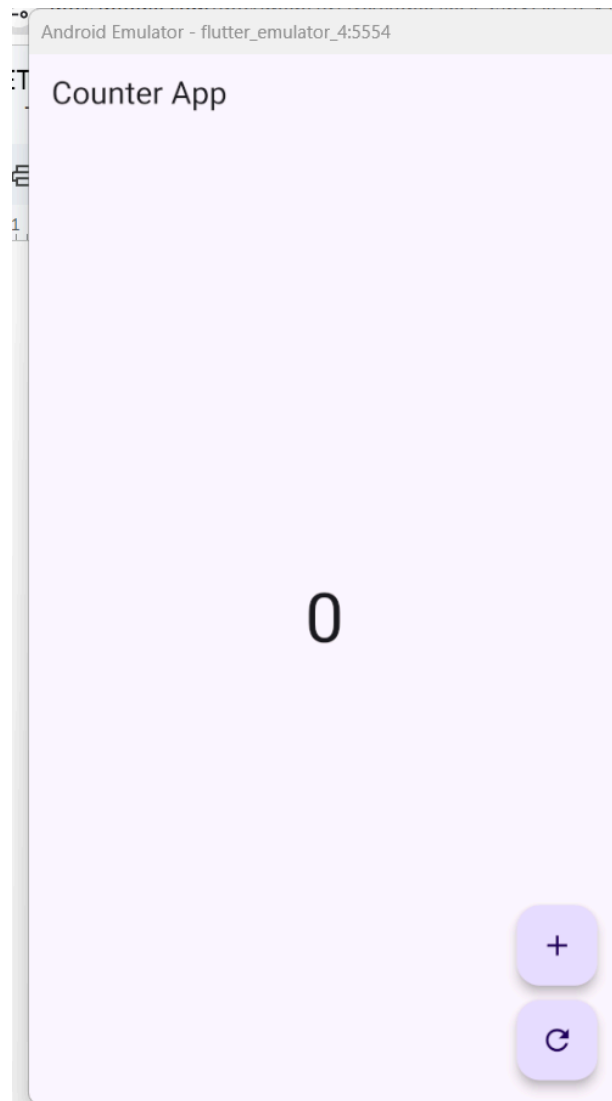
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Counter App")),
      body: Center(
        child: Obx(() {
          // Menampilkan nilai counter
          return Text(
            "${controller.counter}",
            style: TextStyle(fontSize: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

```

```
        controller.reset();
    },
    child: Icon(Icons.refresh),
  ),
],
),
);
}
}

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

Screenshoot Output :



Deskripsi Program:

Program ini adalah sebuah aplikasi sederhana bernama Counter App yang berfungsi sebagai penghitung (counter). Aplikasi menggunakan framework Flutter dengan bantuan pustaka GetX untuk mengelola state secara reaktif. Program dimulai dengan menampilkan angka counter pada layar, yang dapat diubah menggunakan dua tombol:

- Tombol tambah (+): Menambah nilai counter sebesar 1 setiap kali ditekan.
- Tombol reset (↺): Mengembalikan nilai counter ke 0.

Algoritma program bekerja dengan memanfaatkan GetxController yang menyimpan data state dari nilai counter. Perubahan nilai counter secara otomatis diperbarui pada UI menggunakan widget Obx. Program ini

memanfaatkan dependency injection dengan `Get.put()` untuk memastikan controller dapat diakses di mana saja selama aplikasi berjalan. Output berupa angka dinamis akan langsung ditampilkan pada layar setelah interaksi pengguna.