

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



Disusun Oleh :
Mohammad Dhimas Afrizal / 2211104023
SE-06-01

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

A. SOAL NOMOR 1

Apa yang dimaksud dengan state management pada Flutter?

Jawab:

State management adalah cara mengelola dan menyimpan data/state aplikasi agar dapat diakses dan diperbarui dengan mudah di berbagai bagian aplikasi. Dalam Flutter, state management membantu mengontrol perubahan data dan memperbarui UI secara efisien tanpa harus merender ulang seluruh widget.

B. SOAL NOMOR 2

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

Jawab:

- `GetxController`: Kelas dasar untuk mengelola state dan logika bisnis
- `Obx`: Widget reaktif untuk membuat UI yang dapat berubah otomatis saat state berubah
- `Get.put()`: Metode untuk mendaftarkan controller
- `Get.find()`: Metode untuk menemukan dan mengakses controller yang sudah didaftarkan
- `Reactive Variables`: Variabel yang dapat mengupdate UI secara otomatis saat nilainya berubah

C. SOAL NOMOR 3

Lengkapilah code di bawah ini, dan tampilkan hasil outputnya serta jelaskan

Sourcecode

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';

/// Controller untuk mengelola state counter
class CounterController extends GetxController
{
  // Variabel untuk menyimpan nilai counter
  dengan tipe reactive
  final RxInt counter = 0.obs;

  // Fungsi untuk menambah nilai counter
  void incrementCounter() {
    counter.value++;
  }

  // Fungsi untuk mereset nilai counter
  void resetCounter() {
    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.value}",
            style: TextStyle(fontSize: 48),
          );
        })
      ),
      floatingActionButton: Column(
        mainAxisAlignment:
MainAxisAlignment.end,
        children: [
          FloatingActionButton(
            onPressed: () {
              // Menambah nilai counter
              controller.incrementCounter();
            },
            child: Icon(Icons.add),
          ),
          SizedBox(height: 10),
          FloatingActionButton(
            onPressed: () {
              // Mereset nilai counter
              controller.resetCounter();
            },
            child: Icon(Icons.refresh),
          ),
        ],
      ),
    );
  }
}

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

```
});  
}
```

Screenshoot Output

Tampilan awal > click (+) button (6x) > click restart button (back to 0)



Deskripsi Program

1. Pada CounterController, saya menambahkan:
 - o final RxInt counter = 0.obs; untuk menyimpan nilai counter dengan tipe reactive.
 - o void incrementCounter() { counter.value++; } untuk menambahkan nilai counter.
 - o void resetCounter() { counter.value = 0; } untuk mereset nilai counter.
2. Pada HomePage, saya melengkapi:
 - o Pada body, saya menggunakan Obx(() { ... }) untuk memperbarui tampilan nilai counter ketika nilainya berubah.
 - o Di dalam Obx, saya menampilkan nilai counter dengan "\${controller.counter.value}".
 - o Pada floatingActionButton, saya menambahkan:
 - Tombol "+" yang memanggil controller.incrementCounter() saat ditekan.
 - Tombol "Refresh" yang memanggil controller.resetCounter() saat ditekan.

Ketika aplikasi dijalankan, akan terlihat:

1. Halaman utama menampilkan nilai counter awal, yaitu 0.
2. Saat tombol "+" ditekan, nilai counter akan bertambah.
3. Saat tombol "Refresh" ditekan, nilai counter akan kembali ke 0.