

**TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XIII
NETWORKING**



Disusun Oleh :
Dzikri Naufal Wisnu
Pravida
2211104063
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

A. SOAL NOMOR 1. Apa yang dimaksud dengan state management pada Flutter

Jawab :

State management adalah teknik di Flutter untuk mengelola data aplikasi dan status UI di berbagai widget. Ini membantu mengontrol bagaimana data berubah dan bagaimana perubahan tersebut tercermin dalam antarmuka pengguna, memungkinkan penanganan status aplikasi secara efisien dan terorganisir.

B. SOAL NOMOR 2. Sebutkan dan jelaskan komponen komponen yang ada di dalam GetX

Jawab :

- a. GetX Controller : Mengelola status aplikasi dan logika bisnis
- b. Reactive state management : Memperbarui UI secara otomatis ketika status berubah
- c. Dependency injection : Membuat dan mengelola instance kelas dengan mudah
- d. Route management : Menyederhanakan navigasi antar halaman
- e. Dependency management : Menangani ketergantungan layanan dan pengontrol

C. SOAL NOMOR 3. Lengkapilah kode dibawah ini dan tampilkan hasil output 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
}

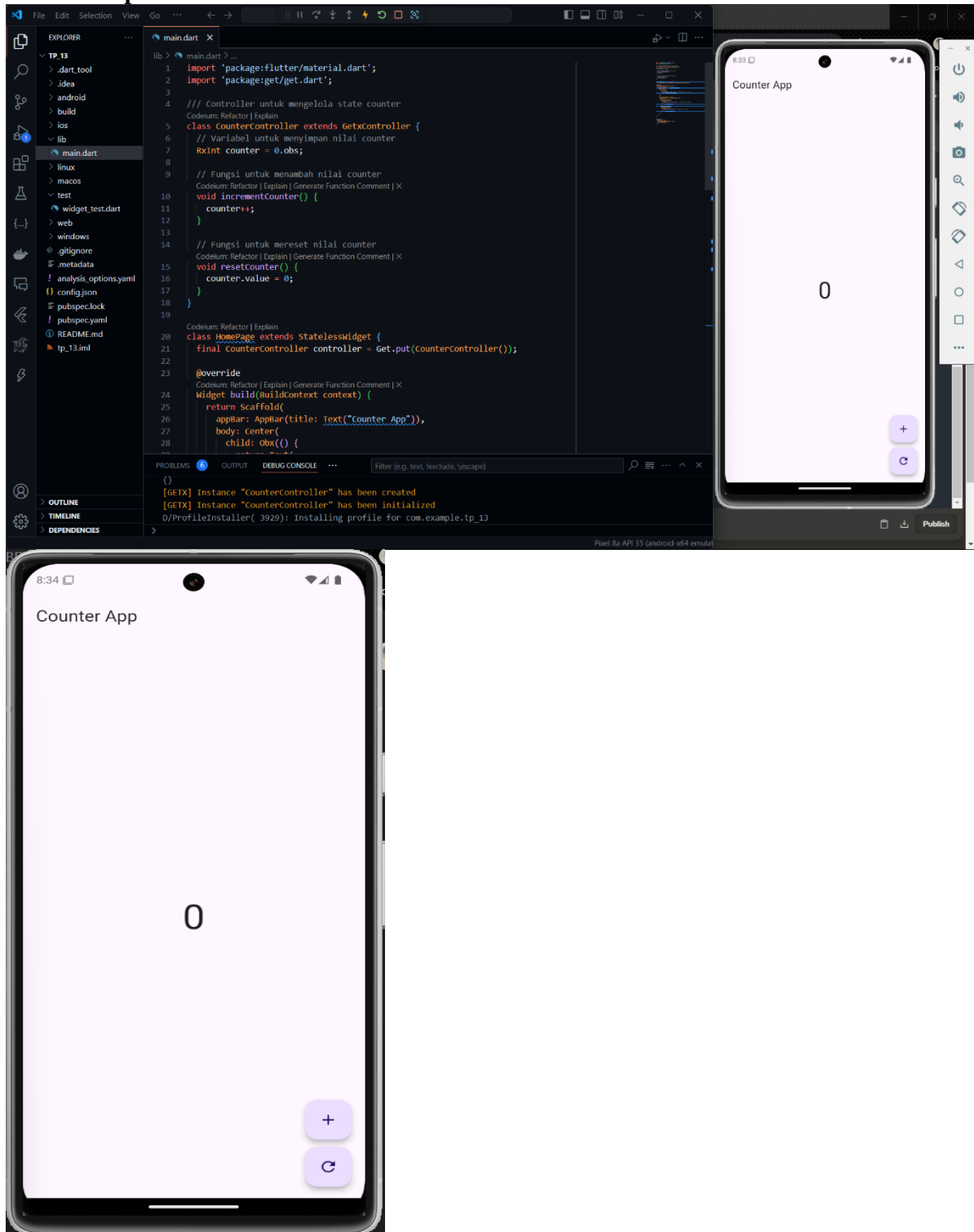
class HomePage extends StatelessWidget {
  final CounterController controller =
  Get.put(CounterController());
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Counter App")),
      body: Center(
        child: Obx(() {
          // TODO: Lengkapi 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(),
  ));
}
```

Jawab :

```
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3
4  /// Controller untuk mengelola state counter
5  class CounterController extends GetxController {
6    // Variabel untuk menyimpan nilai counter
7    RxInt counter = 0.obs;
8
9    // Fungsi untuk menambah nilai counter
10   void incrementCounter() {
11     counter++;
12   }
13
14   // Fungsi untuk mereset nilai counter
15   void resetCounter() {
16     counter.value = 0;
17   }
18 }
19
20 class HomePage extends StatelessWidget {
21   final CounterController controller = Get.put(CounterController());
22
23   @override
24   Widget build(BuildContext context) {
25     return Scaffold(
26       appBar: AppBar(title: Text("Counter App")),
27       body: Center(
28         child: Obx(() {
29           return Text(
30             "${controller.counter.value}", // Menampilkan nilai counter
31             style: TextStyle(fontSize: 48),
32           );
33         })),
34       floatingActionButton: Column(
35         mainAxisAlignment: MainAxisAlignment.end,
36         children: [
37           FloatingActionButton(
38             onPressed: () {
39               controller.incrementCounter(); // Menambah nilai counter
40             },
41             child: Icon(Icons.add),
42           ),
43           SizedBox(height: 10),
44           FloatingActionButton(
45             onPressed: () {
46               controller.resetCounter(); // Mereset nilai counter
47             },
48             child: Icon(Icons.refresh),
49           ),
50         ],
51       ),
52     );
53   }
54 }
55
56 void main() {
57   runApp(MaterialApp(
58     debugShowCheckedModeBanner: false,
59     home: HomePage(),
60   ));
61 }
62
63
```

Output :



Penjelasan :

Ini adalah program aplikasi sederhana menggunakan GetX untuk menghitung dengan menekan tombol '+' untuk menambah, dan tombol reset dibawahnya. Aplikasi ini membuat CounterController untuk mengelola status penghitung secara reaktif. Widget Obx memastikan UI diperbarui secara otomatis setiap kali nilai penghitung berubah. Ini mendemonstrasikan prinsip manajemen status reaktif dengan memisahkan UI dan logika, menjadikan kode lebih modular dan mudah dikelola.