

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



**Disusun Oleh :**  
**Maria Nathasya Desfera Pangestu / 2211104008**  
**SE0601**

**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**

## SOAL

1. Apa yang dimaksud dengan state management pada Flutter?
2. Sebut dan jelaskan komponen-komponen yang ada di dalam GetX.
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 screenshoot output dari sourcecode)

### Deskripsi Program

(deskripsikan program apa yang dibuat, memakai algoritma, dan cara kerja program sampai ke output yang dihasilkan dengan bahasa sendiri) **minimal 5 kalimat.**

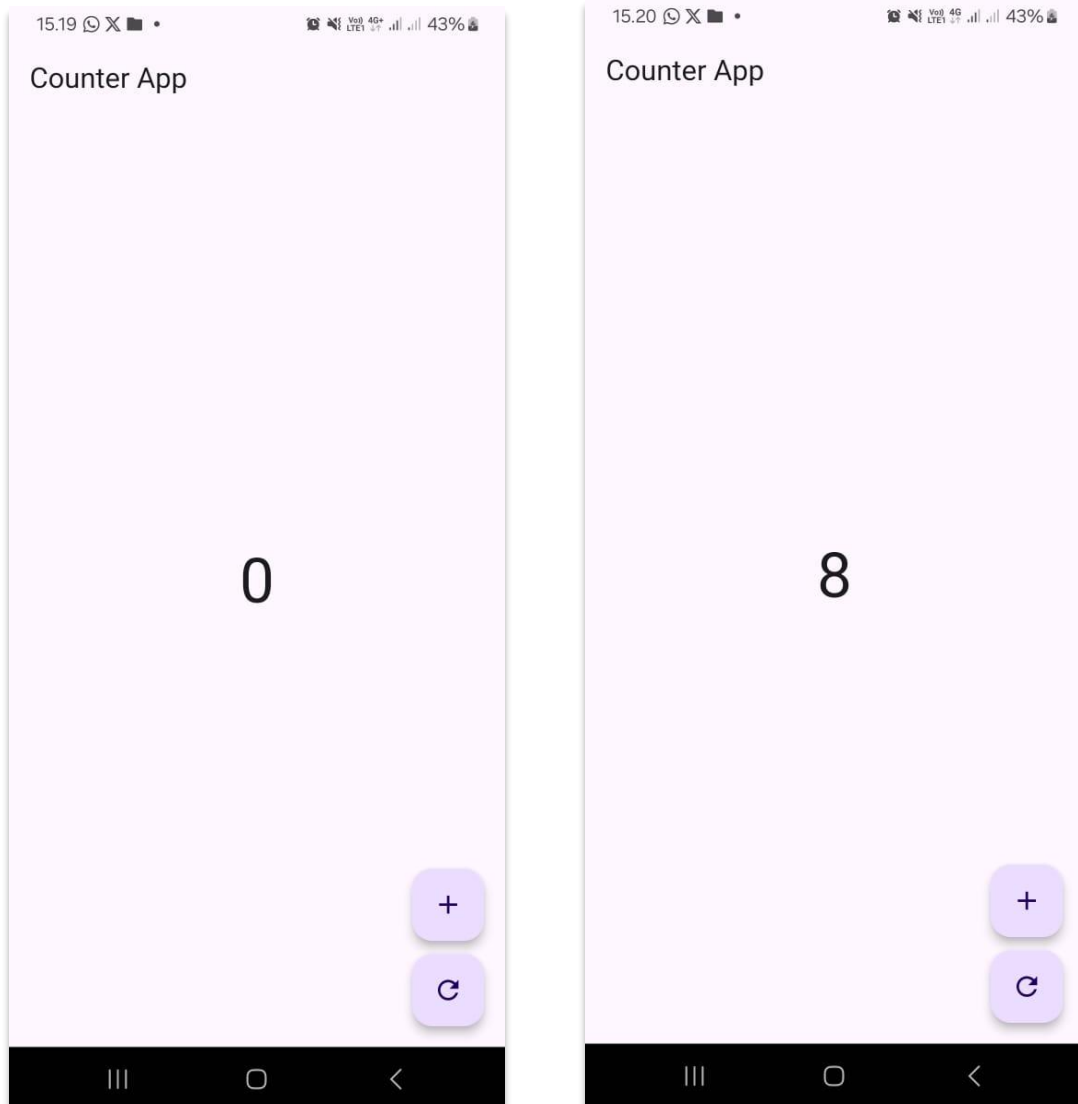
## Jawab

1. Yang dimaksud dengan state management pada Flutter yaitu cara mengelola dan mengontrol state aplikasi terdiri dari data dan informasi yang bisa berubah setiap saat yang mempengaruhi tampilan antarmuka pengguna. State disini seperti informasi misalnya input pengguna, status aplikasi, atau data dari internet. Tujuan utama state management ini agar memudahkan pemeliharaan, skalabilitas, dan menjaga agar tampilan UI tetap konsisten.
2. GetX yaitu salah satu library populer pada Flutter yang gunanya untuk state management, dependency injection, dan routing. Komponen-komponen yang ada di dalam GetX yaitu:
  - a. GetX (state management), yaitu inti dari state management pada GetX yang memungkinkan pengelolaan state reaktif menggunakan reactive programming sehingga state management jadi lebih efisien.
  - b. GetBuilder (state management), yaitu widget yang gunanya untuk mengelola state secara manual berbeda dengan GetX yang otomatis. Kontrol atas pembaruan state serta kapan widget harus diperbarui menjadi kelebihan dari GetBuilder.
  - c. GetXController (controller), merupakan kelas untuk mengelola state dan logika bisnis aplikasi. Disini juga bisa untuk mendefinisikan variable-variabel yang akan dikelola GetX serta fungsi untuk memperbarui state.
  - d. GetView (stateless widget), yaitu widget stateless untuk mengakses controller GetX. GetView akan otomatis mencari controller yang sesuai dan menggunakannya untuk menampilkan UI.
  - e. Get.put() dan Get.find() (dependency injection), Get.put() gunanya untuk memasukkan objek controller atau dependensi ke GetX dependency injection container. Sedangkan Get.find() untuk mengambil objek controller atau dependensi yang sudah disediakan sebelumnya dengan Get.put().
  - f. GetX (routing and navigation), dengan menggunakan Get.to(), Get.off(), dan Get.back() dapat melakukan navigasi antar halaman tanpa memerlukan navigator.
  - g. Snackbars, Dialogs, and BottomSheets, yaitu untuk menampilkan UI secara global tanpa harus mengakses konteks. Snackbars gunanya menampilkan pesan pendek di bagian bawah layar. Dialogs menampilkan dialog konfirmasi atau informasi. Dan BottomSheets untuk menampilkan menu atau informasi tambahan di bagian bawah layar.
  - h. GetStorage (persistent storage), yaitu untuk penyimpanan data lokal yang ringan pada GetX dalam bentuk key-value pair serta memungkinkan penyimpanan data tetap ada walaupun aplikasi tertutup.

3. Berikut ini adalah code yang sudah dilengkapi  
Source code main.dart

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

Hasil screenshot output sebelum di klik tombol + dan setelah di klik tombol +



#### Deskripsi program

Program tersebut bernama Counter App menggunakan GetX untuk pengelolaan state reaktif. Terdiri dari CounterController, yang menggunakan variabel reaktif counter untuk mengelola nilai counter dan memiliki dua metode yaitu tambahCounter() untuk menambah nilai dan resetCounter() untuk mengatur ulang nilai menjadi 0. Pada tampilan UI, halaman utama, atau HomePage, menampilkan nilai counter secara real-time menggunakan Obx. Tombol + untuk menambah nilai counter dan tombol refresh untuk mereset nilai. Menggunakan widget MaterialApp sebagai entry point, tanpa menampilkan label debug. Aplikasi ini responsif, efisien, dan mudah dikelola karena menggunakan GetX.