

**TUGAS PENDAHULUAN  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL 13  
NETWORKING**



**Disusun Oleh :**

**Fahmi Hasan Asagaf/ 2311104074**

**S1SE-07-02**

**Dosen Pengampu :**

**Yudha Islami Sulistya, S.Kom., M.Cs.**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

## TUGAS UNGUIDED 11

### A. SOAL

#### 1. 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(),
  ));
}
```

## 2.JAWABAN

### 1. Apa yang dimaksud dengan state management pada Flutter?

State management pada Flutter adalah cara mengelola dan mengatur perubahan data (state) dalam aplikasi agar tampilan UI selalu sinkron dengan data tersebut. Ketika state berubah (misalnya nilai counter bertambah), Flutter akan membangun ulang widget yang bergantung pada state tersebut. State management membantu aplikasi menjadi lebih terstruktur, mudah dipelihara, dan efisien, terutama untuk aplikasi yang kompleks. Contoh state management di Flutter antara lain: `setState`, `Provider`, `Bloc`, dan `GetX`.

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

Komponen utama dalam `GetX` adalah:

#### -State Management

Mengelola state dengan reaktif menggunakan `Obx`, `Rx`, dan `GetxController` tanpa perlu `setState`.

#### -Dependency Injection

Mengelola dan menyediakan dependency menggunakan `Get.put()`, `Get.find()`, sehingga kode lebih rapi dan tidak saling bergantung langsung.

#### -Route Management (Navigation)

Navigasi antar halaman tanpa context, menggunakan `Get.to()`, `Get.back()`, dan `Get.off()`.

#### -Reactive Programming (Rx)

Menggunakan variabel reaktif seperti `RxInt`, `RxString`, yang otomatis memperbarui UI saat nilainya berubah.

### 3.Kode yang sudah dilengkapi

```
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(() {
          return Text(
            controller.counter.value.toString(),
            style: TextStyle(fontSize: 48),
          );
        }),
      ),
    ),
  )
}
```

```

floatingActionButton: Column(
  mainAxisAlignment: MainAxisAlignment.end,
  children: [
    FloatingActionButton(
      onPressed: () {
        controller.increment();
      },
      child: Icon(Icons.add),
    ),
    SizedBox(height: 10),
    FloatingActionButton(
      onPressed: () {
        controller.reset();
      },
      child: Icon(Icons.refresh),
    ),
  ],
),
);
}
}

```

```

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

```

## Output

### Hasil Output

Saat aplikasi dijalankan, layar menampilkan angka 0.

Tombol ( + ) akan menambah nilai counter (1, 2, 3, dan seterusnya).

Tombol refresh akan mereset nilai counter kembali ke 0.

Perubahan angka terjadi secara real-time tanpa setState.

### Penjelasan kode

Kode ini menggunakan GetX untuk mengelola state counter. CounterController menyimpan state berupa variabel reaktif counter menggunakan .obs. Widget Obx berfungsi untuk mendengarkan perubahan nilai counter dan otomatis memperbarui UI ketika nilainya berubah. Dependency controller dikelola dengan Get.put(), sehingga UI dapat mengakses dan memanggil fungsi increment() dan reset() tanpa menggunakan setState.