TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK

MODUL XIII

NETWORKING

Disusun Oleh :

Devrin Anggun Saputri / 2211104001

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

TUGAS PENDAHULUAN

SOAL

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

   Jawaban:

   State management dalam Flutter adalah metode untuk mengatur "state" atau data yang berubah-ubah dalam aplikasi. Contohnya meliputi teks yang ditampilkan di layar, posisi tombol, atau daftar item yang terlihat. Selain itu, state management memungkinkan pemisahan antara logika aplikasi dan tampilan, sehingga logika tersebut dapat digunakan kembali. State sendiri adalah informasi yang mempengaruhi tampilan aplikasi dan dapat mengalami perubahan, terutama saat pengguna berinteraksi dengan aplikasi.

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

   Jawaban:

   GetX adalah salah satu solusi state management yang menawarkan kinerja tinggi, dependency injection yang cerdas, serta manajemen navigasi yang cepat dan mudah. Berikut adalah komponen-komponen utama dalam GetX:

   1. **Pengelolaan State (State Management)**

      GetX menyediakan dua pendekatan untuk mengelola state: **Reactive State** dan **Simple State**.

      a. **Reactive State Management**

         Dengan menggunakan kelas Rx, data menjadi reaktif, sehingga UI otomatis diperbarui saat data berubah. Komponen utamanya:

         - **Rx**: Membuat data menjadi reaktif (contoh: RxInt, RxString).
         - **.obs**: Shortcut untuk menjadikan data reaktif.

      b. **Simple State Management**
      Menggunakan widget seperti **GetBuilder** atau **GetX** untuk memperbarui UI secara manual tanpa membuat data reaktif. Komponennya meliputi:

         - **GetBuilder**: Memperbarui widget hanya ketika ada perubahan state.

   2. **Manajemen Navigasi (Route Management)**

      GetX memungkinkan navigasi tanpa menggunakan Navigator bawaan Flutter. Fitur utamanya:

      - **Get.to()**: Berpindah ke halaman lain.
      - **Get.off()**: Berpindah ke halaman lain sekaligus menutup halaman sebelumnya.
      - **Get.back()**: Kembali ke halaman sebelumnya.
      - **Get.arguments**: Mengirim data antar halaman.

   3. **Dependency Injection (DI)**

      GetX memiliki dependency injection bawaan untuk mengelola lifecycle objek dan membuatnya dapat diakses di mana saja. Komponen utamanya:

      - **Get.put()**: Menyimpan instance controller untuk akses global.
      - **Get.lazyPut()**: Menyimpan instance yang hanya dibuat saat diperlukan.

- **Get.find()**: Mengambil instance yang sudah disimpan.

4. **Middleware**

   Middleware berfungsi untuk mengontrol akses ke halaman, seperti memeriksa apakah pengguna sudah login sebelum masuk. Komponen utamanya:

   - **GetMiddleware**: Kelas untuk membuat middleware.
   - **redirect**: Mengarahkan pengguna ke halaman lain jika syarat tertentu tidak terpenuhi.

5. **Snackbar, Dialog, dan BottomSheet**

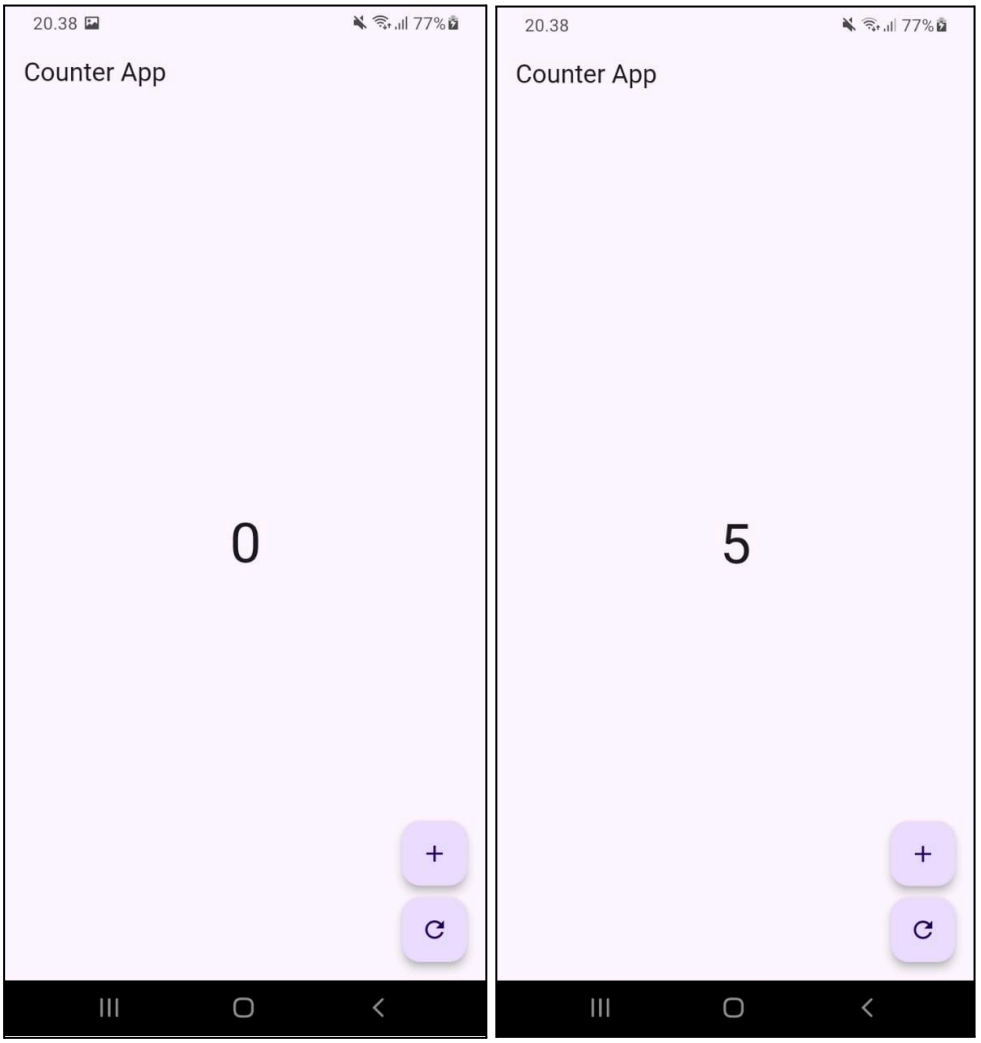
   GetX menyediakan utilitas bawaan untuk menampilkan notifikasi, dialog, atau bottom sheet tanpa memerlukan BuildContext. Komponennya:

- **Get.snackbar**: Menampilkan snackbar.
- **Get.defaultDialog**: Menampilkan dialog.
- **Get.bottomSheet**: Menampilkan bottom sheet.

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

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

/// Controller untuk mengelola state counter
class CounterController extends GetxController {
  // Variabel untuk menyimpan nilai counter, menggunakan RxInt agar reaktif
  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 {
  // Menghubungkan Controller dengan view
  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 yang diperbarui secara reaktif
          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(),
  ));
}
```

Output:



Deskripsi Program:

Program di atas adalah aplikasi Flutter sederhana yang menggunakan **GetX** untuk mengelola state. Aplikasi ini menampilkan nilai counter yang dapat bertambah atau direset sesuai interaksi pengguna. State dikelola oleh sebuah controller bernama CounterController, yang memiliki variabel reaktif counter untuk menyimpan nilai, serta fungsi increment dan reset untuk mengubah nilai tersebut.

Antarmuka utama aplikasi berada pada widget HomePage. Di sini, controller dihubungkan menggunakan metode Get.put(), sehingga dapat diakses di seluruh bagian widget. Nilai counter ditampilkan dalam teks besar di tengah layar dengan widget reaktif Obx, yang memastikan UI otomatis diperbarui setiap kali nilai counter berubah. Dua tombol aksi disediakan, yaitu tombol "+" untuk meningkatkan nilai counter dan tombol "⟳" untuk meresetnya ke nol.

Fungsi utama main() menjalankan aplikasi dengan widget MaterialApp, menampilkan HomePage sebagai layar utama. Secara keseluruhan, program ini memanfaatkan GetX untuk mengelola state secara efisien, menghadirkan aplikasi sederhana dan responsif untuk pengguna.

Pada tampilan awal program, angka yang ditampilkan adalah 0. Saat pengguna menekan tombol (+), angka tersebut akan bertambah. Jika tombol ditekan sebanyak 5 kali, angka di layar akan berubah menjadi 5. Ketika tombol reset ditekan, angka kembali menjadi 0.