TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK

MODUL XIII

NETWORKING

Disusun Oleh :

S a l m a n A l f a R

2 2 1 1 1 0 4 0 5 6

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?
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(),
  ));
}
```

1. Status aplikasi Flutter merujuk pada semua objek yang digunakannya untuk menampilkan UI atau mengelola sumber daya sistem. Manajemen status adalah cara kita mengatur aplikasi kita agar dapat mengakses objek-objek ini secara paling efektif dan membaginya di antara berbagai widget.

   Halaman ini membahas banyak aspek manajemen status, termasuk:

   Menggunakan StatefulWidget

   Membagi status di antara widget menggunakan konstruktor, InheritedWidget, dan callback

   Menggunakan Listenable untuk memberi tahu widget lain saat ada yang berubah

   Menggunakan Model-View-ViewModel (MVVM) untuk arsitektur aplikasi Anda

2. GetX Controller

Deskripsi: Komponen utama untuk mengelola status dan logika bisnis aplikasi.

Fungsi: Menyimpan dan mengelola data serta mengontrol pembaruan status.

Rx (Reactive) Types

Deskripsi: Tipe data reaktif yang memungkinkan nilai untuk dipantau dan diperbarui otomatis.

Fungsi: Memungkinkan pembaruan UI otomatis saat nilai berubah.

GetBuilder

Deskripsi: Widget untuk memperbarui UI berdasarkan status yang ada di controller.

Fungsi: Membuat ulang UI hanya saat status controller berubah.

Obx

Deskripsi: Widget untuk membangun UI yang bergantung pada perubahan nilai reaktif.

Fungsi: Memperbarui UI otomatis saat nilai reaktif berubah.

GetX

Deskripsi: Widget yang mengelola state secara global, memungkinkan akses ke controller di seluruh aplikasi.

Fungsi: Menyediakan akses ke controller dan state secara langsung.

Get.put()

Deskripsi: Digunakan untuk mendaftarkan controller agar dapat diakses di seluruh aplikasi.

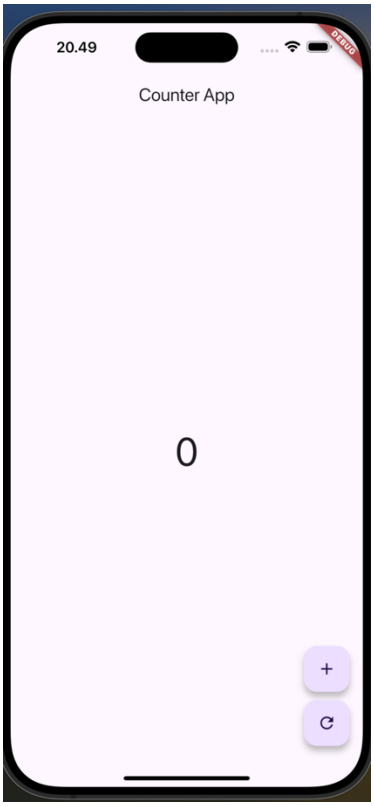
Fungsi: Memungkinkan akses global ke controller yang sudah diinisialisasi.

Get.lazyPut()

Deskripsi: Mendaftarkan controller hanya saat dibutuhkan.

Fungsi: Menghemat memori dengan menunda inisialisasi controller sampai diperlukan.

3. **Screenshoot Output**



Deskripsi Program

Program di atas adalah contoh implementasi penggunaan package GetX dalam Flutter untuk mengelola state secara reaktif. Di dalam aplikasi ini, CounterController bertanggung jawab untuk menyimpan dan mengelola nilai counter, yang diwakili oleh variabel counter yang bertipe RxInt. Dengan menggunakan Rx (Reactive) types, nilai counter dapat dipantau secara otomatis, dan setiap perubahan nilai akan langsung memperbarui UI yang menggunakan widget Obx. Fungsi increment() digunakan untuk menambah nilai counter, sedangkan reset() mengembalikan nilai counter ke 0. Obx memungkinkan tampilan nilai counter untuk selalu terupdate sesuai dengan perubahan yang terjadi, tanpa perlu manual memanggil setState(). FloatingActionButton yang ada pada aplikasi ini digunakan untuk memicu fungsi-fungsi tersebut: satu untuk menambah nilai dan satu lagi untuk mereset nilai counter ke nilai awal.