LAPORAN TEORI MOBILE PROGRAMMING
MODUL 11

Nama : Firman Fadilah Noor
NIM : 240605110083
Kelas : B
Tanggal : 13 Oktober 2025

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
GANJIL 2025/2026

I. Tujuan

1. Memahami konsep manajemen *state* menggunakan GetX
2. Menerapkan *controller* dan variable *reaktif* dalam pembaruan tampilan
3. Membandingkan penggunaan *GetX* dengan konvensional *setState()* pada Flutter.

II. Langkah Kerja

1. Membuat proyek Flutter baru dengan nama getx_app.
2. Menambahkan depensi pada file *pubspec.yaml.*
3. Membuat struktur *folder* yang terdiri atas *model, view,* dan *viewmodel/.*
4. Membuat kelas *TasbihController* yang berisi variabel reaktif *counter* dan *progress,* serta *method invrementCounter()* dan *resetCounter().*
5. Menghubungkan *controller* dengan tampilan menggunakan *widget Obx.*
6. Menjalankan aplikasi untuk melihat perubahan nilai conter dan *progress bar* secara *real time.*

III. Screenshot Hasil

a. Kode Program

- view/

home.dart

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

class Home extends StatelessWidget {
  const Home({super.key});

  @override
  Widget build(BuildContext context) {
    final TasbihController controller = Get.put(TasbihController());

    return Scaffold(
      backgroundColor: const Color.fromARGB(255, 119, 210, 145),
      body: SafeArea(
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              // Nilai counter (angka besar di tengah)
              Obx(
                    () => Text(
                  '${controller.counter.value.round()}',
                  style: const TextStyle(
                    fontSize: 250,
                    color: Colors.white,
                    fontWeight: FontWeight.bold,
                  ), // TextStyle TextStyle
                ), // Text Text
              ), // Obx Obx

              // Progress bar linear
              Obx(
                    () => Padding(
```

```
                padding: const EdgeInsets.symmetric(horizontal: 40.0),
                child: LinearProgressIndicator(
                  value: controller.progress.value / 100,
                  backgroundColor: Colors.white54,
                  color: Colors.amberAccent.shade400,
                  minHeight: 15,
                  borderRadius: BorderRadius.circular(10),
                ), // LinearProgressIndicator LinearProgressIndicator
              ), // Padding Padding
            ), // Obx Obx

            const SizedBox(height: 75),

            // Tombol Fingerprint (Tambah Hitungan)
            ClipRRect(
              borderRadius: const BorderRadius.all(Radius.circular(50)),
              child: InkWell(
                onTap: controller.incrementCounter,
                child: Container(
                  decoration: const BoxDecoration(color: Colors.white),
                  padding: const EdgeInsets.all(30),
                  child: const Icon(
                    Icons.fingerprint,
                    size: 100,
                    color: Colors.green,
                  ), // Icon Icon
                ), // Container Container
              ), // InkWell InkWell
            ), // ClipRRect ClipRRect
          ],
        ), // Column Column
      ), // Center Center
    ), // SafeArea SafeArea
```

```
      // Tombol reset
      floatingActionButton: FloatingActionButton(
        onPressed: controller.resetCounter,
        backgroundColor: Colors.white,
        child: const Icon(
          Icons.refresh_outlined,
          color: Colors.black,
        ), // Icon Icon
      ), // FloatingActionButton FloatingActionButton
    ); // Scaffold Scaffold
  }
}
```

- viewmodel/

tasbih_controller

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

class TasbihController extends GetxController {
  var counter = 0.0.obs;
  var progress = 0.0.obs;
  final double maxCount = 33;

  void incrementCounter() {
    if (counter < maxCount) {
      counter.value++;
      progress.value = (counter.value / maxCount) * 100;
    }
  }

  void resetCounter() {
    counter.value = 0;
    progress.value = 0;
  }
}

```

Main.dart

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

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Tasbih Digital GetX',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.green),
        useMaterial3: true,
      ), // ThemeData
      home: const Home(),
    ); // GetMaterialA
  }
}
```

b. Penjelasan Kode Program.

Pada aplikasi **Tasbih Digital GetX**, kode program dibagi menjadi tiga bagian utama, yaitu file main.dart, tasbih_controller.dart, dan home.dart. Ketiga file ini saling terhubung dan memiliki fungsi masing-masing untuk membuat aplikasi dapat berjalan dengan baik dan menampilkan hasil secara interaktif.

File **main.dart** berfungsi sebagai titik awal atau pintu utama dari aplikasi. Di dalamnya digunakan GetMaterialApp() yang menggantikan MaterialApp() agar aplikasi bisa memanfaatkan fitur manajemen state dari GetX. Pada bagian ini juga diatur tampilan awal aplikasi, tema warna, serta halaman utama yang akan ditampilkan pertama kali, yaitu halaman Home(). Dengan menggunakan GetX, navigasi dan pengelolaan data di aplikasi menjadi lebih mudah dan efisien tanpa perlu banyak kode tambahan.

Kemudian, pada file **tasbih_controller.dart**, dibuat sebuah kelas bernama TasbihController yang bertugas mengatur logika dan data utama dari aplikasi. Di

dalamnya terdapat dua variabel reaktif, yaitu counter dan progress. Variabel counter digunakan untuk menyimpan jumlah hitungan tasbih, sedangkan progress menunjukkan seberapa jauh hitungan tersebut berjalan dari total 33 kali. Karena kedua variabel tersebut bersifat reaktif (menggunakan .obs), maka setiap kali nilainya berubah, tampilan di layar juga ikut berubah secara otomatis tanpa perlu memanggil setState(). Selain itu, terdapat dua fungsi utama yaitu incrementCounter() untuk menambah hitungan, dan resetCounter() untuk mengembalikan nilai hitungan ke nol.

Bagian terakhir, yaitu file **home.dart**, berisi tampilan utama dari aplikasi. Pada bagian ini controller dipanggil menggunakan Get.put(TasbihController()) agar data dari controller bisa digunakan di dalam widget. Tampilan aplikasi terdiri dari angka besar di tengah layar yang menunjukkan jumlah hitungan, progress bar yang menampilkan perkembangan hitungan, serta tombol fingerprint besar di bawahnya untuk menambah angka setiap kali ditekan. Semua elemen yang menampilkan data dibungkus dengan widget Obx(), yang membuat tampilan bisa langsung berubah ketika data di controller diperbarui. Selain itu, terdapat tombol **refresh** di bagian bawah kanan layar untuk mengembalikan hitungan ke nol.

Secara keseluruhan, kode program ini menunjukkan bagaimana **GetX** bekerja dalam mengelola state secara reaktif di Flutter. Dengan menggunakan GetX, tampilan aplikasi menjadi lebih dinamis dan mudah dikontrol tanpa perlu banyak kode. Aplikasi **Tasbih Digital GetX** ini juga menjadi contoh sederhana bagaimana Flutter dapat digunakan untuk membuat aplikasi yang interaktif, efisien, dan nyaman digunakan oleh pengguna.

Output:

IV. Kesimpulan.

Dari hasil praktikum Modul 11 tentang **Manajemen State dengan GetX**, dapat disimpulkan bahwa penggunaan GetX dalam pengembangan aplikasi Flutter memberikan banyak kemudahan, terutama dalam hal pengelolaan data dan pembaruan tampilan. Dengan menggunakan pendekatan reaktif, setiap perubahan data yang terjadi di dalam controller langsung ditampilkan secara otomatis di layar tanpa perlu memanggil setState(). Hal ini membuat proses pengembangan menjadi lebih cepat, efisien, dan kode yang dihasilkan juga lebih rapi serta mudah dipahami.

Aplikasi **Tasbih Digital GetX** yang dibuat dalam praktikum ini menjadi contoh sederhana bagaimana GetX bekerja dalam mengelola state secara real-time. Saat tombol ditekan, angka hitungan langsung bertambah dan progress bar ikut berubah tanpa jeda, menunjukkan bahwa sistem reaktif dari GetX berjalan dengan baik. Selain itu, fitur reset juga berfungsi sempurna untuk mengembalikan nilai ke nol, membuktikan bahwa semua logika pada controller terhubung dengan benar ke tampilan.

Secara keseluruhan, melalui praktikum ini dapat dipahami bahwa GetX sangat membantu dalam membangun aplikasi Flutter yang interaktif dan responsif. Selain membuat tampilan lebih dinamis, penggunaan GetX juga menjadikan aplikasi lebih ringan dan mudah dikembangkan di masa depan.