

LAPORAN TEORI MOBILE PROGRAMMING
MODUL 11



Nama : Erai Bagusalm
NIM : 240605110088
Kelas : Mobile Programming B
Tanggal : 18 Oktober 2025

PROGRAM STUDI 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 variabel reaktif dalam pembaruan tampilan
3. Membandingkan penggunaan GetX dengan pendekatan konvensional setState() pada Flutter.

ii. Langkah Kerja

1. Membuat proyek Flutter baru dengan nama getx_app.
2. Menambahkan dependensi 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 incrementCounter() dan resetCounter().
5. Menghubungkan controller dengan tampilan menggunakan widget Obx.
6. Menjalankan aplikasi untuk melihat perubahan nilai counter dan progress bar secara real time.

iii. Screenshot Hasil

a. Kode Program

Main.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import 'view/home.dart';
4
5 void main() {
6   runApp(const MyApp());
7 }
8
9 class MyApp extends StatelessWidget {
10   const MyApp({super.key});
11
12   @override
13   Widget build(BuildContext context) {
14     return GetMaterialApp(
15       debugShowCheckedModeBanner: false,
16       title: 'Tasbih Digital GetX',
17       theme: ThemeData(
18         colorScheme: ColorScheme.fromSeed(seedColor: Colors.green),
19         useMaterial3: true
20       ),
21       home: const Home(),
22     );
23   }
24 }
25
```

tasbih_controller.dart

```
1 import 'package:get/get.dart';
2
3 class TasbihController extends GetxController {
4   var counter = 0.0.obs;
5   var progress = 0.0.obs;
6   final double maxCount = 33;
7
8   void incrementCounter() {
9     if (counter < maxCount) {
10       counter.value++;
11       progress.value = (counter.value) / maxCount * 100;
12     }
13   }
14
15   void resetCounter() {
16     counter.value = 0;
17     progress.value = 0;
18   }
19 }
```

Home.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:simple_circular_progress_bar/simple_circular_progress_bar.dart';
3 import 'package:get/get.dart';
4 import '../viewmodel/tasbih_controller.dart';
5
6 final TasbihController Controller = Get.put(TasbihController());
7
8 class Home extends StatelessWidget {
9   const Home({super.key});
10
11
12
13   @override
14   Widget build(BuildContext context) {
15     return Scaffold(
16       backgroundColor: const Color.fromARGB(255, 119, 210, 145),
17       body: SafeArea(
18         child: Center(
19           child: Column(
20             mainAxisAlignment: MainAxisAlignment.center,
21             children: [
22               Obx(
23                 () => Text(
24                   '${Controller.counter.value.round()}',
25                   style: const TextStyle(
26                     fontSize: 250,
27                   ),
28                 ),
29               ),
30               Obx(
31                 () => Padding(
32                   padding: const EdgeInsets.symmetric(horizontal: 40.0),
33                   child: LinearProgressIndicator(
34                     value: Controller.progress.value / 100,
35                     backgroundColor: Colors.white54,
36                     color: Colors.amberAccent.shade400,
37                     minHeight: 15,
38                     borderRadius: BorderRadius.circular(10),
39                   ),
40                 ),
41               ),
42               const SizedBox(height: 75),
43               ClipRRect(
44                 borderRadius: const BorderRadius.all(Radius.circular(50)),
45                 child: InkWell(
46                   onTap: Controller.incrementCounter,
47                   child: Container(
48                     decoration: const BoxDecoration(color: Colors.white),
49                     padding: const EdgeInsets.all(30),
50                     child: const Icon(
51                       Icons.fingerprint,
52                       size: 100,
53                     ),
54                   ),
55                 ),
56               ),
57             ],
58           ),
59         ),
60       ),
61       floatingActionButton: FloatingActionButton(
62         onPressed: Controller.resetCounter,
63         backgroundColor: Colors.white,
64         child: const Icon(
65           Icons.refresh_outlined,
66           color: Colors.black,
67         ),
68       ),
69     );
70   }
71 }
```

b. Penjelasan Kode Program

- Cara Kerja Controller, Variabel Reaktif, dan Obx dalam Update Real-Time.

Pada aplikasi Tasbih Digital ini, controller (TasbihController) berperan sebagai pusat pengelola data dan logika. Semua proses seperti

menambah hitungan tasbih, menghitung progres, dan mengatur batas maksimal hitungan (33) dikendalikan melalui controller. Dengan menggunakan `Get.put()`, controller bisa langsung diakses oleh tampilan tanpa perlu dikirim lewat parameter.

Variabel seperti counter dan progress dibuat sebagai variabel reaktif dengan `.obs`. Artinya, setiap kali nilainya berubah, sistem akan langsung memberi tahu tampilan bahwa ada data yang diperbarui. Jadi, tidak perlu lagi perintah refresh manual seperti pada cara lama.

Agar tampilan bisa menyesuaikan perubahan data secara otomatis, digunakan widget `Obx`. `Obx` akan “mengawasi” variabel reaktif di dalamnya. Saat nilai counter berubah, angka di layar langsung ikut berubah. Saat progress berubah, batang progres juga langsung bergerak. Semua ini terjadi secara real-time dan otomatis.

- **Perbedaan `GetX` dan `setState()`**

Pada pendekatan `setState()`, setiap perubahan data harus dibungkus dengan `setState`. Akibatnya, seluruh tampilan dalam satu halaman akan diperbarui, meskipun hanya satu bagian kecil yang berubah. Cara ini cocok untuk aplikasi kecil, tapi bisa jadi kurang efisien untuk aplikasi yang lebih kompleks.

Sedangkan dengan `GetX`, data dikelola terpisah di dalam controller dan menggunakan variabel reaktif. Tampilan hanya akan memperbarui bagian yang benar-benar berubah melalui `Obx`. Ini membuat aplikasi:

- Lebih ringan
- Lebih cepat
- Kode lebih rapi dan terstruktur
- Lebih mudah dikembangkan untuk skala besar

Output :



iv. Kesimpulan

Penggunaan GetX pada pengembangan aplikasi Tasbeeh Digital memberikan kemudahan dalam pengelolaan state dan pembaruan tampilan secara real-time. Dengan pendekatan ini, proses pengembangan menjadi lebih efisien, kode lebih terstruktur, serta lebih mudah untuk dikembangkan di masa depan.

- GetX memungkinkan pembaruan tampilan secara otomatis tanpa perlu memanggil `setState()`, sehingga proses update UI menjadi lebih sederhana dan efisien.
- Variabel reaktif (`.obs`) membantu menjaga sinkronisasi antara data dan tampilan, karena setiap perubahan nilai langsung ditampilkan secara real-time.
- Controller pada GetX memisahkan logika bisnis dari tampilan, sehingga struktur kode menjadi lebih rapi, terorganisir, dan lebih mudah untuk dikembangkan maupun dipelihara.