

LAPORAN PRAKTIKUM MOBILE PROGRAMMING
MODUL 11 – MANAJEMEN STATE DENGAN GET X



Nama : KHARISMA ROMADHON EKA SAFITRI

NIM : 230605110107

Kelas : E

Tanggal : Kamis, 18 September 2025

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

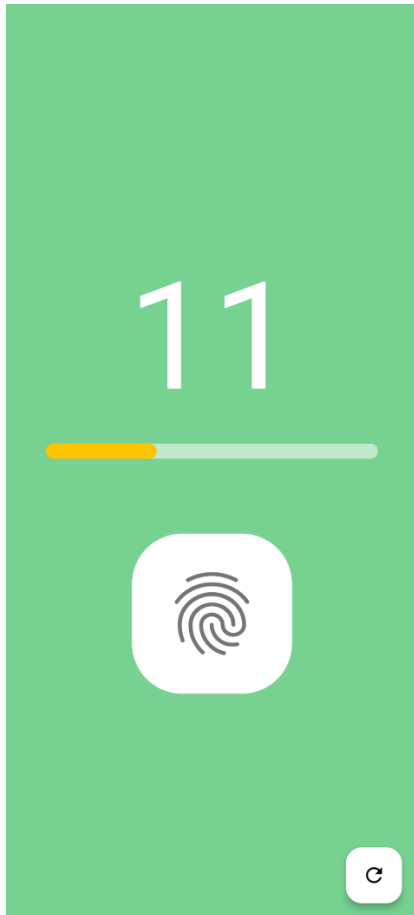
I. Tujuan

- Memahami konsep manajemen state menggunakan GetX dalam Flutter.
- Mampu memisahkan logika bisnis (controller) dan antarmuka (view) agar kode lebih bersih.
- Mengimplementasikan variabel reaktif (.obs) dan widget Obx untuk memperbarui UI secara otomatis tanpa menggunakan setState().

II. Langkah Kerja

1. Membuat proyek Flutter baru bernama tasbih_getx dan menambahkan dependensi get pada pubspec.yaml.
2. Membuat struktur folder view (untuk UI) dan viewmodel (untuk Logic).
3. Membuat class TasbihController (extends GetxController) yang berisi variabel counter dan progress yang ditandai dengan .obs agar bersifat reaktif.
4. Membuat fungsi logika incrementCounter() dan resetCounter() di dalam controller untuk mengubah nilai data.
5. Membuat halaman Home dan menginisialisasi controller menggunakan Get.put().
6. Membungkus widget teks angka dan progress bar dengan widget Obx(() => ...) agar UI diperbarui otomatis saat data berubah.
7. Mengubah MaterialApp menjadi GetMaterialApp di main.dart.

III. Screenshot Hasil



IV. Source Code

Main.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,
      ),
      home: const Home(),
    );
  }
}

```

V. Kesimpulan

Efisiensi: GetX memungkinkan pembaruan tampilan yang sangat efisien. Hanya widget yang dibungkus Obx yang akan dibangun ulang (rebuild) saat data berubah, berbeda dengan setState() yang bisa merender ulang satu halaman penuh.

Kebersihan Kode: Logika aplikasi terpisah sepenuhnya di TasbihController, sehingga file UI (home.dart) hanya berisi kode tampilan. Ini membuat kode lebih mudah dibaca dan dipelihara (maintainable).

Reaktif: Penggunaan `.obs` membuat variabel menjadi observable (dapat dipantau), sehingga sinkronisasi antara data dan tampilan terjadi secara real-time dan otomatis..