

LAPORAN PRAKTIKUM PEMROGRAMAN MOBILE

(StatefulWidget)



Dosen Pembimbing:

A'LA SYAUQI,M.Kom

Disusun oleh¹:

Nama : Muhammad Salman Alfarisi

NIM : 230605110060

Kelas : C

Tanggal : Rabu, 1 Oktober 2025

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG 2025/2026**

¹ Muhammad Salman Alfarisi (230605110060)

I. Tujuan

untuk memahami dan mengimplementasikan konsep *StatefulWidget* dalam pengembangan aplikasi Flutter. Mahasiswa diharapkan mampu membuat aplikasi *tasbih digital* yang dapat menambah dan mereset hitungan menggunakan *state management* dasar seperti `setState()` dan `ValueNotifier`. Selain itu, mahasiswa juga belajar menggunakan *package* eksternal `simple_circular_progress_bar` untuk menampilkan progres hitungan dalam bentuk lingkaran secara visual dan interaktif. Melalui tugas ini, mahasiswa dapat memahami bagaimana suatu *widget* dapat berubah secara dinamis sesuai interaksi pengguna dan memperkuat dasar dalam pengelolaan *state* di Flutter sebelum mempelajari manajemen *state* yang lebih kompleks seperti *Provider* atau *BLoC*.

II. Langkah Kerja

1. Membuat proyek Flutter baru dengan nama **tasbih_app**.
2. Menambahkan dependensi `simple_circular_progress_bar` pada file **pubspec.yaml** dan menjalankan perintah `flutter pub get`.
3. Membuka file **main.dart** dan melakukan *import* terhadap paket `simple_circular_progress_bar`.
4. Membuat *StatefulWidget* bernama **MyApp** yang berisi variabel `_valueNotifier` dan `counter` untuk mengatur progres dan jumlah hitungan tasbih.
5. Menulis method **incrementCounter()** untuk menambah nilai tasbih hingga maksimal 33 dan memperbarui progres lingkaran.
6. Menulis method **resetCounter()** untuk mengembalikan nilai tasbih ke 0 dan me-reset progres lingkaran.
7. Menyusun tampilan UI dalam method **build()** menggunakan komponen seperti `Text`, `SimpleCircularProgressBar`, `InkWell` (tombol fingerprint untuk menambah hitungan), serta `FloatingActionButton` untuk reset.
8. Menjalankan aplikasi dan menguji fungsi tombol tambah dan reset untuk memastikan perubahan *state* bekerja dengan benar.

Kode Program:

```
main.dart x pubspec.yaml
1 import 'package:flutter/material.dart';
2 import 'package:flutter/services.dart';
3 import 'package:simple_circular_progress_bar/simple_circular_progress_bar.dart';
4
5 void main() {
6   runApp(const MyApp());
7 }
8
9 class MyApp extends StatefulWidget {
10   const MyApp({super.key});
11
12   @override
13   State<MyApp> createState() => _MyAppState();
14 }
15
16 class _MyAppState extends State<MyApp> {
17   late ValueNotifier<double> _valueNotifier;
18   late double counter;
19
20   @override
21   void initState() {
22     super.initState();
23     _valueNotifier = ValueNotifier(0.0);
24     counter = 0.0;
25   }
26
27   @override
28   void dispose() {
29     _valueNotifier.dispose();
30     super.dispose();
31   }
32 }
```

```
32
33 void incrementCounter() {
34   setState() {
35     if (counter < 33) {
36       counter++;
37       _valueNotifier.value = (counter / 33) * 100;
38     }
39   });
40 }
41
42 void resetCounter() {
43   setState() {
44     counter = 0.0;
45     _valueNotifier.value = (counter / 33) * 100;
46   });
47 }
48
49 @override
50 Widget build(BuildContext context) {
51   SystemChrome.setSystemUIOverlayStyle(
52     const SystemUiOverlayStyle(statusBarColor: Colors.transparent));
53   return MaterialApp(
54     debugShowCheckedModeBanner: false,
55     theme: ThemeData(
56       colorScheme: ColorScheme.fromSeed(
57         seedColor: const Color.fromARGB(255, 119, 210, 145)), ColorScheme.from
58       useMaterial3: true,
59     ), ThemeData
60     home: Scaffold(
61       backgroundColor: const Color.fromARGB(255, 119, 210, 145),
62       body: SafeArea(
```

```

63     child: Center(
64       child: Column(
65         mainAxisAlignment: MainAxisAlignment.center,
66         children: [
67           Text(
68             '${counter.round()}',
69             style: const TextStyle(fontSize: 50),
70           ), Text
71           SimpleCircularProgressBar(
72             progressColors: [Colors.amberAccent.shade400],
73             size: 300,
74             progressStrokeWidth: 20,
75             backStrokeWidth: 10,
76             mergeNode: true,
77             maxValue: 100,
78             animationDuration: 0,
79             valueNotifier: _valueNotifier,
80             onGetText: (value) {
81               return Text(
82                 '${(value.toInt() / 3).round()}',
83                 style: const TextStyle(fontSize: 170),
84               ); Text
85             },
86           ), SimpleCircularProgressBar
87           const SizedBox(height: 50),
88           ClipRect(
89             borderRadius: const BorderRadius.all(Radius.circular(50)),
90             child: InkWell(
91               onTap: incrementCounter,
92               child: Container(
93                 decoration: const BoxDecoration(color: Colors.white),

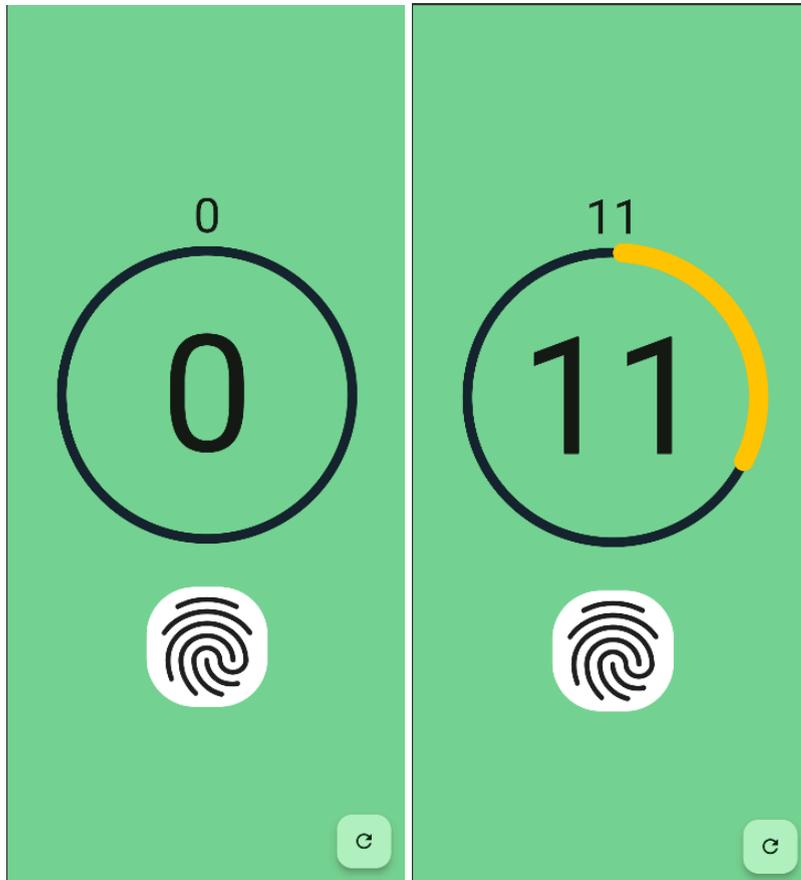
```

```

94             child: const Icon(
95               Icons.fingerprint,
96               size: 125,
97             ), Icon
98           ), Container
99         ), InkWell
100       ), ClipRect
101     ],
102   ), Column
103 ), Center
104 ), SafeArea
105 floatingActionButton: FloatingActionButton(
106   onPressed: resetCounter,
107   child: const Icon(Icons.refresh_outlined),
108 ), FloatingActionButton
109 ), Scaffold
110 ); MaterialApp
111 }
112 }
```

III. Screenshot Hasil

1. Tampilan Aplikasi:



IV. Kesimpulan

dapat disimpulkan bahwa *StatefulWidget* berperan penting dalam menciptakan antarmuka aplikasi yang interaktif dan dinamis di Flutter. Dengan menggunakan *setState()* dan *ValueNotifier*, perubahan data internal seperti hitungan tasbih dapat langsung tercermin pada tampilan aplikasi tanpa harus memulai ulang. Implementasi *simple_circular_progress_bar* juga memberikan representasi visual yang menarik dan informatif terhadap nilai hitungan. Selain itu, penggunaan *widget* seperti *InkWell* dan *FloatingActionButton* menunjukkan bagaimana Flutter memudahkan pembuatan elemen interaktif dengan desain modern. Konsep-konsep dasar ini menjadi fondasi penting dalam memahami manajemen *state* yang lebih kompleks pada pengembangan aplikasi Flutter selanjutnya.