

**LAPORAN TEORI MOBILE PROGRAMMING**  
**MODUL 1**



Nama : Erai Bagusalim  
NIM : 240605110088  
Kelas : Mobile Programming B  
Tanggal : 03 September 2025

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

## I. Tujuan

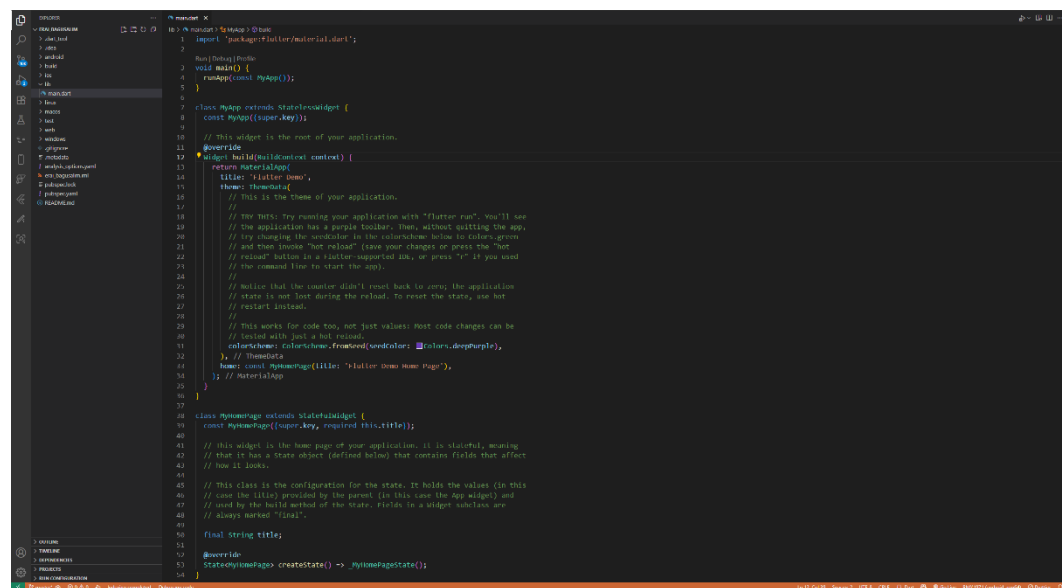
1. Memahami dan Mempelajari Flutter sebagai framework mobile cross-platform.
2. Melakukan instalasi *Flutter SDK*, *Android Studio*, *Android SDK*, serta plugin pendukung.
3. Mencoba menjalankan aplikasi Flutter pada emulator maupun perangkat Android (Smartphone).

## II. Langkah Kerja

1. Instalasi *Flutter SDK (Windows)* Pada *Code Editor Visual Studio Code*
2. Instalasi *Android Studio & Emulator*
3. Menjalankan aplikasi default pada emulator.
4. Memodifikasi teks *AppBar* dan tampilan “Hello World”.
5. Melakukan uji aplikasi pada smartphone melalui USB debugging.
6. Mencatat waktu eksekusi aplikasi.

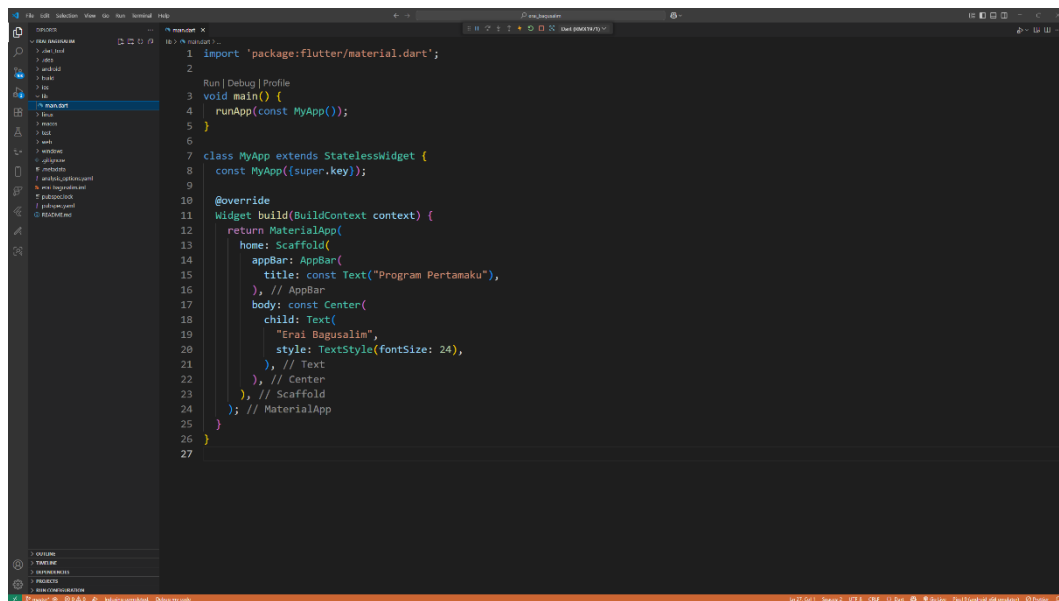
## III. Screenshot Hasil

### a. Menjalankan Program



```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9
10  // This widget is the root of your application.
11
12  @override
13  Widget build(BuildContext context) {
14    return MaterialApp(
15      title: 'Flutter Demo',
16      theme: ThemeData(
17        // This is the theme of your application.
18        //
19        // TRY THIS: Try running your application with "flutter run". You'll see
20        // the application has a purple toolbar. Then, without quitting the app,
21        // try changing the theme to the one below to colors.green.
22        // and then invoke "hot reload" (save your changes or press the "hot
23        // reload" button in a flutter-supported IDE) or press "r" if you used
24        // the command line to start the app.
25        //
26        // Notice that the counter didn't reset back to zero; the application
27        // state is not lost during the reload. To reset the state, use hot
28        // restart instead.
29        //
30        // This works for code too, not just values: Most code changes can be
31        // tested with just a hot reload.
32        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
33        useMaterial3: true,
34      ),
35    );
36  }
37
38  class MyHomePage extends StatefulWidget {
39    const MyHomePage({super.key, required this.title});
40
41    // This widget is the home page of your application. It is stateful, meaning
42    // that it has a state object (defined below) that contains fields that affect
43    // how it looks.
44
45    // This class is the configuration for the state, it holds the values (in this
46    // case the title) provided by the parent (in this case the MyApp widget) and
47    // used by the build method of the state, fields in a state class are
48    // always marked "final".
49
50    final String title;
51
52    @override
53    State<MyHomePage> createState() => _MyHomePageState();
54  }
55
56  class _MyHomePageState extends State<MyHomePage> {
57    // This widget represents the counter and can be manipulated by the user.
58    //
59    // TRY THIS: Try running your application with "flutter run". You'll see
60    // the application has a counter at the bottom of the screen. Try
61    // interacting with the counter by tapping on it.
```

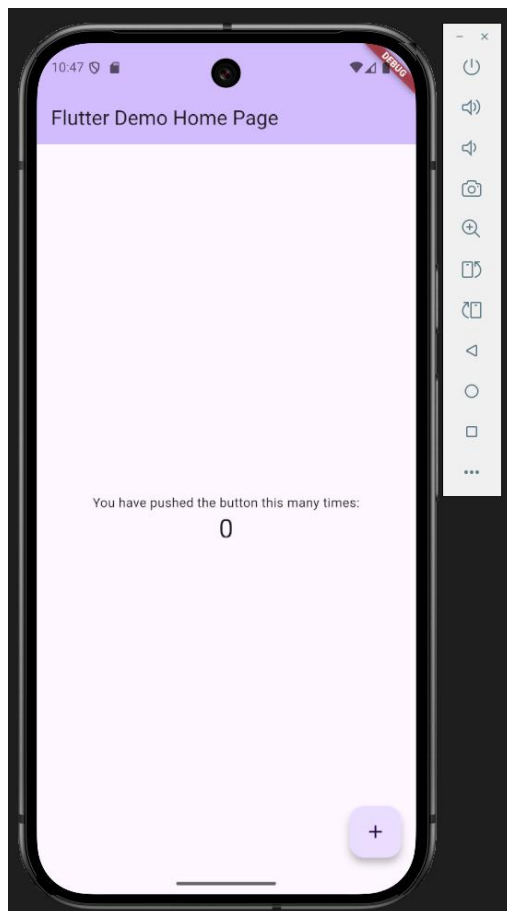
## b. Modifikasi Kode Program



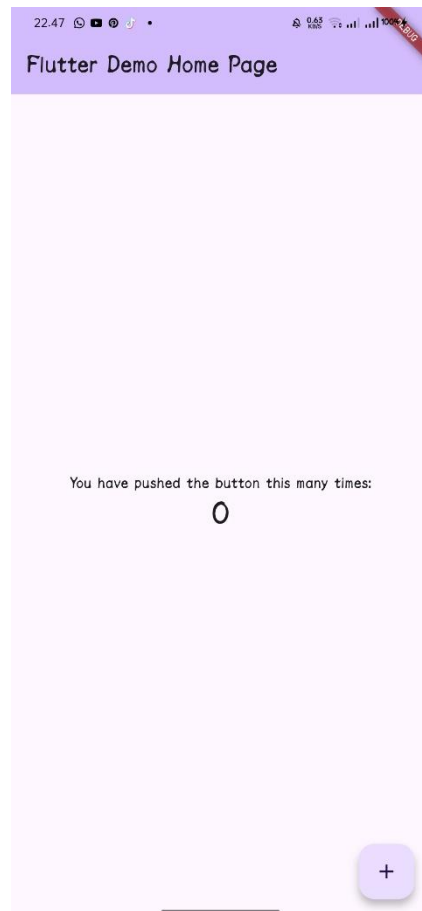
```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9
10  @override
11  Widget build(BuildContext context) {
12    return MaterialApp(
13      home: Scaffold(
14        appBar: AppBar(
15          title: const Text("Program Pertamaku"),
16        ), // AppBar
17        body: const Center(
18          child: Text(
19            "Erai Bagusali",
20            style: TextStyle(fontSize: 24),
21          ), // Text
22        ), // Center
23      ), // Scaffold
24    ); // MaterialApp
25  }
26 }
27
```

## c. Pengujian Pada Perangkat

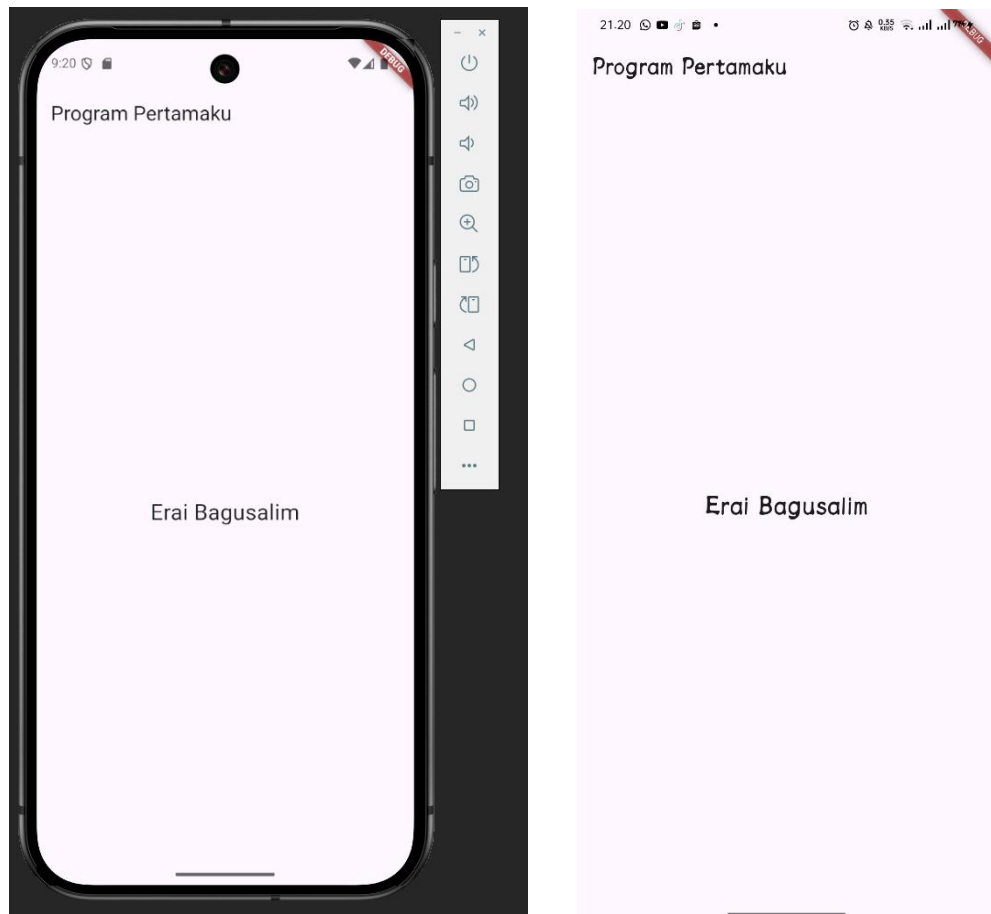
### 1. Tampilan aplikasi di emulator.



## 2. Tampilan aplikasi di smartphone Android.



3. Screenshot 3: Perubahan teks AppBar dan isi body setelah modifikasi.



d. Pengamatan Waktu Eksekusi

- Waktu eksekusi aplikasi pada *Emulator* tercatat sekitar **30 detik** sejak proses dijalankan hingga aplikasi tampil di layar.
- Waktu eksekusi aplikasi pada *Smartphone Android* tercatat sekitar **35 detik** sejak proses dijalankan hingga aplikasi tampil di layar.
- Dari hasil pengamatan tersebut dapat disimpulkan bahwa eksekusi pada *Emulator* sedikit lebih cepat dibandingkan pada *Smartphone Android*, meskipun perbedaan waktunya tidak terlalu signifikan.

#### IV. Kesimpulan

Dari laporan ini dapat disimpulkan bahwa Flutter merupakan framework mobile cross-platform yang memungkinkan pengembangan aplikasi Android dengan mudah dan efisien. Proses instalasi Flutter SDK, Android Studio, Android SDK, serta plugin pendukung berjalan dengan baik sehingga aplikasi default dapat dijalankan

pada emulator maupun perangkat smartphone Android. Melalui laporan ini, diperoleh pemahaman mengenai alur pembuatan dan pengujian aplikasi menggunakan Flutter, mulai dari tahap instalasi hingga uji coba langsung di perangkat.

Berdasarkan hasil pengujian performa, aplikasi pada emulator membutuhkan waktu sekitar **30 detik** untuk berjalan, sedangkan pada smartphone Android membutuhkan waktu sekitar **35 detik**. Hal ini menunjukkan bahwa meskipun keduanya dapat menjalankan aplikasi dengan baik, performa emulator relatif lebih cepat dibandingkan smartphone dalam kasus ini. Kedua waktu ini juga disebabkan oleh spesifikasi Komputer atau Laptop yang digunakan, jika spesifikasi yang digunakan lebih cepat maka hasil waktu yang didapat akan lebih cepat begitu juga sebaliknya.