

Aplikasi bernama “**Hello Flutter App**”, dengan tampilan simpel:

- Menampilkan teks sambutan.
  - Ada tombol untuk menghitung angka (counter).
- 

## 1. Struktur Project

Membuat project-nya dengan perintah:

```
flutter create hello_flutter_app
```

Lalu buka folder-nya di **Visual Studio Code**:

```
cd hello_flutter_app  
code .
```

---

## 2. Ganti kode di `lib/main.dart`

Hapus semua isi default, lalu ganti dengan kode berikut:

```
import 'package:flutter/material.dart';

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

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

    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            title: 'Hello Flutter App',
            theme: ThemeData(
                primarySwatch: Colors.blue,
                useMaterial3: true,
            ),
            home: const HomePage(),
            debugShowCheckedModeBanner: false,
        );
    }
}

class HomePage extends StatefulWidget {
    const HomePage({super.key});

    @override
    State<HomePage> createState() => _HomePageState();
}
```

```
class _HomePageState extends State<HomePage> {
    int counter = 0;

    void incrementCounter() {
        setState(() {
            counter++;
        });
    }

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: const Text('Demo Flutter App'),
                centerTitle: true,
            ),
            body: Center(
                child: Padding(
                    padding: const EdgeInsets.all(16.0),
                    child: Column(
                        mainAxisAlignment: MainAxisAlignment.center,
                        children: [
                            const Text(
                                'Selamat datang di Hello Flutter!',
                                style: TextStyle(
                                    fontSize: 22,
                                    fontWeight: FontWeight.bold,
                                ),
                                textAlign: TextAlign.center,
                            ),
                            const SizedBox(height: 20),
                            const Text(
                                'Tekan tombol di bawah untuk menambah angka:',
                                style: TextStyle(fontSize: 16),
                                textAlign: TextAlign.center,
                            ),
                            const SizedBox(height: 30),
                            Text(
                                '$counter',
                                style: const TextStyle(
                                    fontSize: 50,
                                    fontWeight: FontWeight.bold,
                                    color: Colors.blue,
                                ),
                            ),
                            const SizedBox(height: 30),
                            ElevatedButton.icon(
                                onPressed: incrementCounter,
                                icon: const Icon(Icons.add),
                                label: const Text('Tambah Angka'),
                                style: ElevatedButton.styleFrom(
                                    padding: const EdgeInsets.symmetric(horizontal: 24,
vertical: 12),
                                    textStyle: const TextStyle(fontSize: 18),
                                ),
                            ),
                        ],
                    ),
                ),
            ),
        );
    }
}
```

```
        ],
        ),
        ),
        );
}
}
```

---

## 3. Jalankan di HP atau Emulator

### Opsi A — Jalankan langsung (debug mode)

Sambungkan HP Android ke komputer (aktifkan *USB Debugging*), lalu ketik:

```
flutter run
```

### Opsi B — Buat file APK

Untuk membuat file .apk yang bisa instal di HP:

```
flutter build apk --release
```

Setelah selesai, file APK akan berada di:

```
build/app/outputs/flutter-apk/app-release.apk
```

Kirim file tersebut ke HP kamu dan instal secara manual.

---

## 4. Hasil Akhir

Aplikasi menampilkan teks:

“Selamat datang di Hello Flutter!”

Dan ada tombol “Tambah Angka” — setiap ditekan, angka di layar bertambah. Tampilannya responsif.

---

**Visual Studio Code (VSC)**, buat langkah-langkah **lengkap dan aman** supaya bisa langsung menjalankan dan build APK di HP Android.

---

## 1. Pastikan Persiapan Sudah Siap

Sebelum coding, pastikan sudah punya:

- Flutter** sudah terinstal

Cek dengan:

```
flutter doctor
```

- Android SDK & Device (HP / Emulator)**

- Kalau pakai HP asli → aktifkan *USB Debugging* di mode pengembang.
- Kalau pakai Emulator Android Studio → pastikan sudah aktif.

- VS Code Extensions**

Install ekstensi berikut:

- **Flutter**
  - **Dart**
- 

## 2. Buat Project Flutter Baru

Di terminal VS Code:

```
flutter create hello_flutter_app
```

Masuk ke folder project:

```
cd hello_flutter_app
```

Lalu buka project-nya:

```
code .
```

---

### Ganti Isi File `lib/main.dart`

Buka file `lib/main.dart`, hapus semua isinya, lalu salin kode berikut :

```
import 'package:flutter/material.dart';

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Hello Flutter App',
      theme: ThemeData(
        primarySwatch: Colors.teal,
        useMaterial3: true,
      ),
      home: const HomePage(),
      debugShowCheckedModeBanner: false,
    );
  }
}

class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  int counter = 0;

  void incrementCounter() {
    setState(() {
      counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Demo Flutter App'),
        centerTitle: true,
      ),
      body: Center(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              const Text(
                'Selamat Datang di Flutter!',
                style: TextStyle(
                  fontSize: 24,
                  fontWeight: FontWeight.bold,
                ),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

```
        ),
        textAlign: TextAlign.center,
    ),
    const SizedBox(height: 20),
    const Text(
        'Tekan tombol di bawah untuk menambah angka:',
        style: TextStyle(fontSize: 16),
        textAlign: TextAlign.center,
    ),
    const SizedBox(height: 30),
    Text(
        '$counter',
        style: const TextStyle(
            fontSize: 50,
            fontWeight: FontWeight.bold,
            color: Colors.teal,
        ),
    ),
    const SizedBox(height: 30),
    ElevatedButton.icon(
        onPressed: incrementCounter,
        icon: const Icon(Icons.add),
        label: const Text('Tambah Angka'),
        style: ElevatedButton.styleFrom(
            padding: const EdgeInsets.symmetric(horizontal: 24,
vertical: 12),
            textStyle: const TextStyle(fontSize: 18),
        ),
    ),
],
),
),
),
);
}
```

## **Jalankan Aplikasi di HP / Emulator**

**Jika pakai HP Android:**

1. Sambungkan HP ke laptop via kabel USB.
  2. Pastikan *USB Debugging* aktif.
  3. Ketik di terminal VS Code:  
4. flutter devices

(untuk memastikan HP terdeteksi)

5. Jalankan:
  6. flutter run

Aplikasi akan langsung terinstal dan berjalan di HP.

## 5. Build File APK Siap Instal

Setelah aplikasi sudah berjalan baik, buat APK-nya:

```
flutter build apk --release
```

Tunggu hingga selesai.

File APK akan muncul di:

```
build/app/outputs/flutter-apk/app-release.apk
```

Kirim file ini ke HP, lalu instal seperti biasa.

---

### Error :

```
PS E:\latihanflutter\hello_flutter_app> flutter build apk --release >> You are applying Flutter's
app_plugin_loader Gradle plugin imperatively using the apply script method, which is
deprecated and will be removed in a future release. Migrate to applying Gradle plugins with the
declarative plugins block: https://flutter.dev/to/flutter-gradle-plugin-apply FAILURE: Build
failed with an exception. * Where: Build file
'E:\latihanflutter\hello_flutter_app\android\build.gradle' line: 26 * What went wrong: A problem
occurred evaluating root project 'android'. > A problem occurred configuring project ':app'. >
Could not open cp_proj generic class cache for build file
'E:\latihanflutter\hello_flutter_app\android\app\build.gradle'
(C:\Users\majes\.gradle\caches\7.5\scripts\cvedoq6ndlsvd5jbqv7jwhlve). > BUG! exception in
phase 'semantic analysis' in source unit '_BuildScript_' Unsupported class file major version 65 *
Try: > Run with --stacktrace option to get the stack trace. > Run with --info or --debug option to
get more log output. > Run with --scan to get full insights. * Get more help at
https://help.gradle.org BUILD FAILED in 11s Running Gradle task 'assembleRelease'... 12.8s
└─ Flutter Fix
```

---

```
[!] Your project's Gradle version is incompatible with the Java version that Flutter is using for
Gradle. | | | | If you recently upgraded Android Studio, consult the migration guide at
https://flutter.dev/to/java-gradle-incompatibility. | | | | Otherwise, to fix this issue, first,
check the Java version used by Flutter by running flutter doctor --verbose. | | | | Then,
update the Gradle version specified in
E:\latihanflutter\hello_flutter_app\android\gradle\wrapper\gradle-wrapper.properties to be
compatible with that Java version. See | | the link below for more information on compatible
Java/Gradle versions: | | https://docs.gradle.org/current/userguide/compatibility.html#java | |
| | |
```

---

---

Gradle task assembleRelease failed with exit code 1

Error build APK di Flutter versi baru — penyebabnya jelas dari pesan ini:

Unsupported class file major version 65

dan

“Gradle version is incompatible with the Java version that Flutter is using for Gradle.”

Artinya:

**Java versi 21 atau lebih baru**, tapi **Gradle dan plugin Android di project** masih versi lama (tidak kompatibel).

---

## Solusi Lengkap untuk Flutter + VS Code

Langkah-langkah berikut akan memperbaiki error dan bisa build APK normal lagi.

---

### 1. Cek versi Java & Flutter

Ketik di terminal:

```
flutter doctor --verbose
```

Perhatikan bagian:

```
[√] Java version OpenJDK 21.0.2
```

Kalau versinya **Java 21**, kita harus pakai **Gradle 8.5 atau lebih baru**.

---

### 2. Buka file:

```
android/gradle/wrapper/gradle-wrapper.properties
```

Ubah baris terakhir menjadi:

```
distributionUrl=https\://services.gradle.org/distributions/gradle-8.6-all.zip
```

**Gradle 8.6** kompatibel dengan Java 21 dan Flutter terbaru.

---

### **3. Buka file:**

android/build.gradle

Cari bagian:

```
dependencies {  
    classpath 'com.android.tools.build:gradle:7.1.2'  
}
```

Ubah jadi:

```
dependencies {  
    classpath 'com.android.tools.build:gradle:8.4.0'  
}
```

---

### **4. Bersihkan dan Rebuild**

Setelah edit dua file tadi, jalankan perintah berikut di terminal VS Code:

```
flutter clean  
flutter pub get  
flutter build apk --release
```

---

### **5. Jika Masih Ada Warning “apply plugin deprecated”**

Abaikan dulu atau bisa migrasi dengan cara cepat:

Buka file android/app/build.gradle, lalu ubah bagian paling atas:  
Dari:

```
apply plugin: 'com.android.application'
```

Menjadi:

```
plugins {  
    id 'com.android.application'  
    id 'kotlin-android'  
}
```

---

## Hasil

Setelah langkah-langkah ini, APK kamu akan berhasil dibuat di:

```
build/app/outputs/flutter-apk/app-release.apk
```

---

## Error

```
PS E:\latihanflutter\hello_flutter_app> flutter doctor --verbose >> [√] Flutter (Channel stable, 3.27.0, on Microsoft Windows [Version 10.0.22000.2538], locale en-ID) • Flutter version 3.27.0 on channel stable at C:\Latflutter\flutter • Upstream repository https://github.com/flutter/flutter.git • Framework revision 8495dee1fd (11 months ago), 2024-12-10 14:23:39 -0800 • Engine revision 83bacfc525 • Dart version 3.6.0 • DevTools version 2.40.2 [√] Windows Version (Installed version of Windows is version 10 or higher) [√] Android toolchain - develop for Android devices (Android SDK version 36.1.0) • Android SDK at C:\Users\majes\AppData\Local\Android\sdk • Platform android-36, build-tools 36.1.0 • Java binary at: C:\Program Files\Android\Android Studio\jbr\bin\java • Java version OpenJDK Runtime Environment (build 21.0.8+14196175-b1038.72) • All Android licenses accepted. [√] Chrome - develop for the web • Chrome at C:\Program Files\Google\Chrome\Application\chrome.exe [!] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.14.4) • Visual Studio at C:\Program Files\Microsoft Visual Studio\2022\Community • Visual Studio Community 2022 version 17.14.36202.13 X Visual Studio is missing necessary components. Please re-run the Visual Studio installer for the "Desktop development with C++" workload, and include these components: MSVC v142 - VS 2019 C++ x64/x86 build tools - If there are multiple build tool versions available, install the latest C++ CMake tools for Windows Windows 10 SDK [√] Android Studio (version 2025.2.1) • Android Studio at C:\Program Files\Android\Android Studio • Flutter plugin can be installed from: https://plugins.jetbrains.com/plugin/9212-flutter • Dart plugin can be installed from: https://plugins.jetbrains.com/plugin/6351-dart • Java version OpenJDK Runtime Environment (build 21.0.8+14196175-b1038.72) [√] VS Code (version 1.105.1) • VS Code at C:\Users\majes\AppData\Local\Programs\Microsoft VS Code • Flutter extension version 3.122.0 [√] Connected device (3 available) • Windows (desktop) • windows • windows-x64 • Microsoft Windows [Version 10.0.22000.2538] • Chrome (web) • chrome • web-javascript • Google Chrome 141.0.7390.123 • Edge (web) • edge • web-javascript • Microsoft Edge 141.0.3537.99 [√] Network resources • All expected network resources are available. ! Doctor found issues in 1 category. PS E:\latihanflutter\hello_flutter_app>
```

Laporan flutter doctor --verbose sangat jelas, dan masalah utamanya sudah ketahuan :

Kamu menggunakan **Java 21.0.8**

Tapi **project Flutter** kamu masih pakai **Gradle lama (di bawah 8.2)** yang **tidak kompatibel dengan Java 21.**

Itulah sebabnya muncul error:

```
Unsupported class file major version 65
```

Jadi, solusinya adalah **update file Gradle** supaya sesuai dengan Flutter 3.27 + Java 21.

---

## Langkah Perbaikan Lengkap

Berikut 3 file yang perlu diubah di folder `android/` agar build APK bisa berhasil

---

### 1. File: `android/gradle/wrapper/gradle-wrapper.properties`

Ganti seluruh isi menjadi:

```
# Gradle wrapper configuration
distributionBase=GRADLE_USER_HOME
distributionPath=wrapper/dists
zipStoreBase=GRADLE_USER_HOME
zipStorePath=wrapper/dists
distributionUrl=https\://services.gradle.org/distributions/gradle-8.6-all.zip
```

- Gradle 8.6 sudah 100% kompatibel dengan Java 21 dan Flutter 3.27

---

### 2. File: `android/build.gradle`

Ganti isinya seperti ini (ini versi modern yang stabil untuk Flutter 3.27):

```
buildscript {
    repositories {
        google()
        mavenCentral()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:8.4.2'
        classpath 'org.jetbrains.kotlin:kotlin-gradle-plugin:1.9.22'
    }
}

allprojects {
    repositories {
        google()
        mavenCentral()
    }
}

rootProject.buildDir = "../build"
```

```
subprojects {
    project.buildDir = "${rootProject.buildDir}/${project.name}"
    project.evaluationDependsOn(':app')
}

tasks.register("clean", Delete) {
    delete rootProject.buildDir
}
```

---

### 3. File: android/app/build.gradle

Pastikan bagian atasnya seperti ini:

```
plugins {
    id 'com.android.application'
    id 'kotlin-android'
}

android {
    namespace "com.example.hello_flutter_app"
    compileSdk 34

    defaultConfig {
        applicationId "com.example.hello_flutter_app"
        minSdk 21
        targetSdk 34
        versionCode 1
        versionName "1.0"
    }

    buildTypes {
        release {
            minifyEnabled false
            shrinkResources false
            signingConfig signingConfigs.debug
        }
    }
}

dependencies {
    implementation "org.jetbrains.kotlin:kotlin-stdlib:1.9.22"
}
```

---

### 4. Bersihkan dan Build Ulang

Setelah selesai mengedit 3 file tadi, jalankan perintah berikut di terminal VS Code:

```
flutter clean
flutter pub get
flutter build apk --release
```

---

## 5. Lokasi File APK

Setelah build sukses, file APK siap instal ada di:

```
build/app/outputs/flutter-apk/app-release.apk
```

Bisa langsung salin ke HP Android dan instal manual.

---

android/gradle/wrapper/gradle-wrapper.properties ini **dimana ?**

File gradle-wrapper.properties memang **sering bikin bingung posisi folder-nya**, terutama buat yang **baru main di Flutter**.

Berikut penjelasannya :

---

### Lokasi File `gradle-wrapper.properties`

Path lengkapnya ada di dalam folder proyek Flutter :

```
E:\latihanflutter\hello_flutter_app\android\gradle\wrapper\gradle-wrapper.properties
```

💡 Jadi dari project :

1. Buka folder proyek hello\_flutter\_app di **VS Code**.
2. Lalu buka folder:
  3. android → gradle → wrapper
  4. Di situ akan ada file bernama:
  5. gradle-wrapper.properties

Struktur folder :

```
hello_flutter_app/
  └── android/
    ├── app/
    ├── gradle/
    │   └── wrapper/
    │       └── gradle-wrapper.properties  ← ini
    ├── build.gradle
    ├── settings.gradle
    └── ...
  └── lib/
    └── main.dart
  pubspec.yaml
```

---

## Langkah Praktis di VS Code

1. Di **Explorer panel** (kiri), klik folder android.
2. Klik panah kecil di gradle → wrapper.
3. Klik file gradle-wrapper.properties.
4. Ganti seluruh isinya dengan ini

```
# Gradle wrapper configuration
distributionBase=GRADLE_USER_HOME
distributionPath=wrapper/dists
zipStoreBase=GRADLE_USER_HOME
zipStorePath=wrapper/dists
distributionUrl=https\://services.gradle.org/distributions/gradle-8.6-all.zip
```

Lalu **Save (Ctrl + S)**

---

PS E:\latihanflutter\hello\_flutter\_app> flutter build apk --release [!] Your app is using an unsupported Gradle project. To fix this problem, create a new project by running flutter create -t app <app-directory> and then move the dart code, assets and pubspec.yaml to the new project.

Pesan ini sangat jelas:

! “Your app is using an unsupported Gradle project.”

Artinya struktur project Android-mu (folder android/) sudah **terlalu lama / tidak cocok dengan Flutter versi baru (3.27 + Gradle 8.x + Java 21)**.

Biasanya ini terjadi kalau project Flutter dibuat di versi lama (misalnya Flutter 2.x atau 3.0), lalu dijalankan di versi terbaru tanpa update struktur Android-nya.

---

## Solusi Terbaik & Aman

Buat project Flutter baru yang bersih dan kompatibel dengan Flutter 3.27, lalu pindahkan file Dart & asset kamu ke sana.

Berikut langkah yang teruji :

---

## 1. Buat project Flutter baru

Jalankan di terminal VS Code:

```
flutter create hello_flutter_fixed
```

Ini akan membuat project Flutter baru dengan struktur Gradle dan Android terbaru.

---

## 2. Pindahkan file penting dari project lama

Buka dua folder:

- E:\latihanflutter\hello\_flutter\_app
- E:\latihanflutter\hello\_flutter\_fixed

Lalu salin file dan folder berikut dari proyek lama ke proyek baru:

Dari folder lama	Ke folder baru
lib/	lib/ (replace seluruh isinya)
assets/ ( <i>jika ada</i> )	assets/
pubspec.yaml	Ganti isi pubspec.yaml di proyek baru
analysis_options.yaml ( <i>jika ada</i> )	Ganti juga
File tambahan lain (misal: .dart_tool tidak perlu)	✗ jangan disalin

---

## 3. Jalankan perintah Flutter

Masuk ke project baru:

```
cd E:\latihanflutter\hello_flutter_fixed
```

Lalu jalankan:

```
flutter clean  
flutter pub get  
flutter build apk --release
```

🎉 Hasilnya:

```
build/app/outputs/flutter-apk/app-release.apk
```

Sudah bisa diinstal di HP Android tanpa error lagi.

---

## **Kenapa cara ini disarankan?**

Karena:

- Flutter 3.27 pakai sistem Gradle baru (`plugins {}` block, bukan `apply plugin:` lagi).
  - Plugin Android lama tidak bisa otomatis di-*migrate* oleh Flutter.
  - Dengan membuat project baru, semua dependency, Gradle, dan file konfigurasi sudah sesuai versi terbaru.
- 

### **Error**

```
PS E:\> cd : latihanflutter\hello_flutter_fixed >> Set-Location : A positional parameter cannot be found that accepts argument 'latihanflutter\hello_flutter_fixed'. At line:1 char:1 + cd : latihanflutter\hello_flutter_fixed + ~~~~~+ CategoryInfo : InvalidArgument: () [Set-Location], ParameterBindingException + FullyQualifiedErrorId : PositionalParameterNotFound,Microsoft.PowerShell.Commands.SetLocationCommand
```

---

Pesan error ini berasal dari **PowerShell**, bukan dari Flutter-nya.

Masalahnya sederhana: menulis `cd : latihanflutter\hello_flutter_fixed` dengan tanda titik dua (:) setelah `cd`, padahal PowerShell menganggapnya perintah lain.

---

## **✓ Cara yang Benar**

Gunakan salah satu cara di bawah ini (tanpa tanda :) 

### **Opsi 1 — Jalur relatif**

Kalau folder ada di drive E::

```
cd E:\latihanflutter\hello_flutter_fixed
```

### **Opsi 2 — Kalau sudah di drive E:**

Pastikan di drive E, lalu ketik:

```
E:  
cd latihanflutter\hello_flutter_fixed
```

### Opsi 3 — Gunakan kutip (kalau ada spasi di nama folder)

Kalau nama folder ada spasi, tulis begini:

```
cd "E:\latihanflutter\hello flutter fixed"
```

---

### Cek Posisi Folder

Setelah itu, pastikan benar-benar sudah masuk ke folder proyek:

Output-nya harus seperti ini:

```
Path
-----
E:\latihanflutter\hello_flutter_fixed
```

Kalau sudah benar, lanjutkan dengan:

```
flutter pub get
flutter run
```

atau langsung build:

```
flutter build apk --release
```

---

### Error

```
PS E:\latihanflutter\hello_flutter_app\hello_flutter_fixed> pwd >> Path -----
E:\latihanflutter\hello_flutter_app\hello_flutter_fixed PS
E:\latihanflutter\hello_flutter_app\hello_flutter_fixed>
```

Sudah berada di folder project Flutter baru:

```
E:\latihanflutter\hello_flutter_app\hello_flutter_fixed
```

Sekarang tinggal beberapa langkah lagi agar proyek ini bisa **jalan lancar dan build APK tanpa error.**

---

## 1. Pastikan Project Sudah Siap

Ketik di terminal:

```
flutter pub get
```

Perintah ini akan mengambil semua dependency dari file `pubspec.yaml`.

---

## 2. Jalankan Aplikasi (Debug Mode Dulu)

Untuk memastikan tidak ada error sebelum build APK:

```
flutter run
```

- Jika menghubungkan HP Android dengan *USB Debugging*, app langsung tampil di HP.
  - Kalau pakai emulator, pastikan sudah dijalankan.
- 

## 3. Build File APK Siap Instal

Setelah berjalan dengan baik, lanjutkan:

```
flutter build apk --release
```

File hasilnya nanti ada di:

```
build/app/outputs/flutter-apk/app-release.apk
```

---

## Jika Masih Kosong (Belum Ada File Dart-nya)

Bisa langsung isi file `lib/main.dart` dengan kode sederhana berikut agar aplikasi demo-nya bisa tampil cantik:

**File:** `lib/main.dart`

```
import 'package:flutter/material.dart';

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

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

    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            title: 'Hello Flutter Fixed',
            theme: ThemeData(
                colorScheme: ColorScheme.fromSeed(seedColor: Colors.indigo),
                useMaterial3: true,
            ),
            debugShowCheckedModeBanner: false,
            home: const HomePage(),
        );
}
```

```

    );
}
}

class HomePage extends StatefulWidget {
const HomePage({super.key});

@Override
State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
int counter = 0;

@Override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: const Text('Demo Flutter Fixed'),
centerTitle: true,
),
body: Center(
child: Column(
mainAxisAlignment: MainAxisAlignment.center,
children: [
const Text(
'Halo Flutter!',
style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
),
const SizedBox(height: 16),
Text(
'Angka: $counter',
style: const TextStyle(fontSize: 32, color: Colors.indigo),
),
const SizedBox(height: 24),
ElevatedButton.icon(
onPressed: () {
setState(() {
counter++;
});
},
icon: const Icon(Icons.add),
label: const Text('Tambah'),
)
],
),
);
}
}

```

---

## 4. Tes Build

Setelah isi file `main.dart` di atas, jalankan perintah:

```
flutter run
```

Jika tampil normal, lanjutkan:

```
flutter build apk --release
```

---

Berikut 5 **jenis aplikasi Flutter** yang *paling banyak dibutuhkan tahun 2025*, lengkap dengan penjelasan dan tingkat kesulitannya 

---

## 1. Aplikasi Kasir / Point of Sale (POS)

*Digunakan oleh toko, kafe, laundry, warung, dll.*

**Fitur utama:**

- Login pengguna (admin & kasir)
- Tambah produk & stok
- Transaksi penjualan
- Laporan harian & bulanan
- Cetak struk (optional Bluetooth printer)

**Tingkat kesulitan:**  Sedang

**Teknologi:** Flutter + MySQL / Firebase

**Kelebihan:** Banyak dicari UMKM

---

## 2. Aplikasi Absensi Sekolah / Pegawai

*Bisa dipakai di sekolah, kampus, kantor.*

**Fitur utama:**

- Login siswa/guru/pegawai
- Absensi dengan QR Code / lokasi GPS
- Laporan kehadiran per tanggal
- Dashboard admin

**Tingkat kesulitan:**  Sedang

**Kelebihan:** Dapat dipakai banyak instansi

---

## 3. Aplikasi E-Commerce Sederhana

*Toko online untuk jual produk digital / fisik.*

**Fitur utama:**

- Katalog produk

- Keranjang belanja
- Checkout + pembayaran (manual / Midtrans)
- Dashboard admin (CRUD produk & pesanan)

**Tingkat kesulitan:** ♦ Menengah ke atas

**Kelebihan:** Bisa dikembangkan jadi bisnis nyata

---

#### 4. Aplikasi Booking / Reservasi

*Contoh: booking kendaraan, salon, lapangan, dokter.*

**Fitur utama:**

- Login pengguna
- Pilih layanan/jadwal
- Booking otomatis ke database
- Dashboard admin untuk konfirmasi

**Tingkat kesulitan:** ♦ Sedang

**Kelebihan:** Praktis dan banyak ide turunan

---

#### 5. Aplikasi Chatting Lokal

*Seperti WhatsApp versi mini dengan database lokal.*

**Fitur utama:**

- Login pengguna
- Chat realtime antar user (Socket.io / Firebase)
- Notifikasi pesan masuk
- Status online/offline

**Tingkat kesulitan:** ♦ Menengah

**Kelebihan:** Sangat menarik untuk demo & portofolio

---

Kalau kamu mau **demo untuk dipasang di HP**, saya saranin mulai dari nomor **2 (Absensi Sekolah)** atau **3 (E-Commerce)** karena:

- Mudah dibuat
- Keren untuk ditunjukkan
- Bisa dikembangkan jadi project besar

---

## Tugas

-  Aplikasi **Absensi Sekolah**
-  Aplikasi **E-Commerce**
-  Aplikasi **Kasir Toko**
-  Aplikasi **Booking**
-  Aplikasi **Chatting**