FLUTTER

MEMBUAT USER INTERFACE

CONTENTS

01 Mengenal Widget-widget Dasar Dalam Flutter

02 Mengclone Project Flutter Dari Github

03 Mempelajari Cara Menggunakan Layout Widgets

04 Mempelajari Cara Menggunakan File Pubspec.Yaml

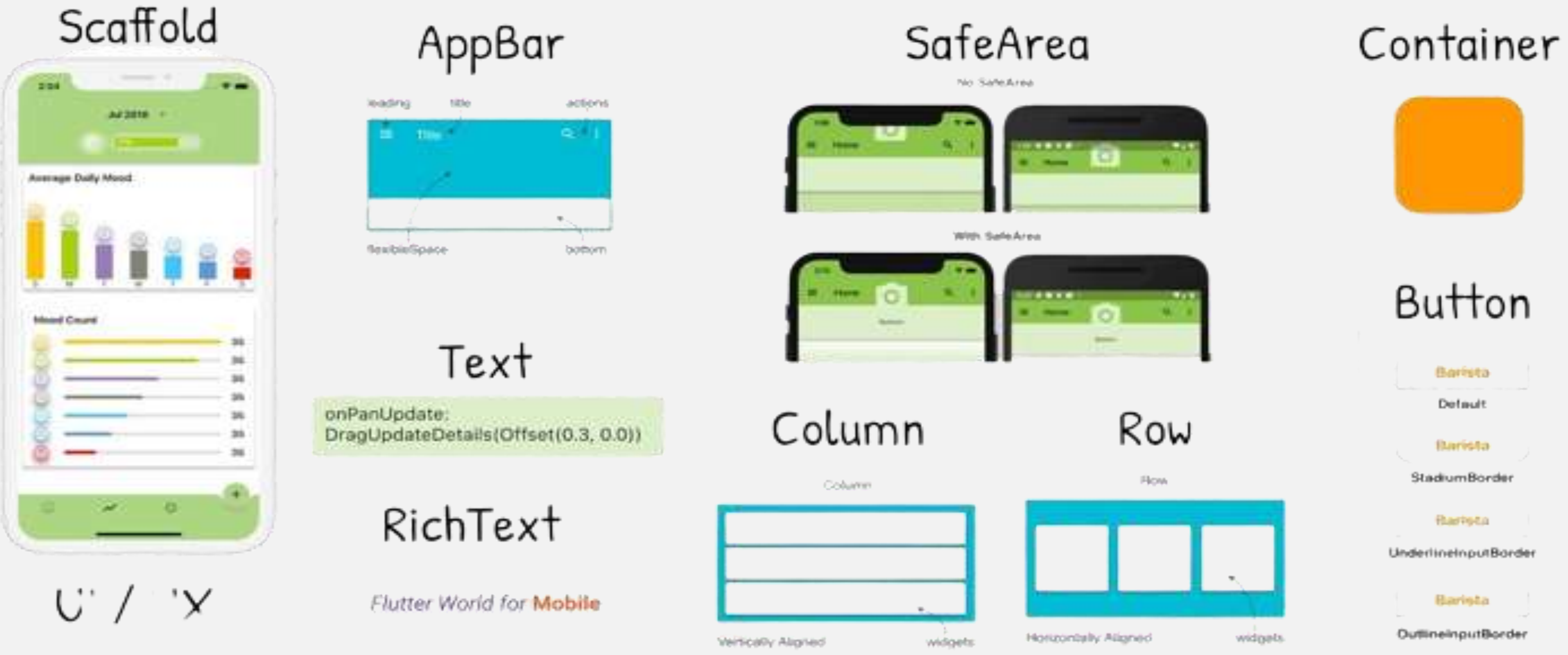
05 Mempelajari Cara Memasukkan App Icon

one

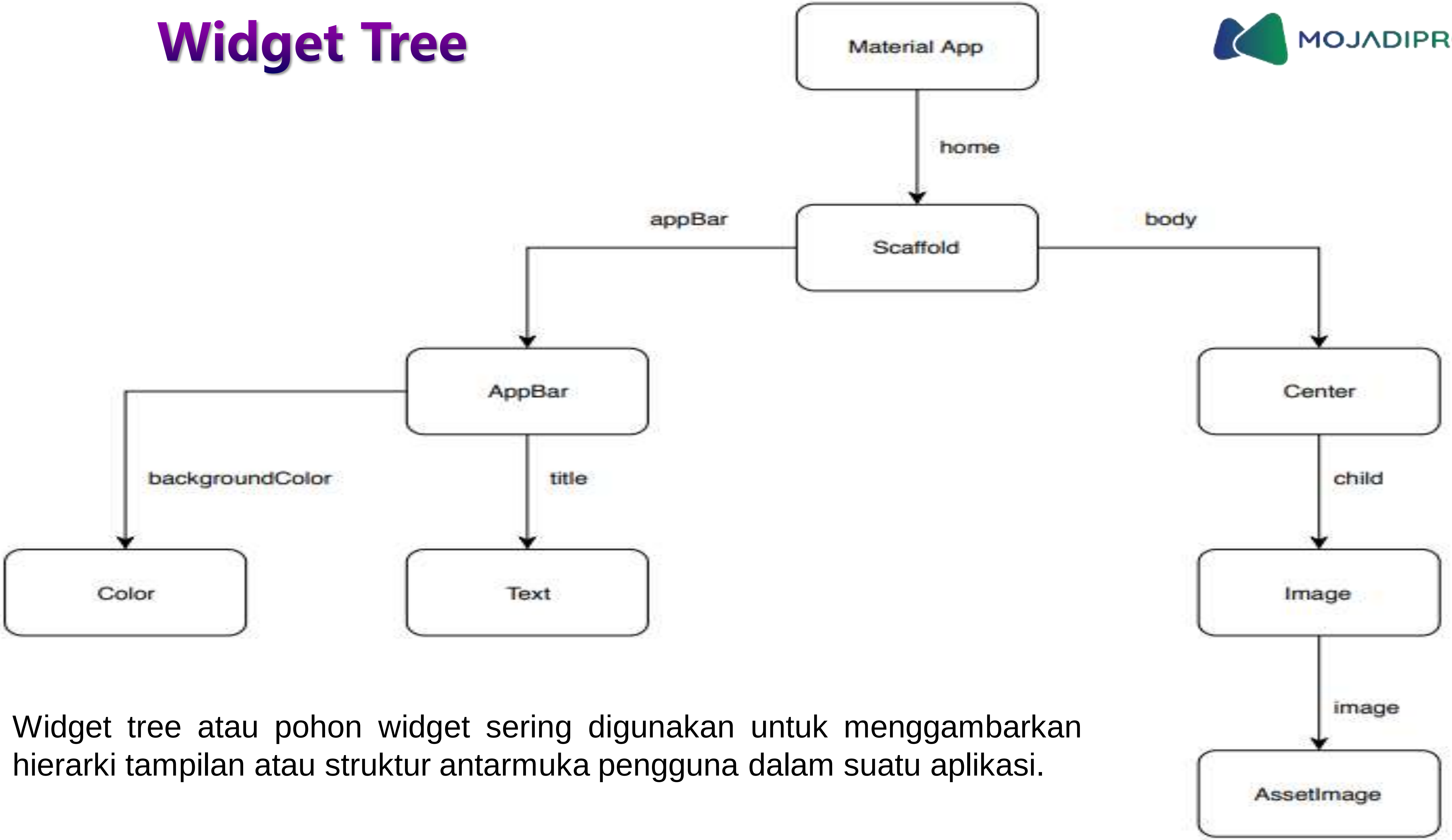
Mengenal Widget-widget dasar dalam Flutter

Widget

Widget merupakan objek visual yang merepresentasikan sebuah bagian dari tampilan aplikasi. Widget dapat berupa tombol, teks, gambar, kotak, atau apapun yang dapat dilihat pada layar.



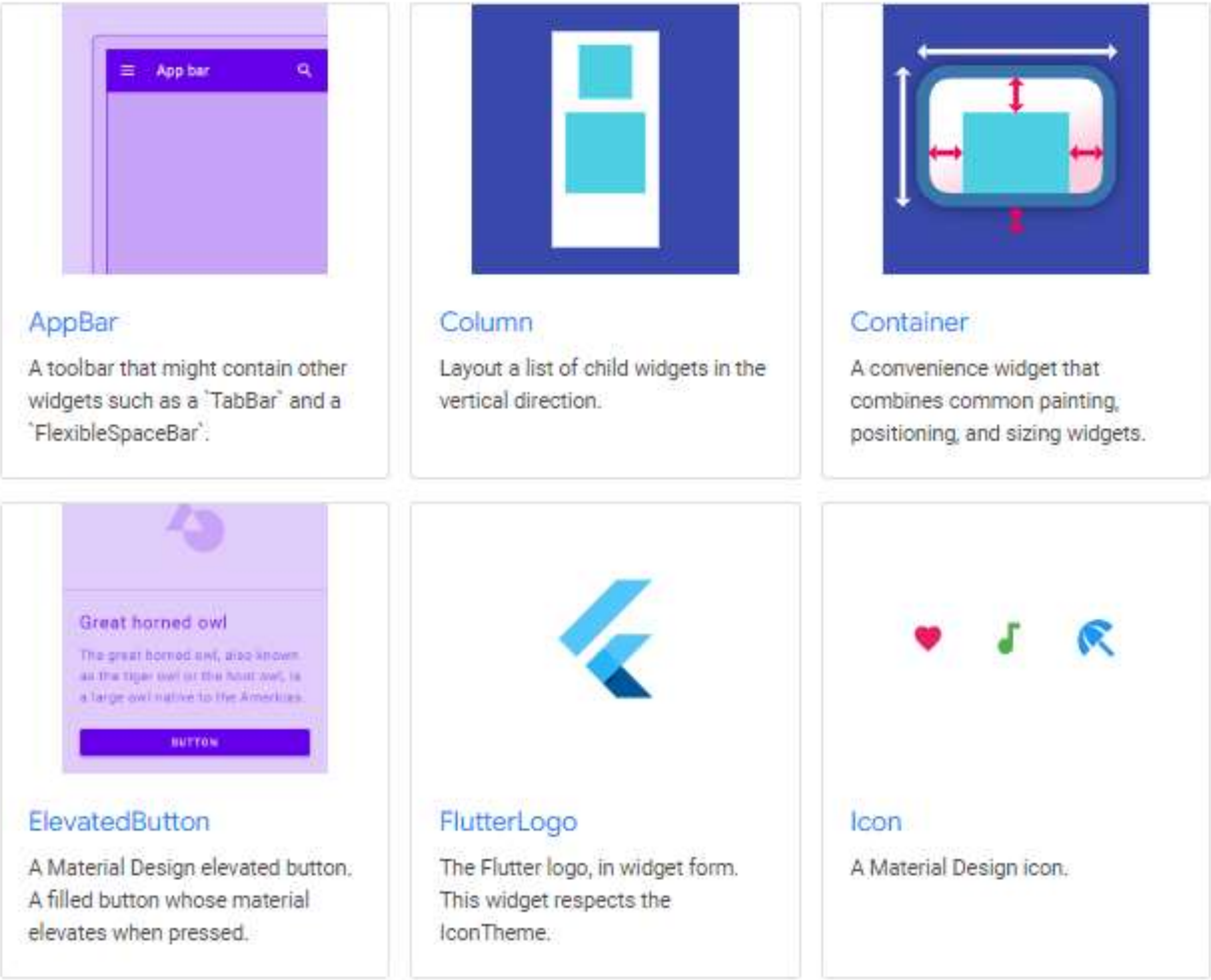
Widget Tree



Widget tree atau pohon widget sering digunakan untuk menggambarkan hierarki tampilan atau struktur antarmuka pengguna dalam suatu aplikasi.

Material App

MaterialApp adalah kelas atau widget yang menjadi komponen utama atau inti dari aplikasi flutter. Widget yang dapat diakses menggunakan kelas MaterialApp :



AppBar

A toolbar that might contain other widgets such as a `TabBar` and a `FlexibleSpaceBar`.

Column

Layout a list of child widgets in the vertical direction.

Container

A convenience widget that combines common painting, positioning, and sizing widgets.

ElevatedButton

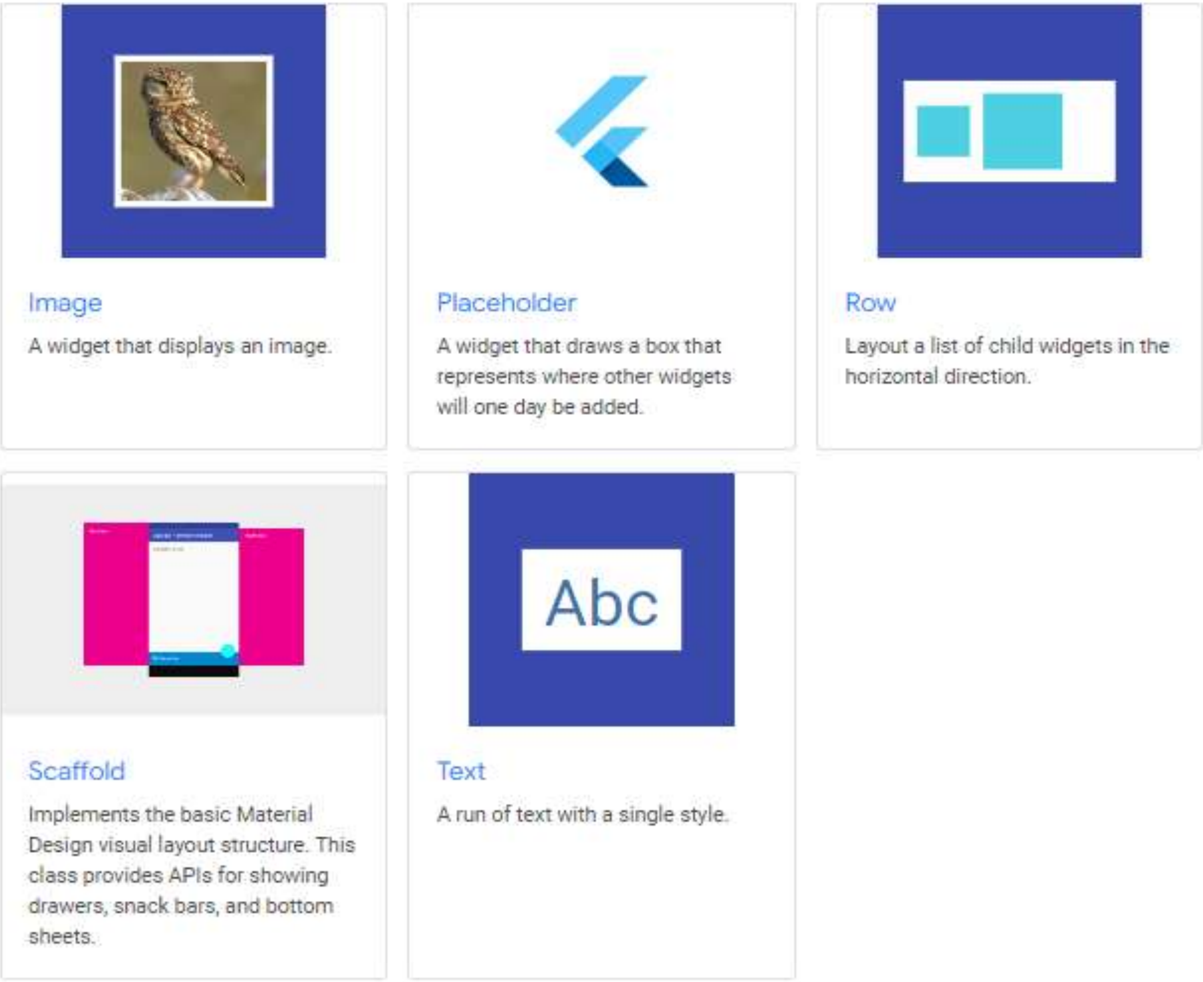
A Material Design elevated button. A filled button whose material elevates when pressed.

FlutterLogo

The Flutter logo, in widget form. This widget respects the IconTheme.

Icon

A Material Design icon.



Image

A widget that displays an image.

Placeholder

A widget that draws a box that represents where other widgets will one day be added.

Row

Layout a list of child widgets in the horizontal direction.

Scaffold

Implements the basic Material Design visual layout structure. This class provides APIs for showing drawers, snack bars, and bottom sheets.

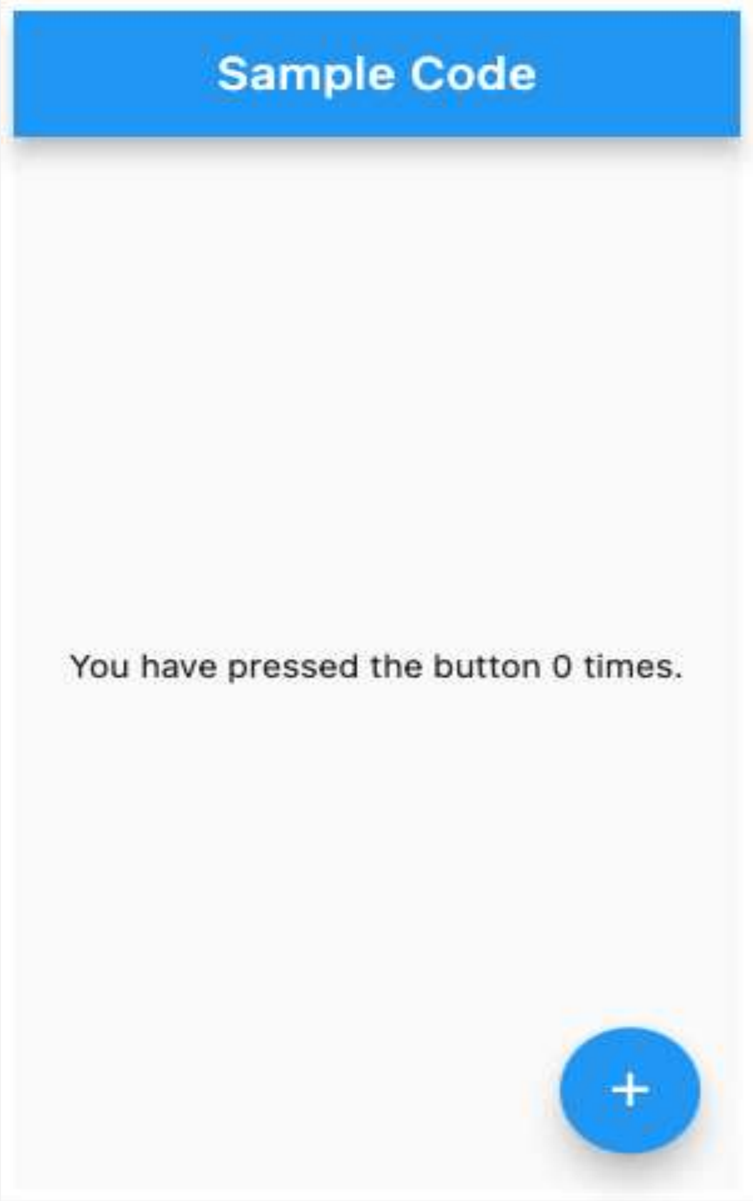
Text

A run of text with a single style.

Scaffold

Di dalam MaterialApp terdapat scaffold class. Scaffold adalah widget yang menyediakan kerangka kerja dasar untuk membangun antarmuka pengguna (UI) aplikasi.

Widget Scaffold menawarkan struktur tata letak yang umum digunakan dalam aplikasi, termasuk App bar, body, drawer, floating action button, dan bottom navigation bar.


Sample Code

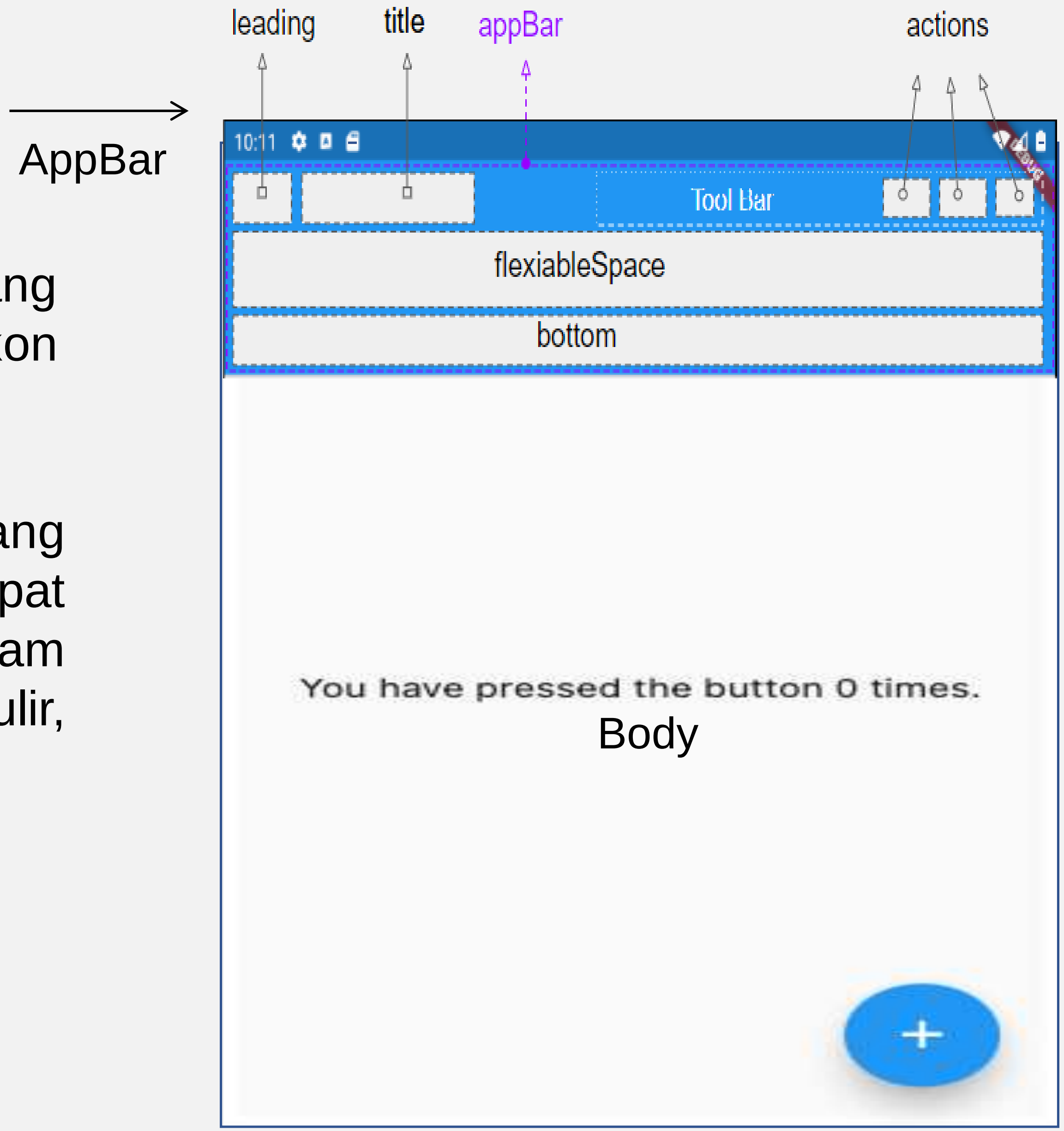
You have pressed the button 0 times.

Scaffold

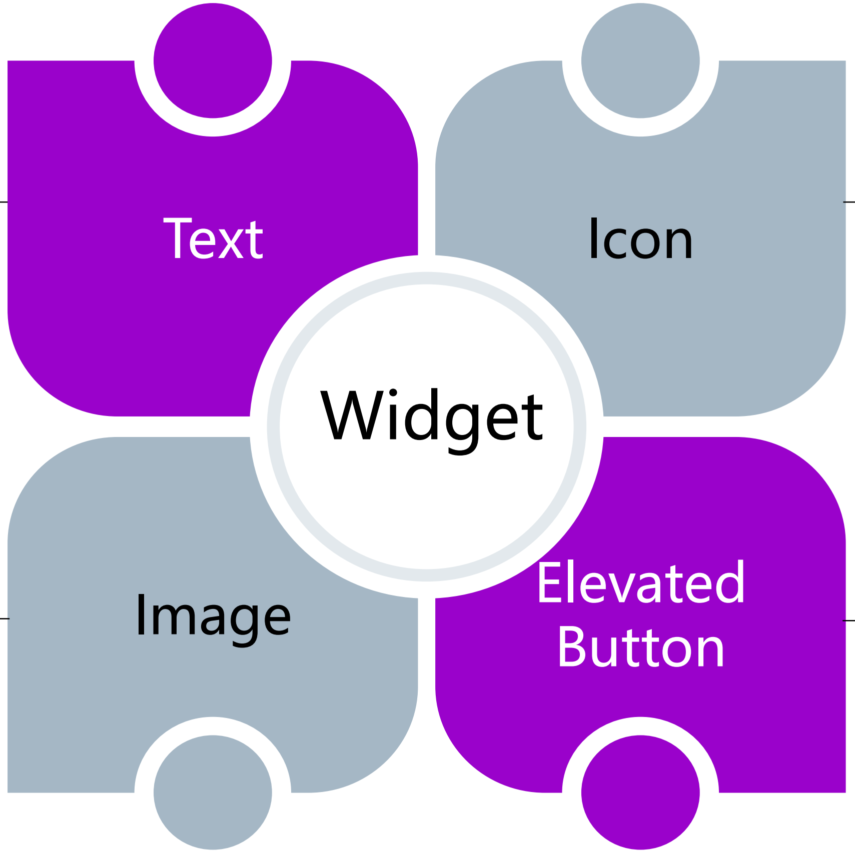
Di dalam Scaffold terdapat Appbar dan Body

01 Appbar adalah bagian atas aplikasi yang biasanya berisi judul, ikon kembali, ikon aksi, dan opsi menu.

02 Body adalah konten utama aplikasi yang ditampilkan di bawah App bar. Kita dapat menempatkan berbagai widget di dalam body untuk menampilkan informasi, formulir, daftar, atau tampilan lainnya.



Berikut adalah beberapa widget dasar yang perlu diketahui



Widget

Text

Digunakan untuk menampilkan teks. Anda dapat mengatur properti seperti style, font, dan alignment.

Icon

Suatu simbol atau gambar kecil yang digunakan untuk mewakili suatu objek, aksi, atau konsep tertentu.

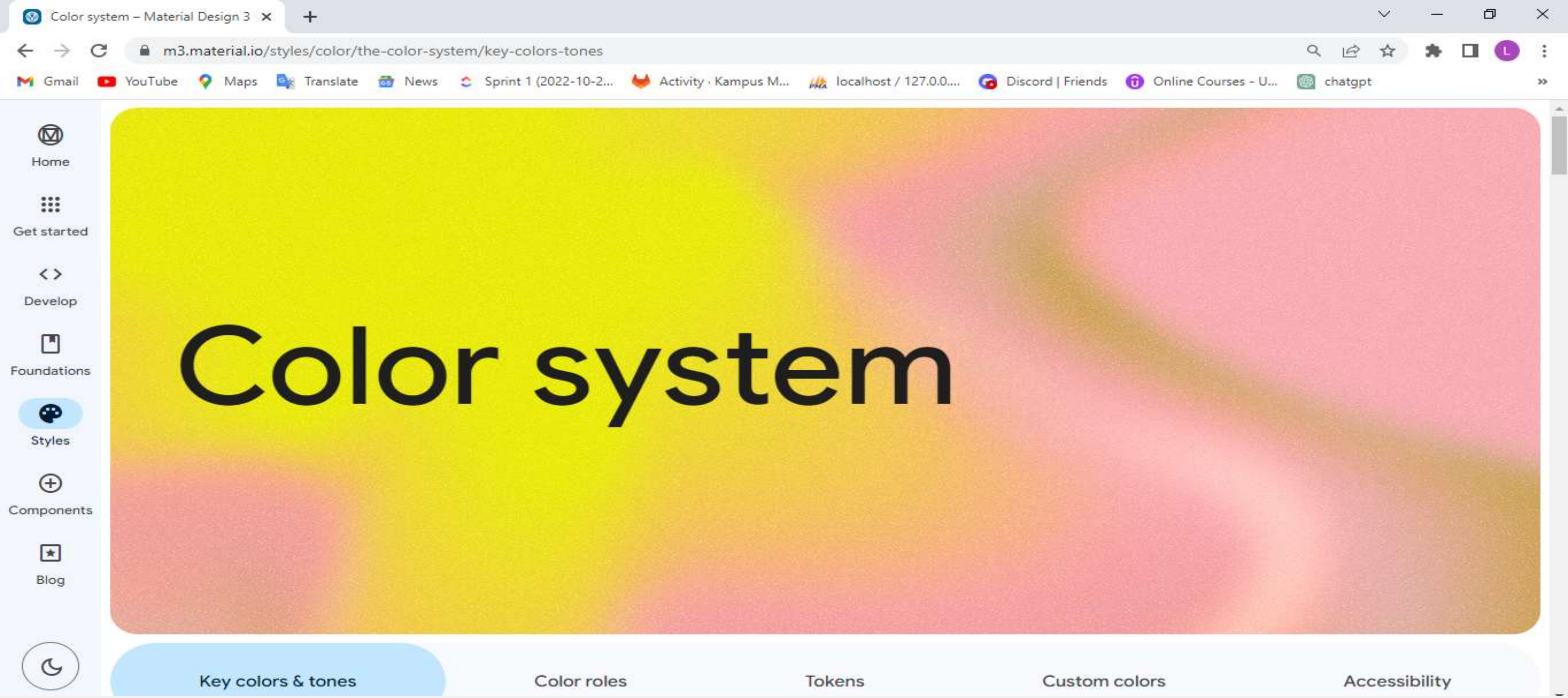
Image

Digunakan untuk menampilkan gambar. Anda dapat menentukan gambar dari file lokal atau URL.

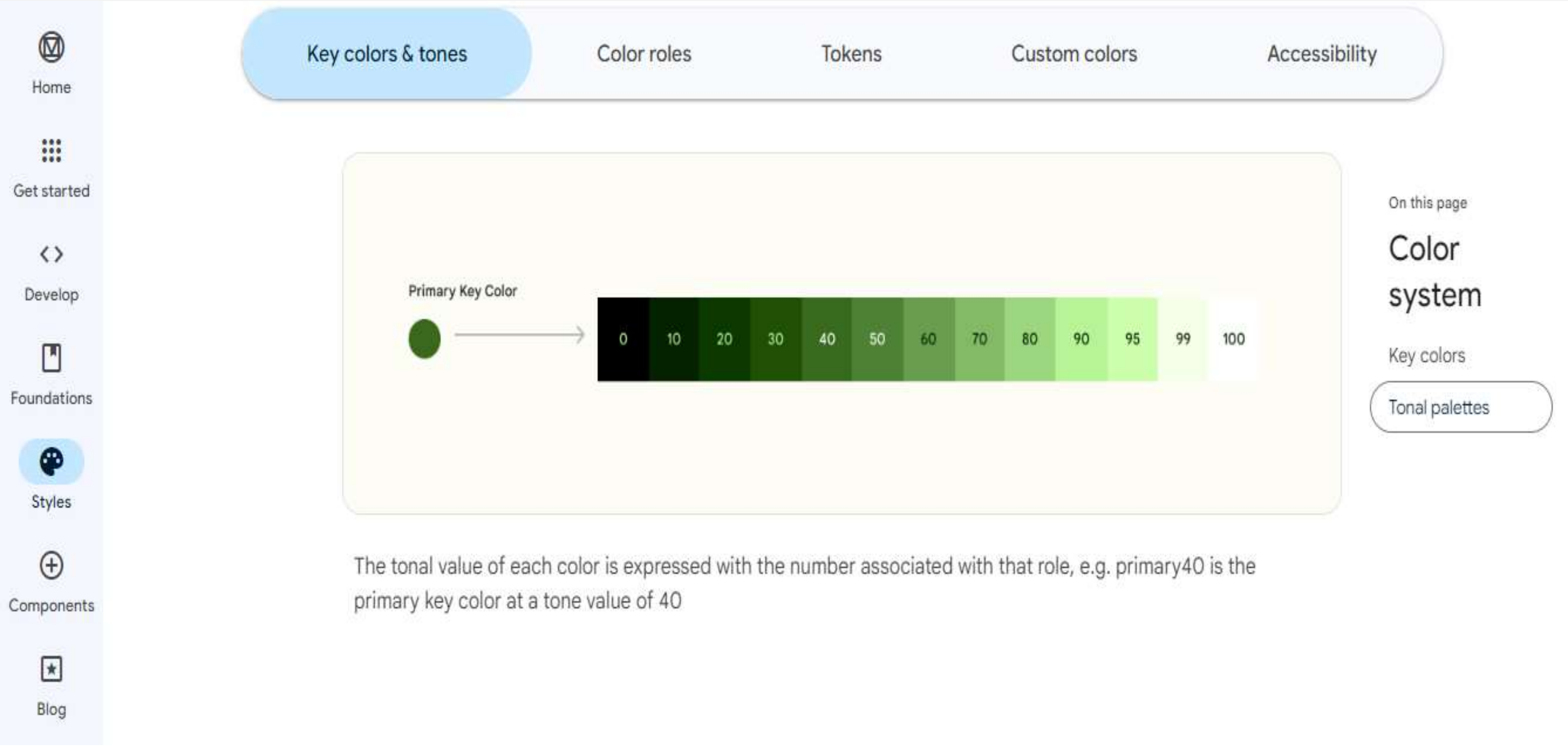
Elevated Button

Digunakan untuk membuat tombol dengan efek tampilan yang meningkat (elevated) atau menonjol.

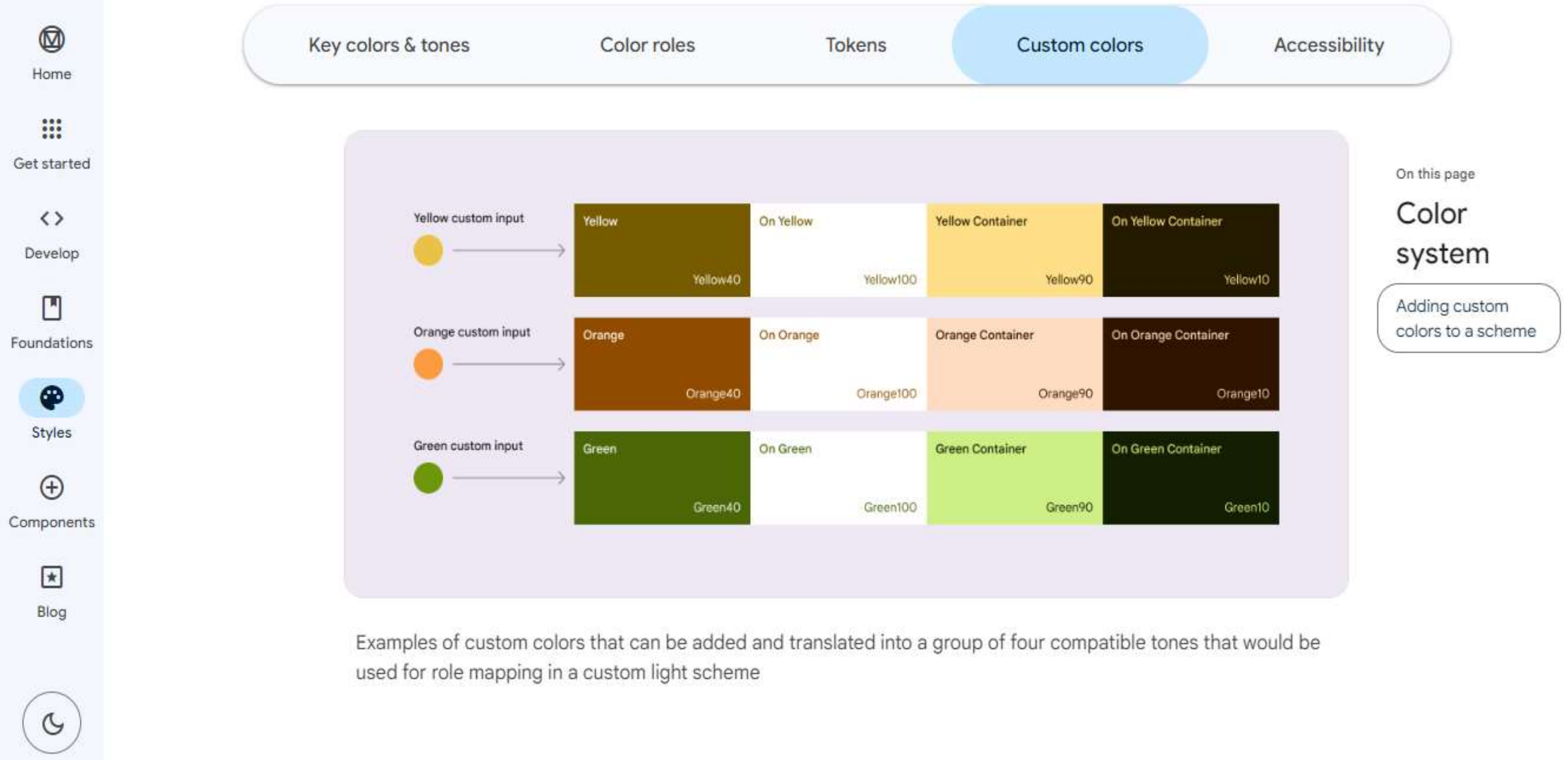
Penggunaan Custom Warna pada Material Design



Buka material design di Google → styles → color → color system



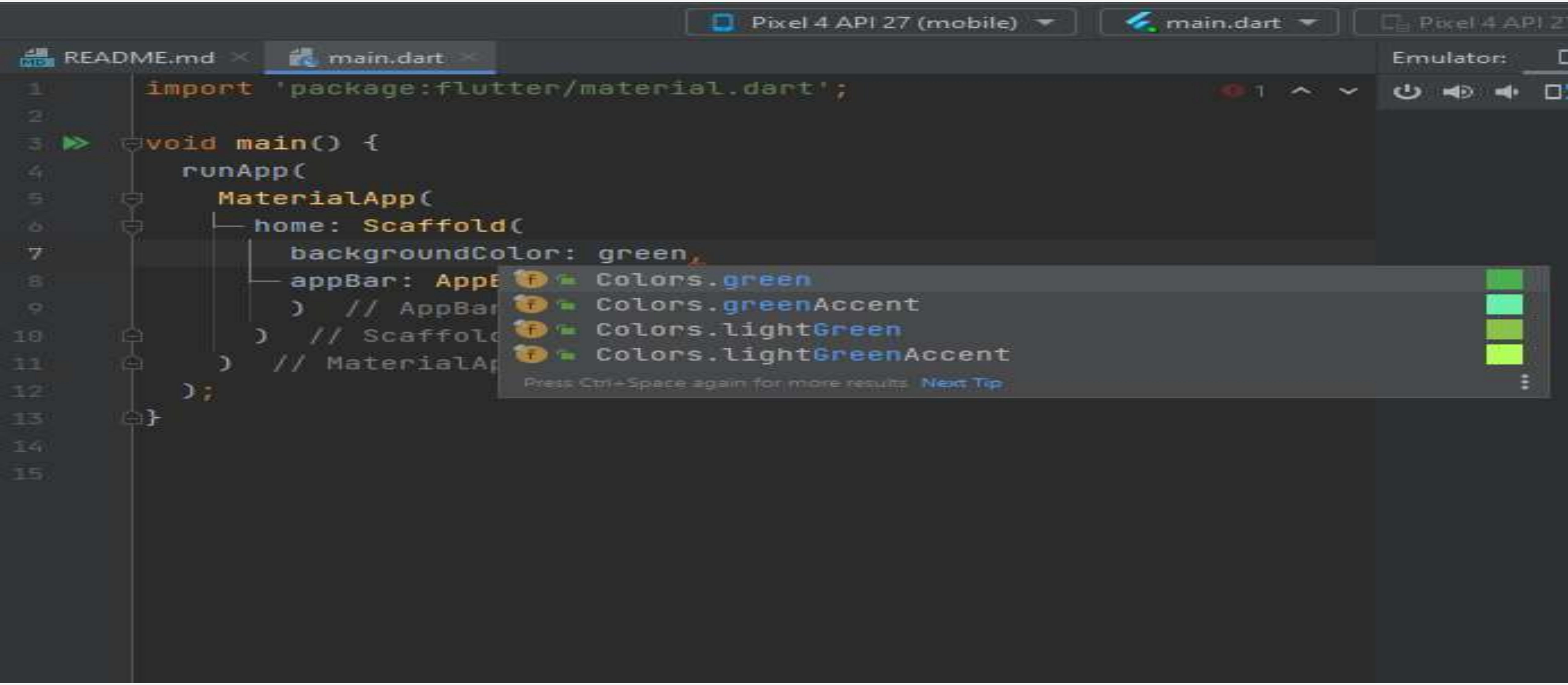
The tonal value of each color is expressed with the number associated with that role, e.g. primary40 is the primary key color at a tone value of 40



Yellow custom input	Yellow Yellow40	On Yellow Yellow100	Yellow Container Yellow90	On Yellow Container Yellow10
Orange custom input	Orange Orange40	On Orange Orange100	Orange Container Orange90	On Orange Container Orange10
Green custom input	Green Green40	On Green Green100	Green Container Green90	On Green Container Green10

Examples of custom colors that can be added and translated into a group of four compatible tones that would be used for role mapping in a custom light scheme

Contoh Kode Penggunaan Warna



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

void main() {
  runApp(
    MaterialApp(
      home: Scaffold(
        backgroundColor: green,
        appBar: AppE
        ) // AppBar
      ) // Scaffolc
    ) // MaterialAp
  );
}
```

```
Colors.green
Colors.greenAccent
Colors.lightGreen
Colors.lightGreenAccent
Press Ctrl+Space again for more results  Next Tip
```

Anda bisa menuliskan nama warna, atau dengan menambahkan angka. Contoh : Colors.green40

Contoh Kode Penggunaan Warna

Berikut ini adalah beberapa contoh kode penggunaan warna pada Flutter :

1. Menggunakan kode warna HEX :

```
Color myColor = Color(0xFF00FF00); // Contoh warna hijau (0xFF00FF00)
```

2. Menggunakan kode warna RGB :

```
Color myColor = Color.fromRGBO(255, 0, 0, 1); // Contoh warna merah (RGB: 255, 0, 0)
```

3. Menggunakan nama warna primitif :

```
Color myColor = Colors.blue; // Contoh warna biru
```

Contoh Kode Penggunaan Warna

Berikut ini adalah beberapa contoh kode penggunaan warna pada Flutter :

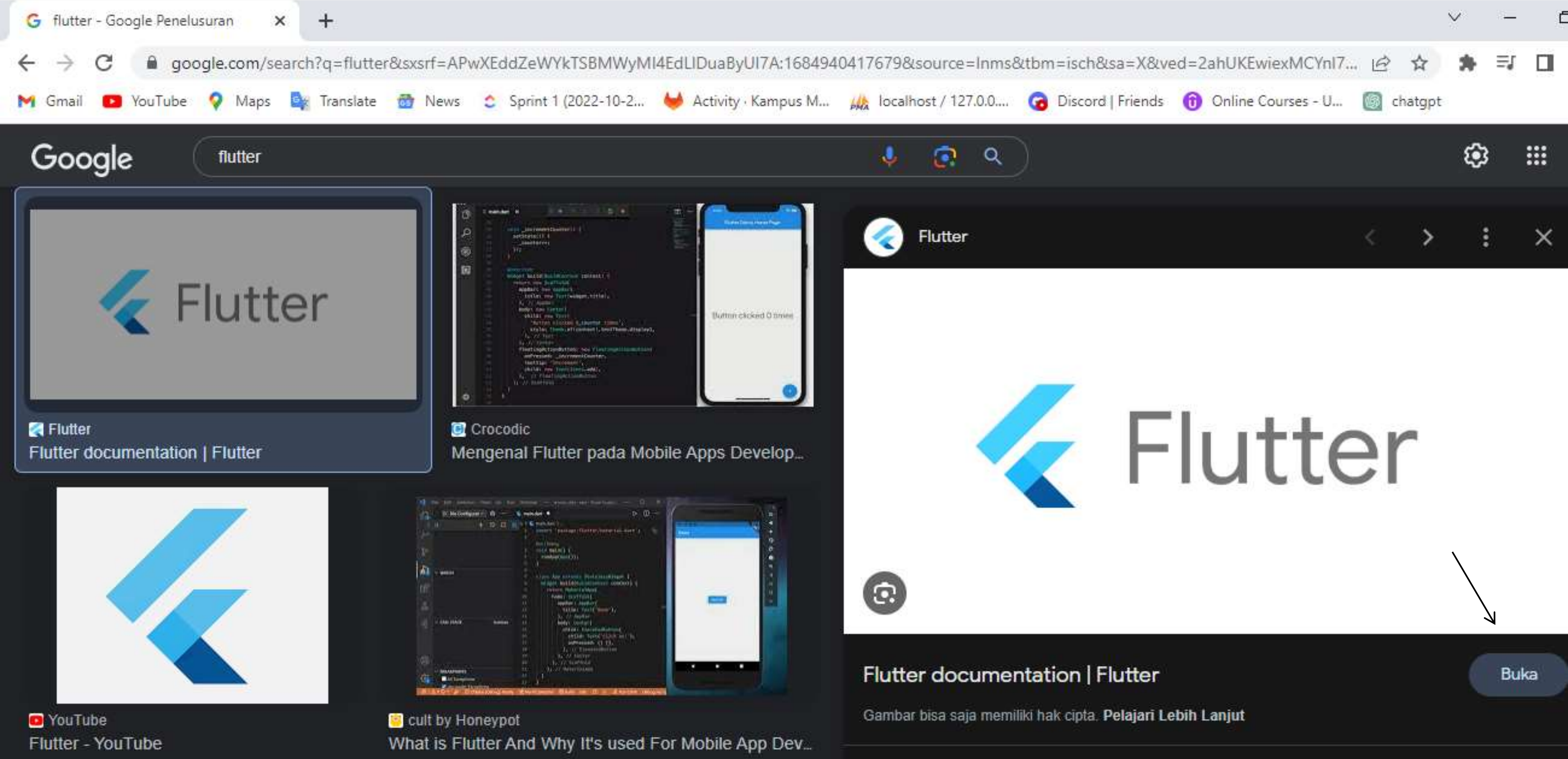
4. Menggunakan Gradient (Gradien) :

```
Container(
  decoration: BoxDecoration(
    gradient: LinearGradient(
      colors: [Colors.red, Colors.blue],
      begin: Alignment.topCenter,
      end: Alignment.bottomCenter,
    ),
  ),
),
```

5. Menggunakan Material Design Colors:

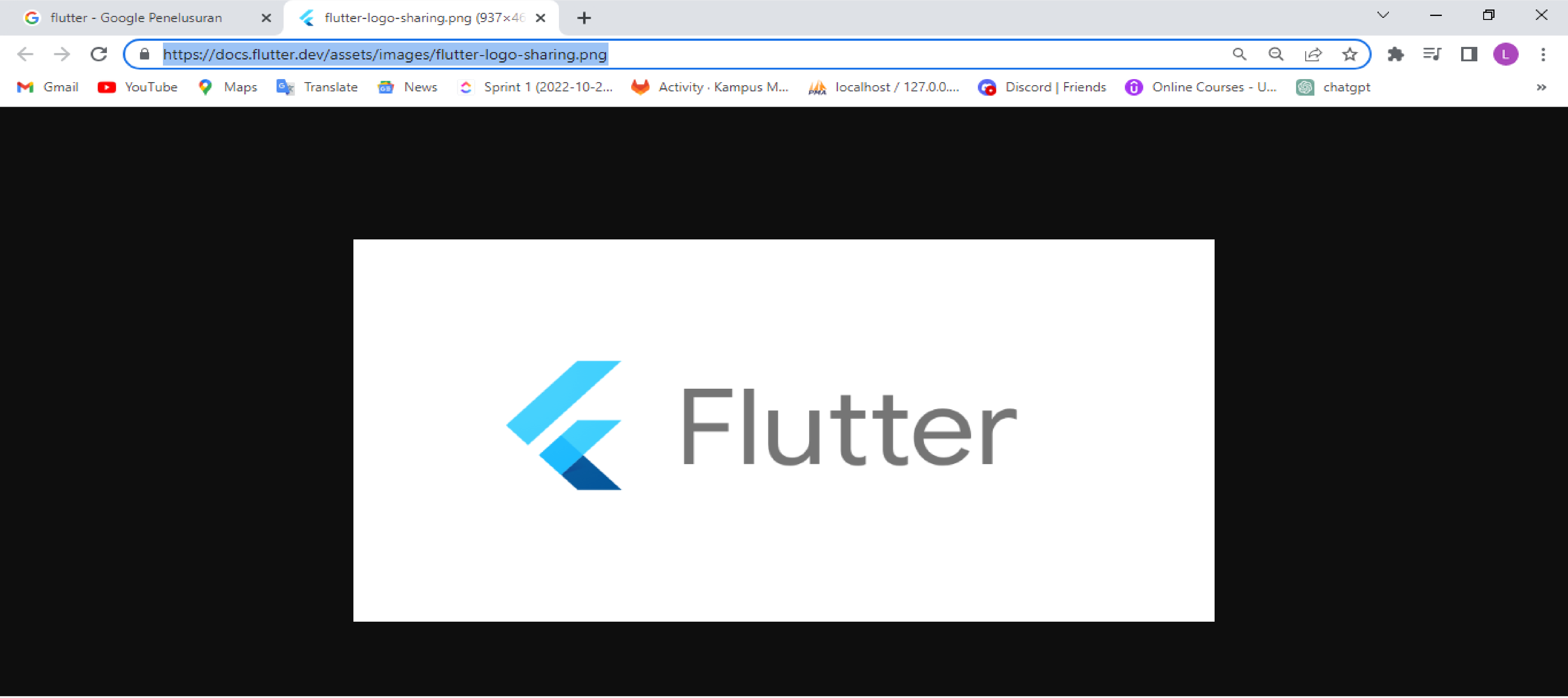
```
import 'package:flutter/material.dart';

Color myColor = Colors.teal; // Contoh warna teal
```

Memasukkan Network Image


flutter - Google Penelusuran
google.com/search?q=flutter&sxsrf=APwXEddZeWYkTSBMWyMI4EdLIDuaByUI7A:1684940417679&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiexMCYnl7...
Gmail
YouTube
Maps
Translate
News
Sprint 1 (2022-10-2...
Activity · Kampus M...
localhost / 127.0.0....
Discord | Friends
Online Courses - U...
chatgpt
Google
flutter
Flutter
Flutter
Flutter documentation | Flutter
Crocodic
Mengenal Flutter pada Mobile Apps Develop...
YouTube
Flutter - YouTube
cult by Honeypot
What is Flutter And Why It's used For Mobile App Dev...
Flutter
Flutter
Flutter documentation | Flutter
Buka

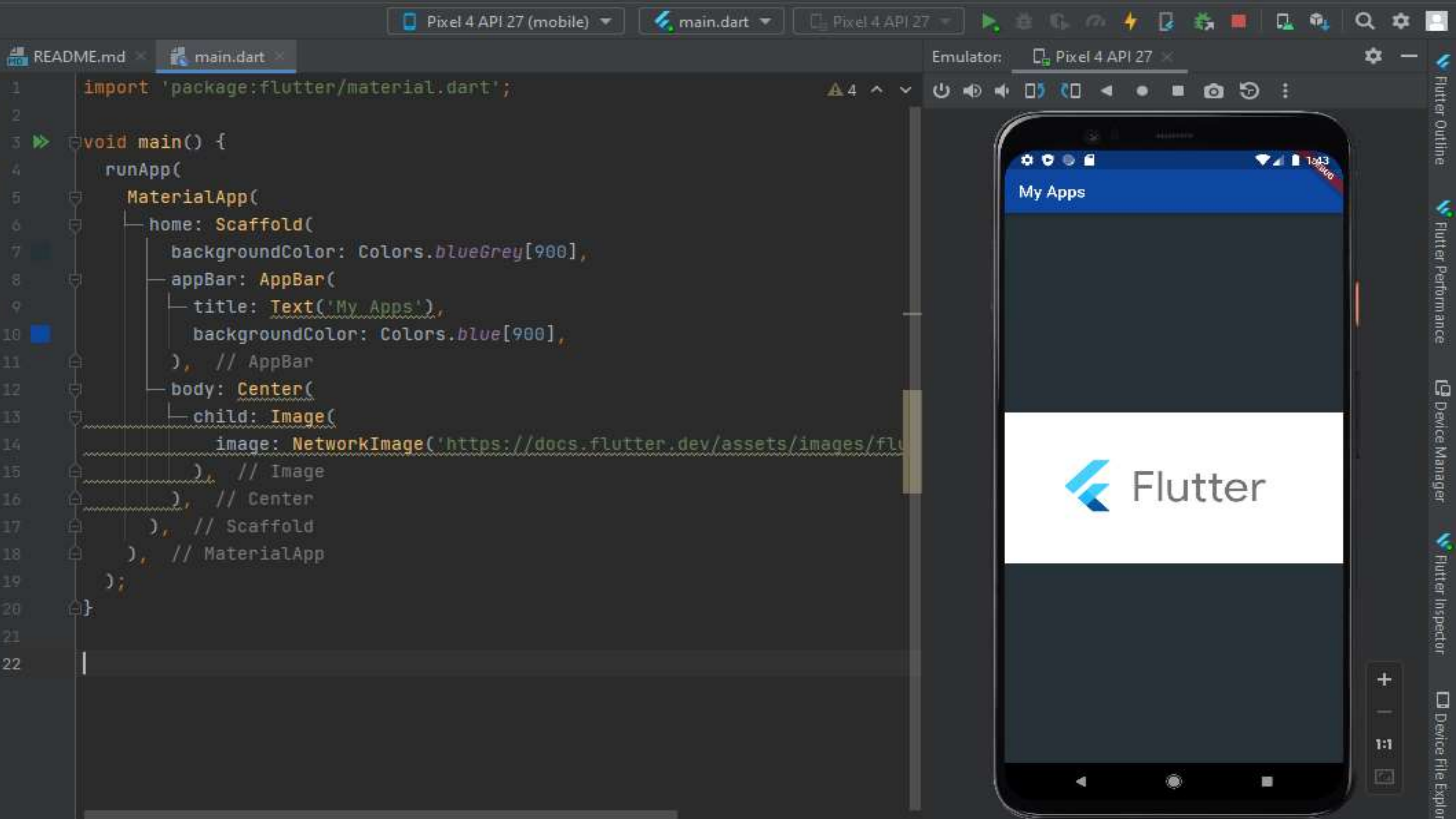
Gambar bisa saja memiliki hak cipta. **Pelajari Lebih Lanjut**

Memasukkan Network Image



https://docs.flutter.dev/assets/images/flutter-logo-sharing.png

Copy URL dan masukkan ke dalam kode



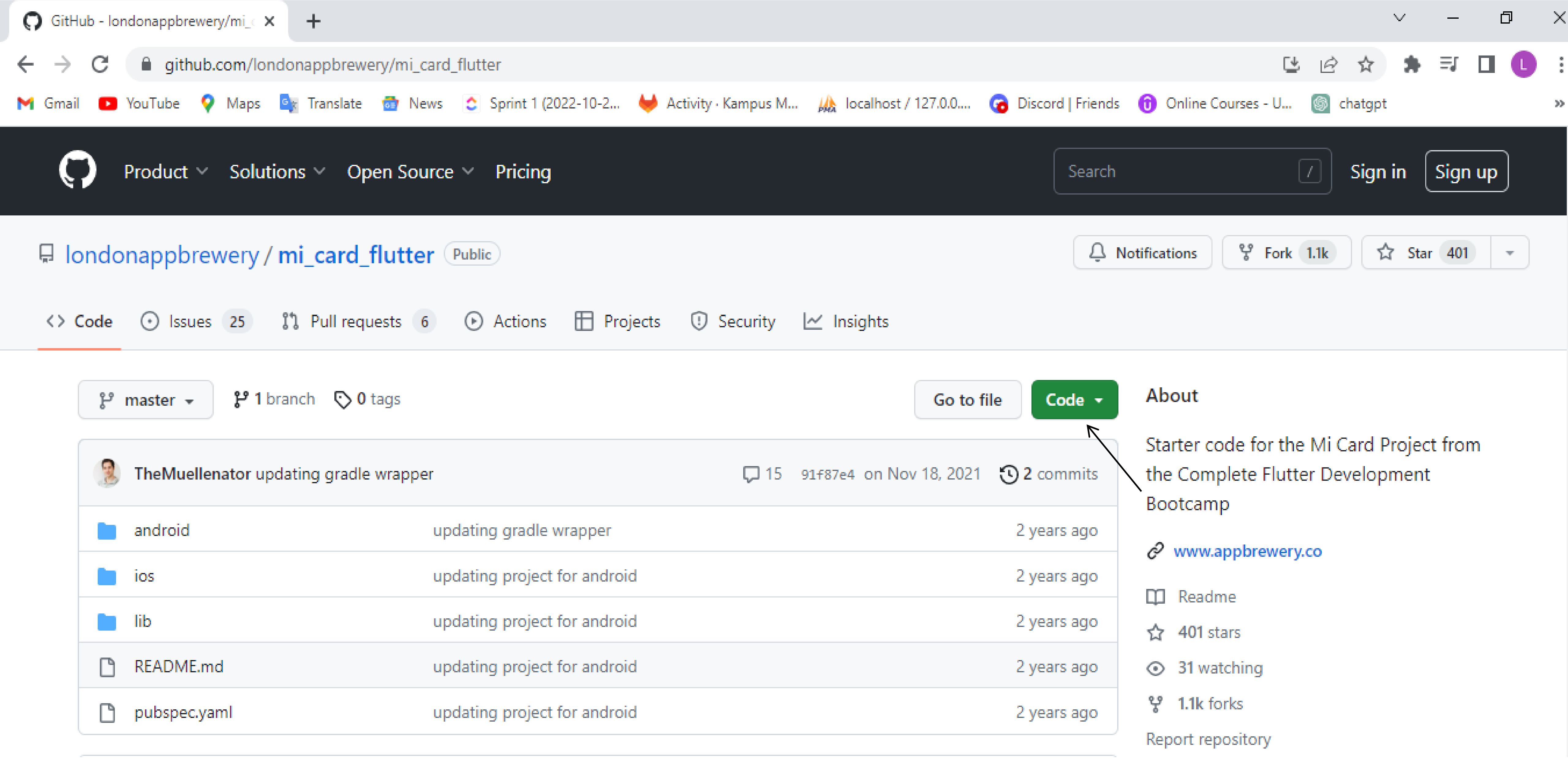
```
import 'package:flutter/material.dart';

void main() {
  runApp(
    MaterialApp(
      home: Scaffold(
        backgroundColor: Colors.blueGrey[900],
        appBar: AppBar(
          title: Text('My Apps'),
          backgroundColor: Colors.blue[900],
        ), // AppBar
        body: Center(
          child: Image(
            image: NetworkImage('https://docs.flutter.dev/assets/images/flu
          ), // Image
        ), // Center
      ), // Scaffold
    ), // MaterialApp
  );
}
```

MOJADIPRO

two

Mengclone Project Flutter dari Github

Mengclone Project Flutter dari Github



londonappbrewery / mi_card_flutter Public

TheMuellenator updating gradle wrapper 15 91f87e4 on Nov 18, 2021 2 commits

android	updating gradle wrapper	2 years ago
ios	updating project for android	2 years ago
lib	updating project for android	2 years ago
README.md	updating project for android	2 years ago
pubspec.yaml	updating project for android	2 years ago

About

Starter code for the Mi Card Project from the Complete Flutter Development Bootcamp

www.appbrewery.co

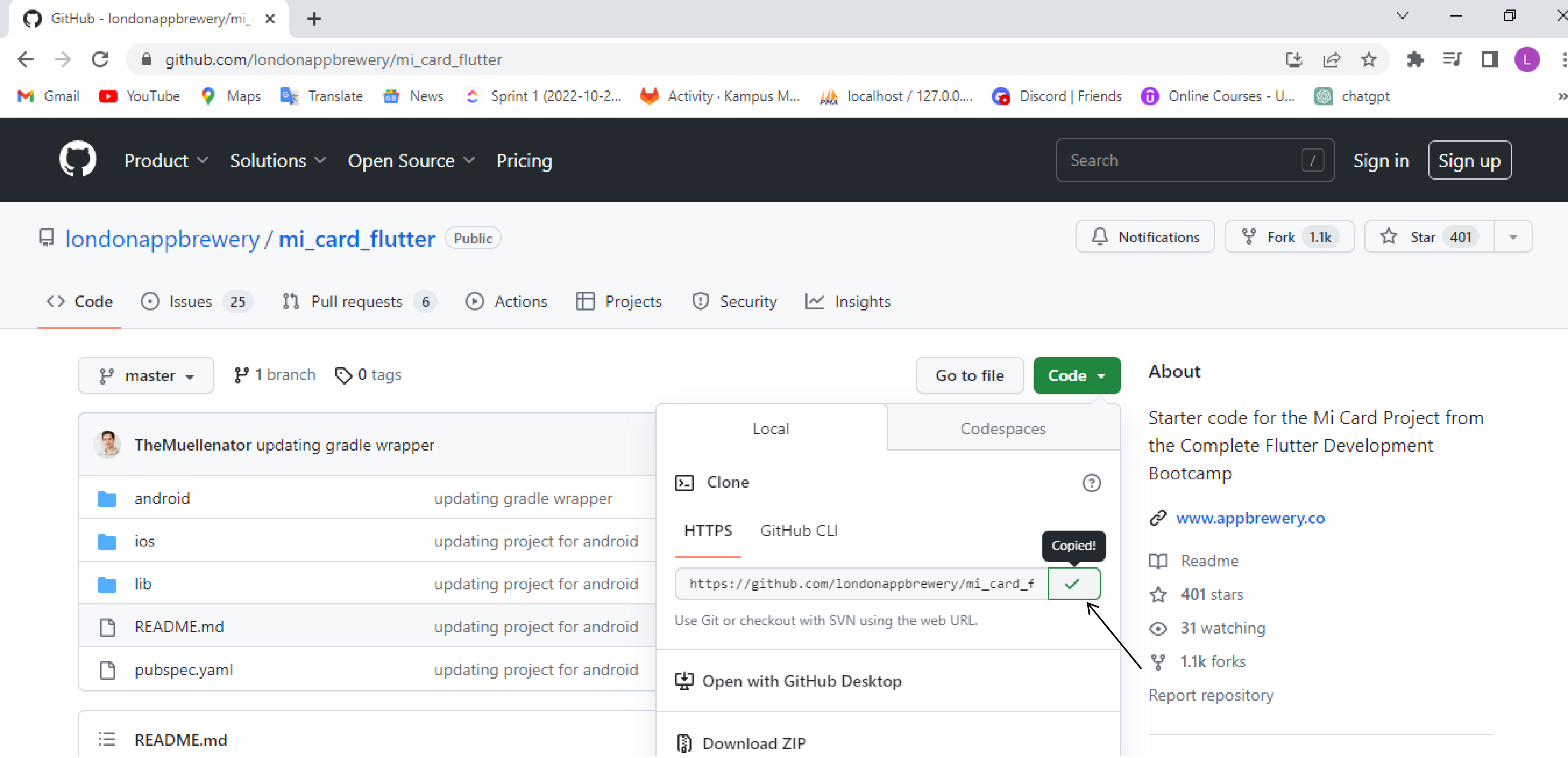
Readme

401 stars

31 watching

1.1k forks

Report repository



TheMuellenator updating gradle wrapper

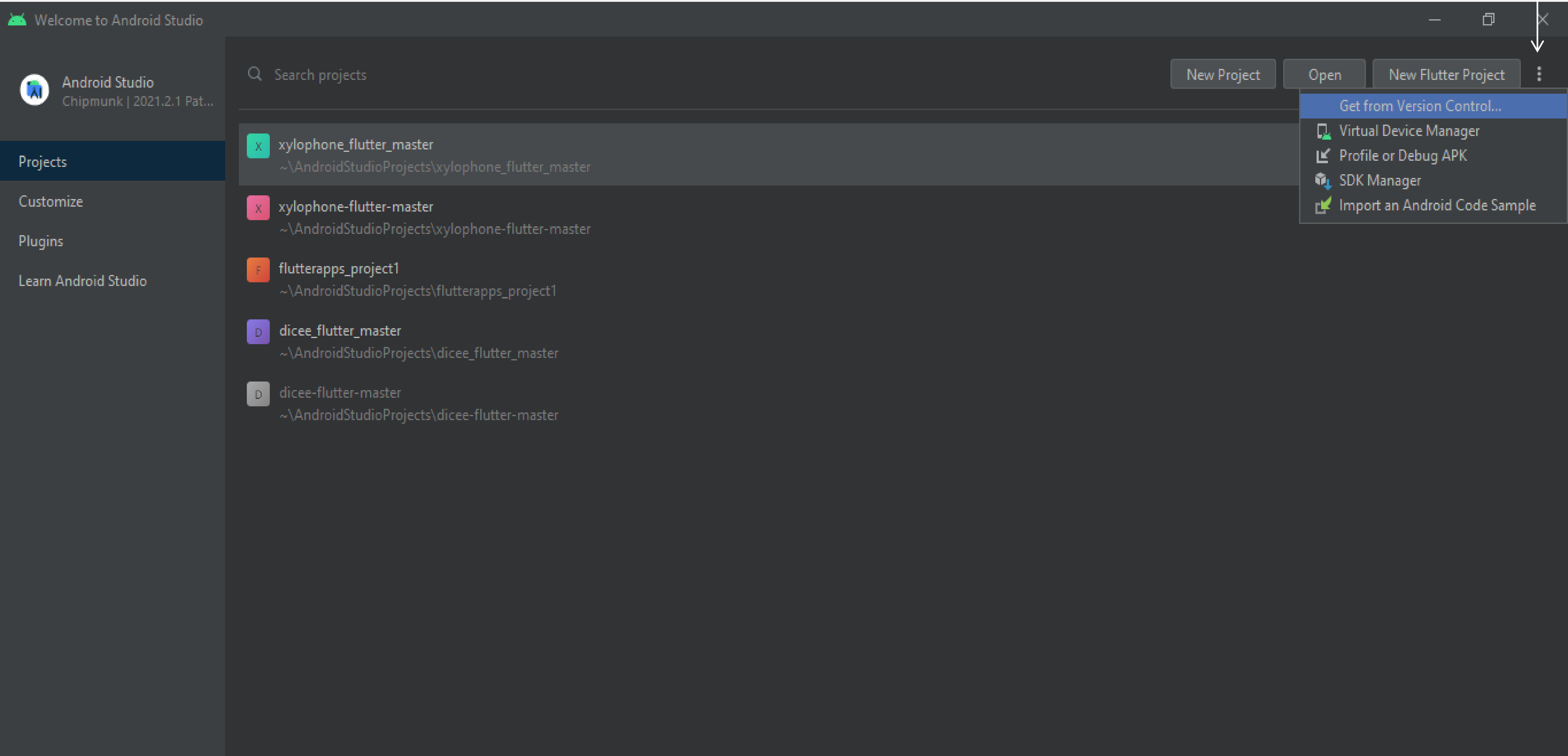
android	updating gradle wrapper
ios	updating project for android
lib	updating project for android
README.md	updating project for android
pubspec.yaml	updating project for android

README.md

```
https://github.com/londonappbrewery/mi_card_f
```

Use Git or checkout with SVN using the web URL.

Starter code for the Mi Card Project from the Complete Flutter Development Bootcamp



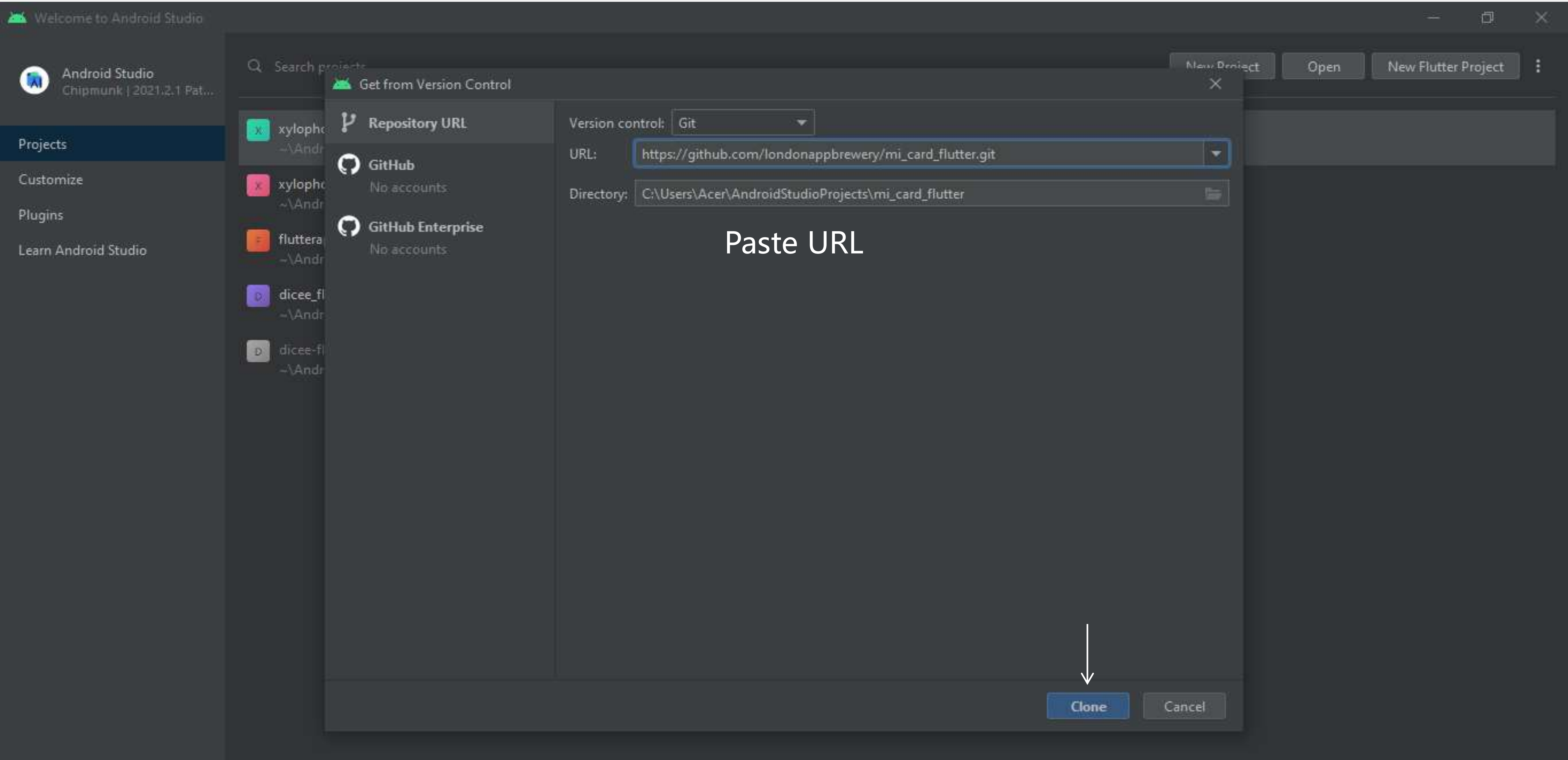
X xylophone_flutter_master
~\AndroidStudioProjects\xylophone_flutter_master

X xylophone-flutter-master
~\AndroidStudioProjects\xylophone-flutter-master

F flutterapps_project1
~\AndroidStudioProjects\flutterapps_project1

D dicee_flutter_master
~\AndroidStudioProjects\dicee_flutter_master

D dicee-flutter-master
~\AndroidStudioProjects\dicee-flutter-master

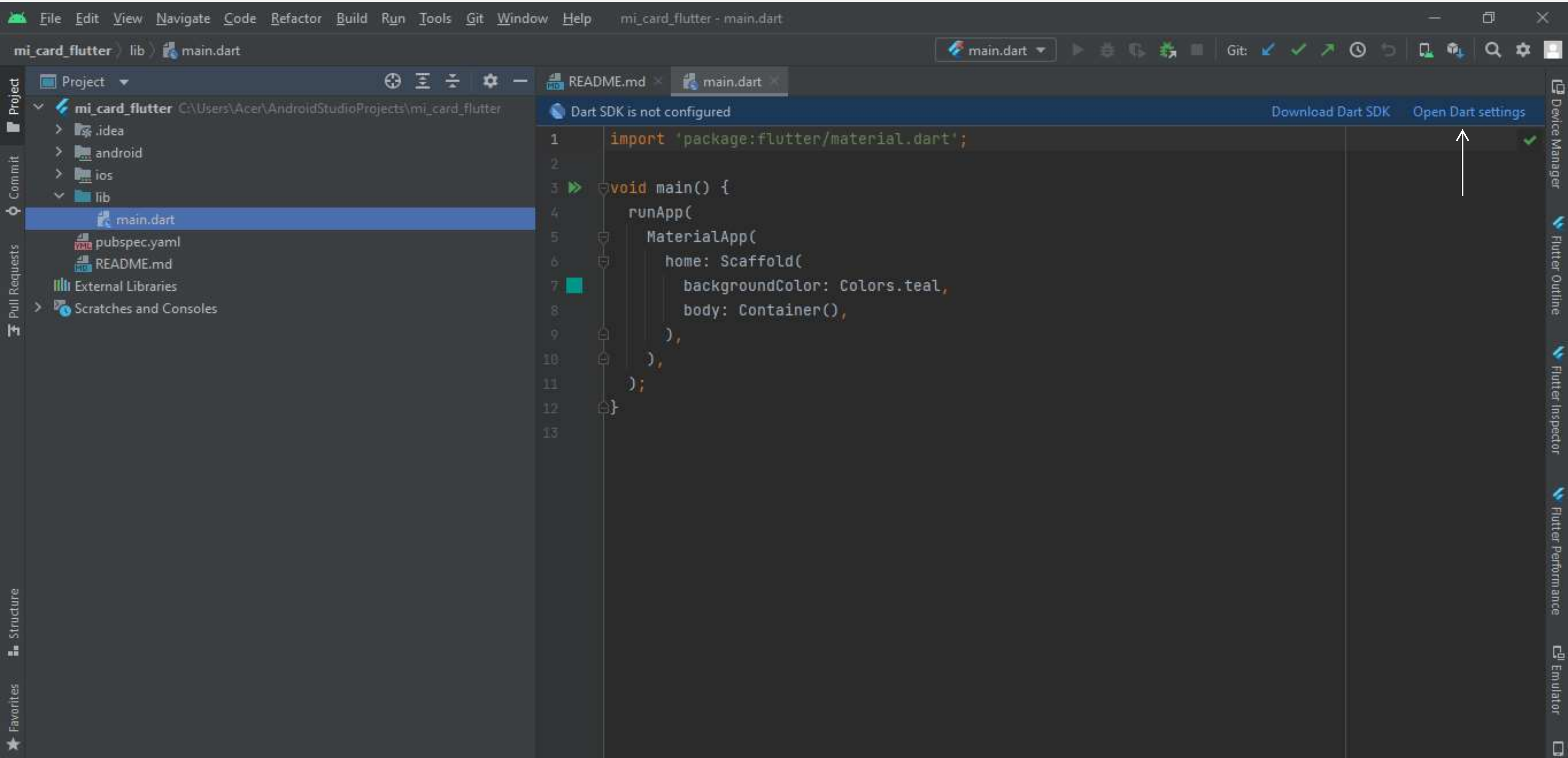


Version control: Git

URL: https://github.com/londonappbrewery/mi_card_flutter.git

Directory: C:\Users\Acer\AndroidStudioProjects\mi_card_flutter

Paste URL

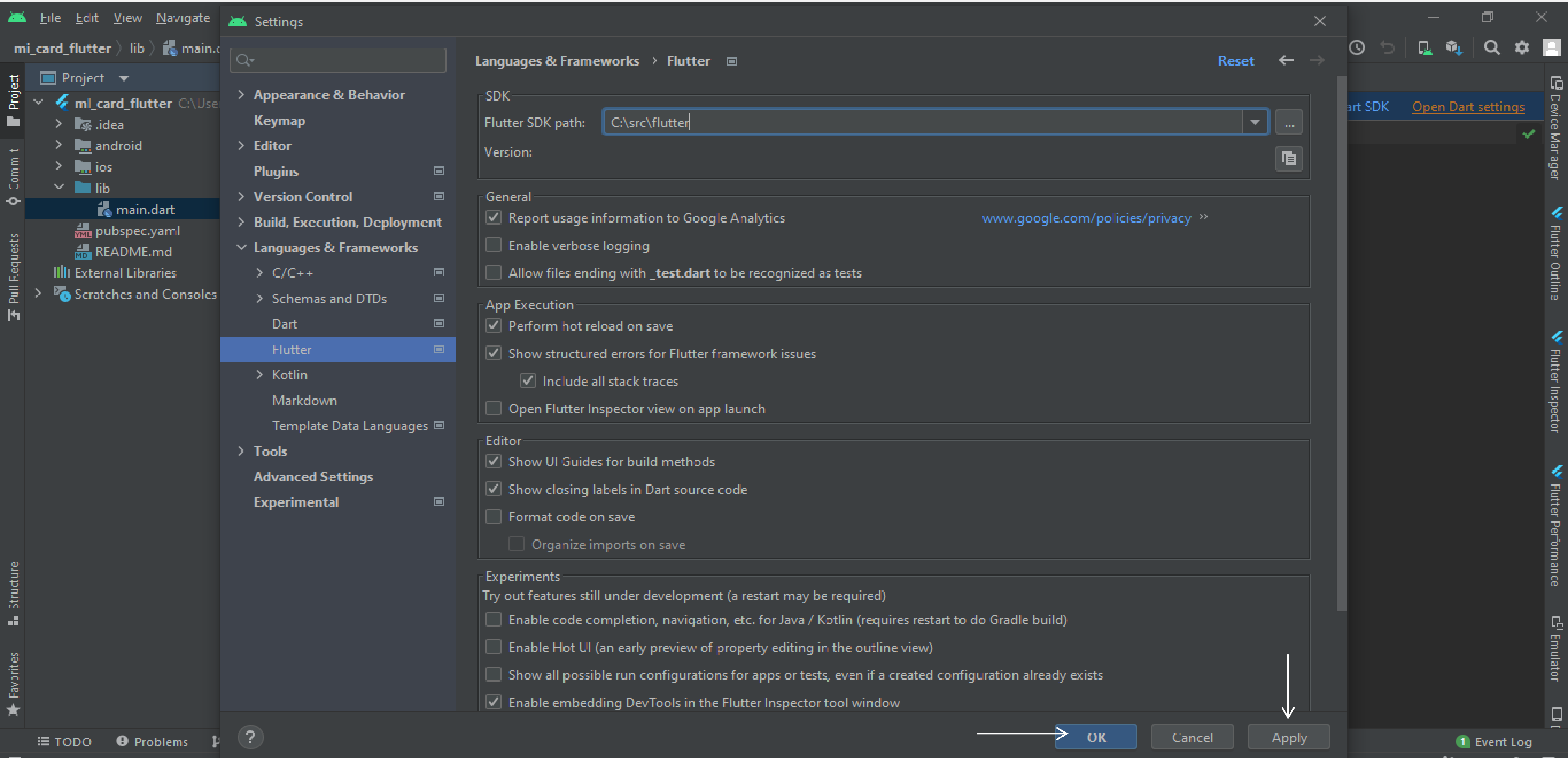


Dart SDK is not configured

Download Dart SDK

Open Dart settings

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

void main() {
  runApp(
    MaterialApp(
      home: Scaffold(
        backgroundColor: Colors.teal,
        body: Container(),
      ),
    ),
  );
}
```



Languages & Frameworks › Flutter

SDK

Flutter SDK path: C:\src\flutter

Version:

General

☑ Report usage information to Google Analytics www.google.com/policies/privacy »

☐ Enable verbose logging

☐ Allow files ending with _test.dart to be recognized as tests

App Execution

☑ Perform hot reload on save

☑ Show structured errors for Flutter framework issues

☑ Include all stack traces

☐ Open Flutter Inspector view on app launch

Editor

☑ Show UI Guides for build methods

☑ Show closing labels in Dart source code

☐ Format code on save

☐ Organize imports on save

Experiments

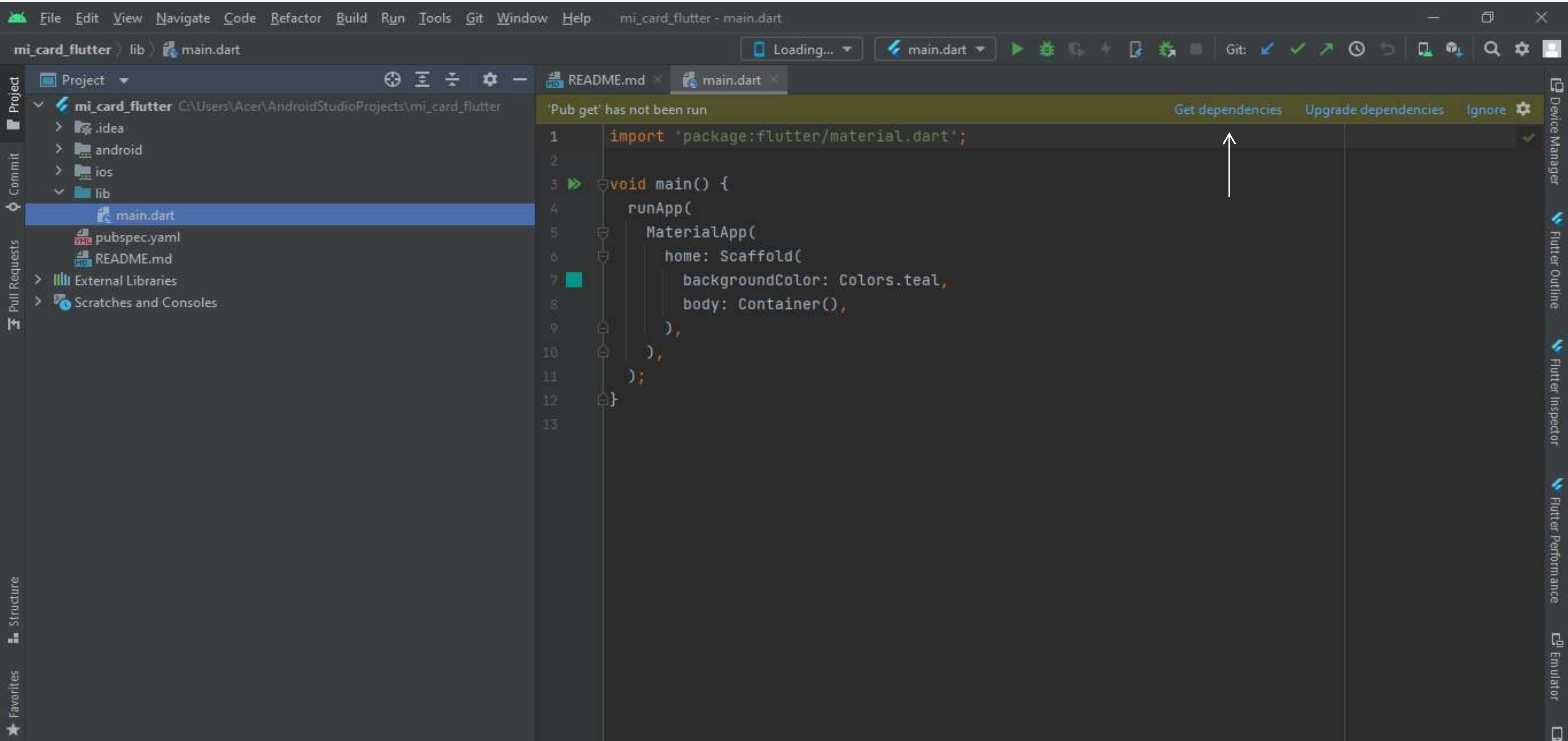
Try out features still under development (a restart may be required)

☐ Enable code completion, navigation, etc. for Java / Kotlin (requires restart to do Gradle build)

☐ Enable Hot UI (an early preview of property editing in the outline view)

☐ Show all possible run configurations for apps or tests, even if a created configuration already exists

☑ Enable embedding DevTools in the Flutter Inspector tool window



'Pub get' has not been run   Get dependencies   Upgrade dependencies   Ignore

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

void main() {
  runApp(
    MaterialApp(
      home: Scaffold(
        backgroundColor: Colors.teal,
        body: Container(),
      ),
    ),
  );
}

```

MOJADIPRO

three

Mempelajari cara menggunakan layout widgets

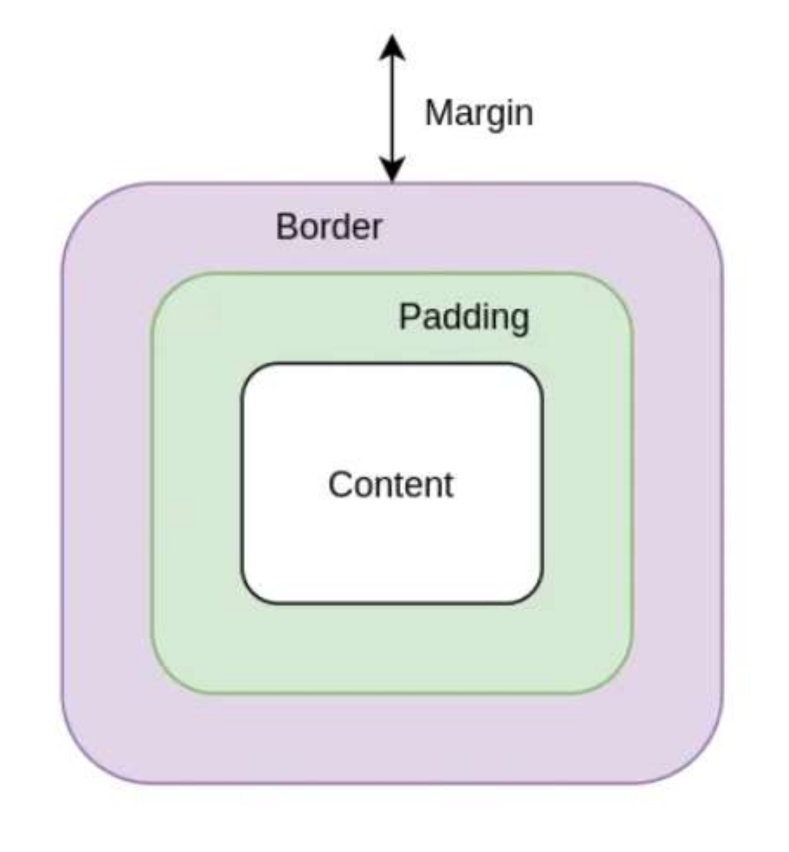
Container, Column, Row, dan Card

three

Container Widget

MOJADIPRO

Container adalah widget yang digunakan untuk mengatur tata letak dan penampilan suatu elemen di dalam tampilan Flutter. Widget ini memiliki banyak properti yang memungkinkan pengguna untuk mengubah ukuran, warna, gaya, dan penempatan elemen di dalamnya.



three

MOJADIPRO

Properti yang digunakan di Container

01

Margin digunakan untuk mengatur jarak antara widget dengan widget lain atau dengan tepi layar.

02

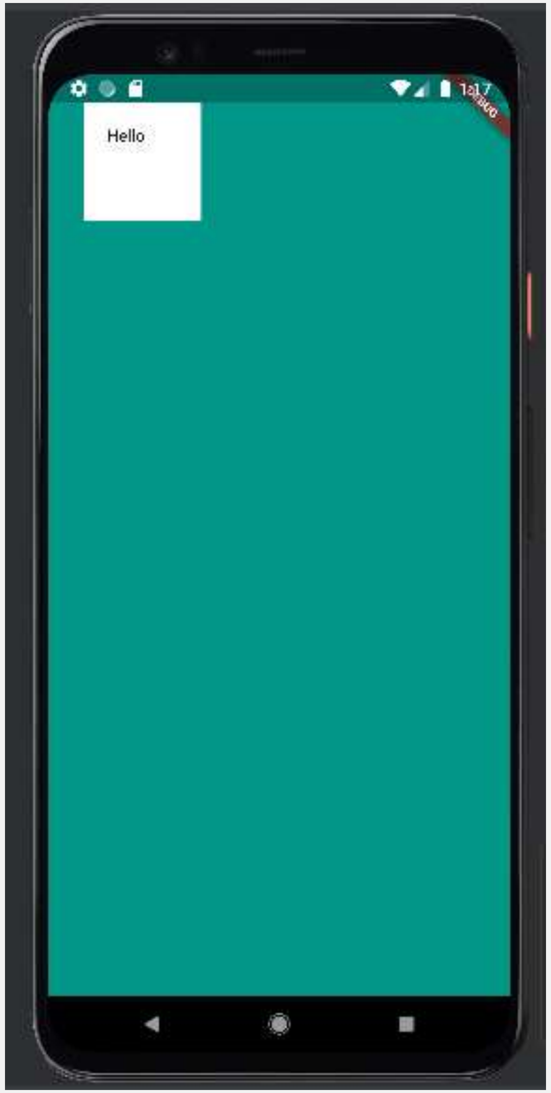
Padding digunakan untuk mengatur jarak antara isi konten atau isi widget dan batas widget tersebut.

03

Height (tinggi) merujuk pada dimensi vertikal dari suatu widget di dalam Flutter.

04

Width (lebar) merujuk pada dimensi horizontal dari suatu widget di dalam Flutter.



Hello

05

EdgeInsets adalah kelas dalam Flutter yang digunakan untuk mengatur jarak (padding) dari tepi widget.

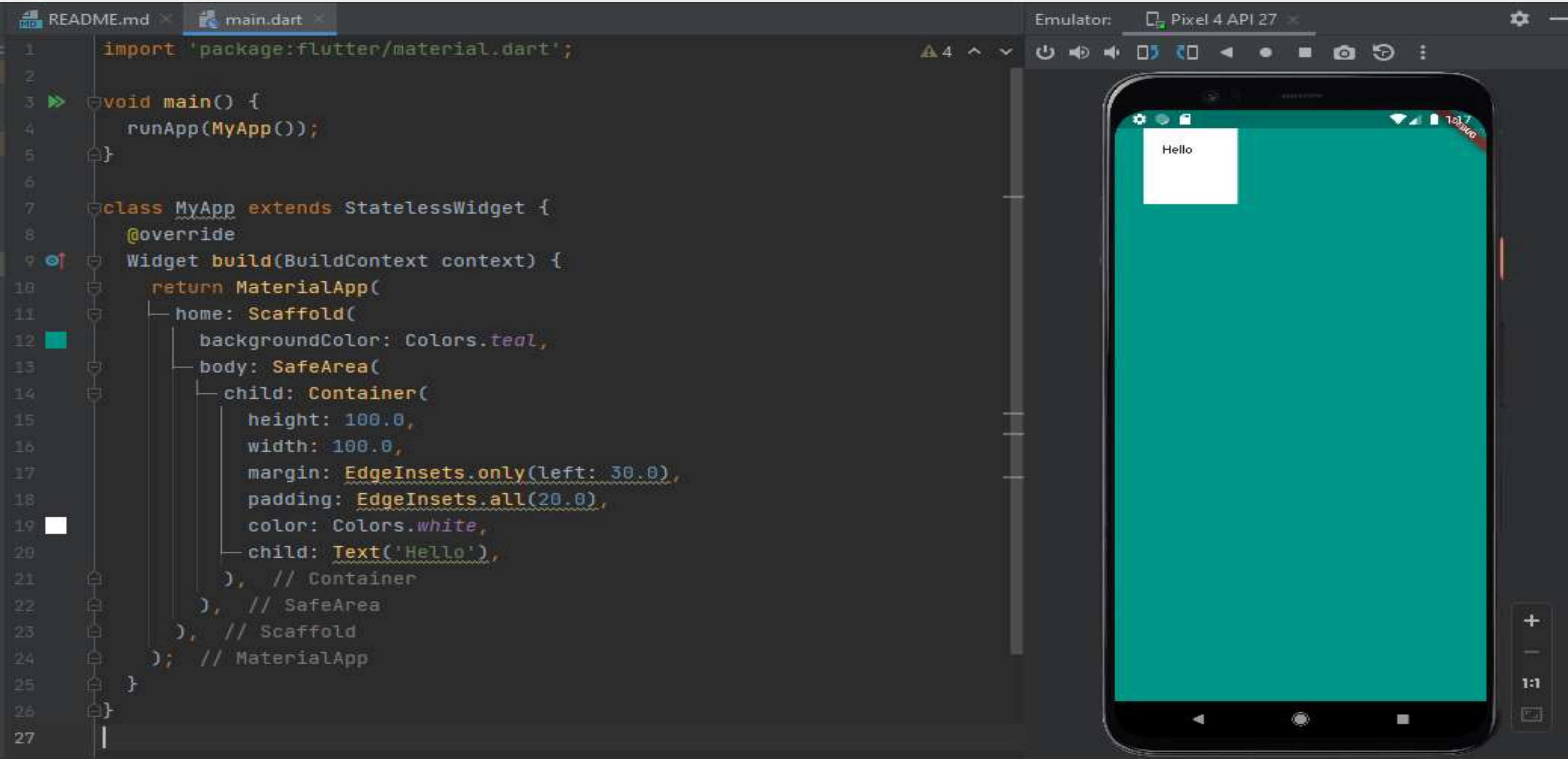
1. EdgeInsets.all(value) mengatur jarak dari semua sisi widget.

2. EdgeInsets.only({left, top, right, bottom}) mengatur jarak yang berbeda-beda dari setiap sisi widget.

3. EdgeInsets.symmetric({vertical, horizontal}) mengatur jarak secara simetris pada sisi vertikal dan horizontal widget.

4. EdgeInsets.fromLTRB(left, top, right, bottom) mengatur jarak yang berbeda-beda, dengan spesifikasi jarak pada setiap sisi yang ingin diubah.

Contoh Penggunaan Kode Pada Container



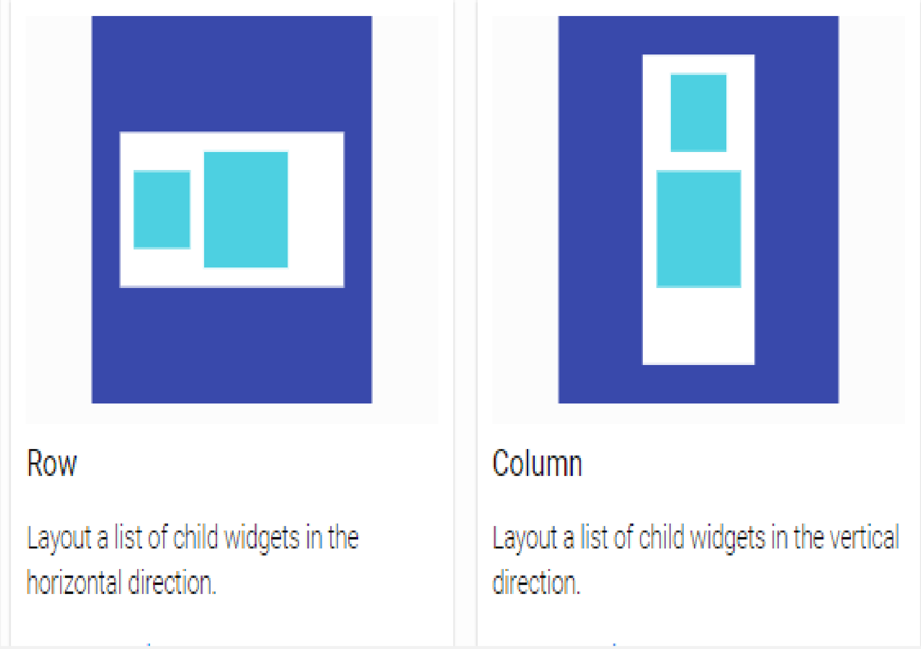
```dart
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        backgroundColor: Colors.teal,
        body: SafeArea(
          child: Container(
            height: 100.0,
            width: 100.0,
            margin: EdgeInsets.only(left: 30.0),
            padding: EdgeInsets.all(20.0),
            color: Colors.white,
            child: Text('Hello'),
          ), // Container
        ), // SafeArea
      ), // Scaffold
    ); // MaterialApp
  }
}
```

Column dan Row Widget

Multi-child layout widgets adalah widget yang memungkinkan penempatan lebih dari satu widget di dalamnya. Widget-widget ini memungkinkan Anda untuk mengatur tata letak dan tampilan dari beberapa widget secara bersamaan.

01 Column (Kolom) adalah widget yang mengatur elemen-elemen secara vertikal, dari atas ke bawah. Anda dapat menempatkan berbagai widget di dalam Column secara bertumpuk.

02 Row (Baris) adalah widget yang mengatur elemen-elemen secara horizontal, dari kiri ke kanan. Anda dapat menempatkan berbagai widget di dalam Row secara berdampingan.



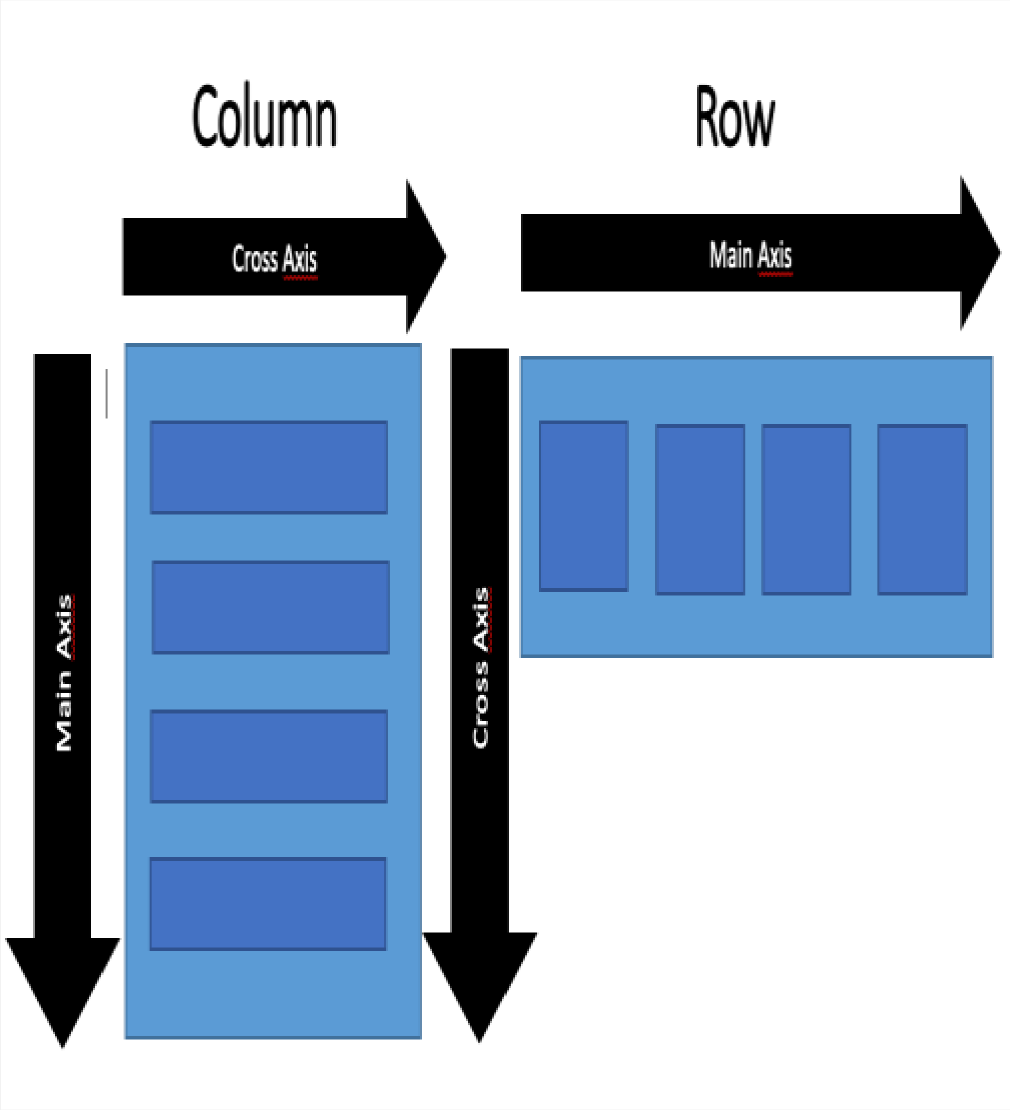
Row

Layout a list of child widgets in the horizontal direction.

Column

Layout a list of child widgets in the vertical direction.

Properti yang digunakan di Column dan Row



children: Properti ini merupakan wajib dan berfungsi untuk menentukan daftar <widget>[], yang akan ditempatkan.

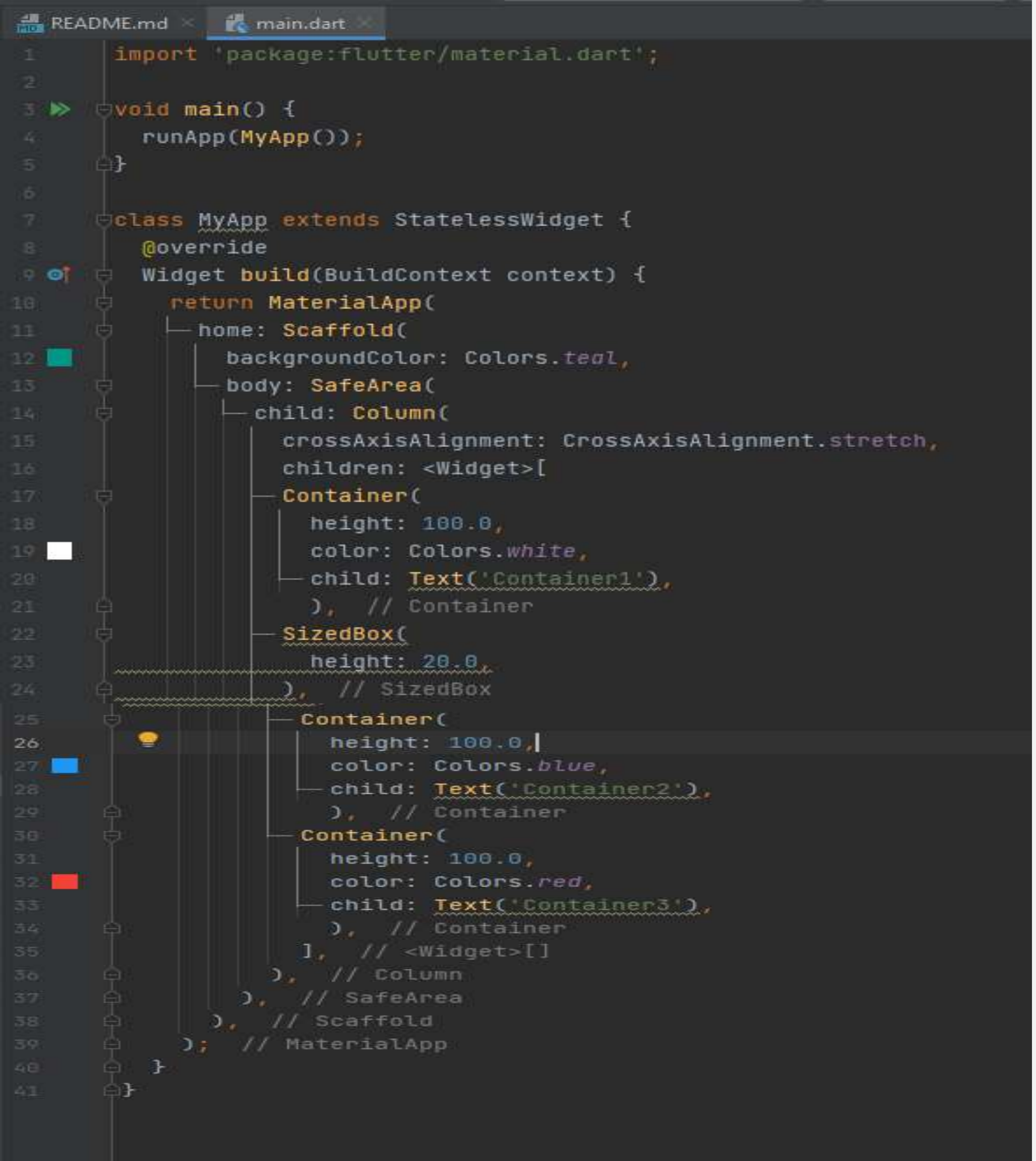
mainAxisAlignment: Properti ini digunakan untuk mengatur tata letak vertical pada kolom, dan horizontal pada baris.

crossAxisAlignment: Properti ini digunakan untuk mengatur tata letak horizontal pada kolom, dan vertical pada baris.

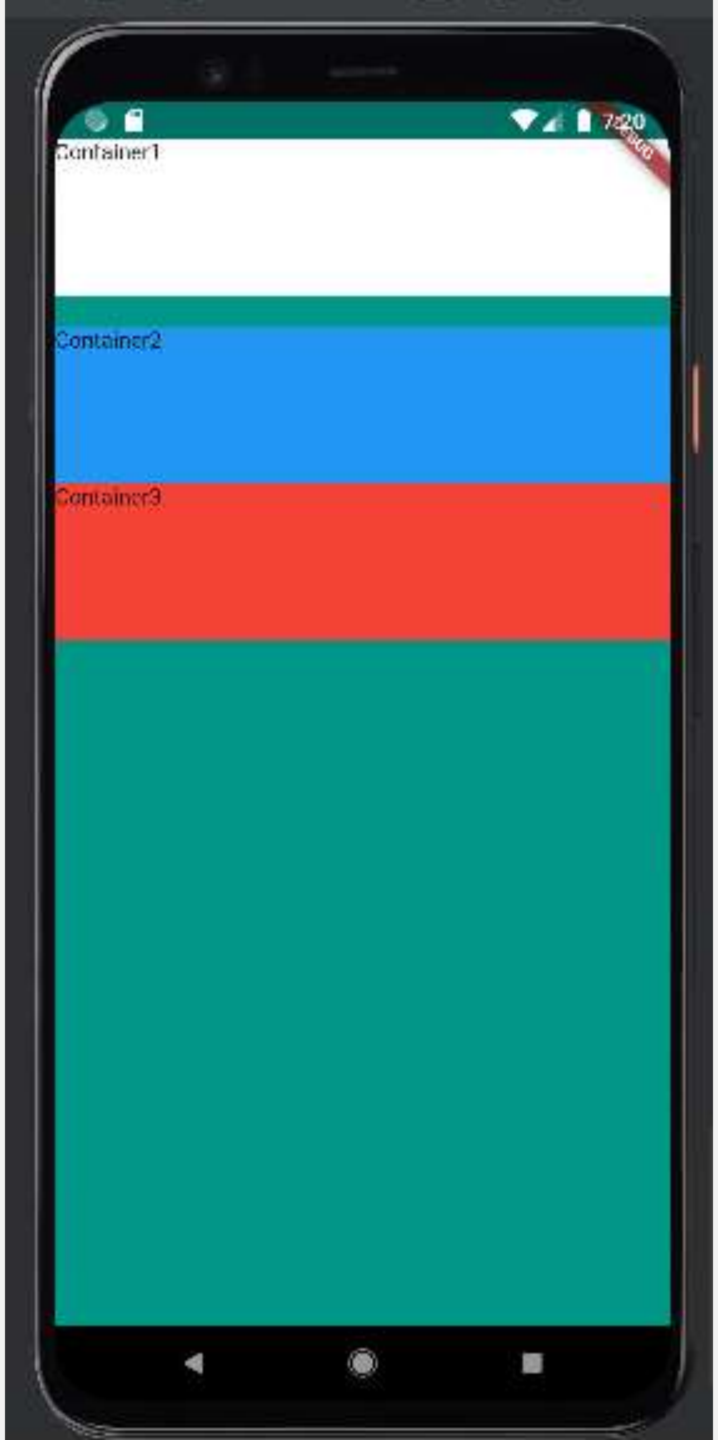
mainAxisSize: Properti ini digunakan untuk mengatur ukuran utama. Jika sebagai MainAxisSize.max, maka akan memperluas ukuran. Jika sebagai MainAxisSize.min, maka akan mengambil ukuran minimum.

MOJADIPRO

Contoh Penggunaan Kode Pada Column



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

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        backgroundColor: Colors.teal,
        body: SafeArea(
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: <Widget>[
            Container(
              height: 100.0,
              color: Colors.white,
              child: Text('Container1'),
              ),  // Container
            SizedBox(
              height: 20.0,
            ),  // SizedBox
             Container(
               height: 100.0,
               color: Colors.blue,
               child: Text('Container2'),
               ),  // Container
             Container(
               height: 100.0,
               color: Colors.red,
               child: Text('Container3'),
               ),  // Container
              ],  // <Widget>[]
            ),  // Column
          ),  // SafeArea
        ),  // Scaffold
      );  // MaterialApp
    }
}
```

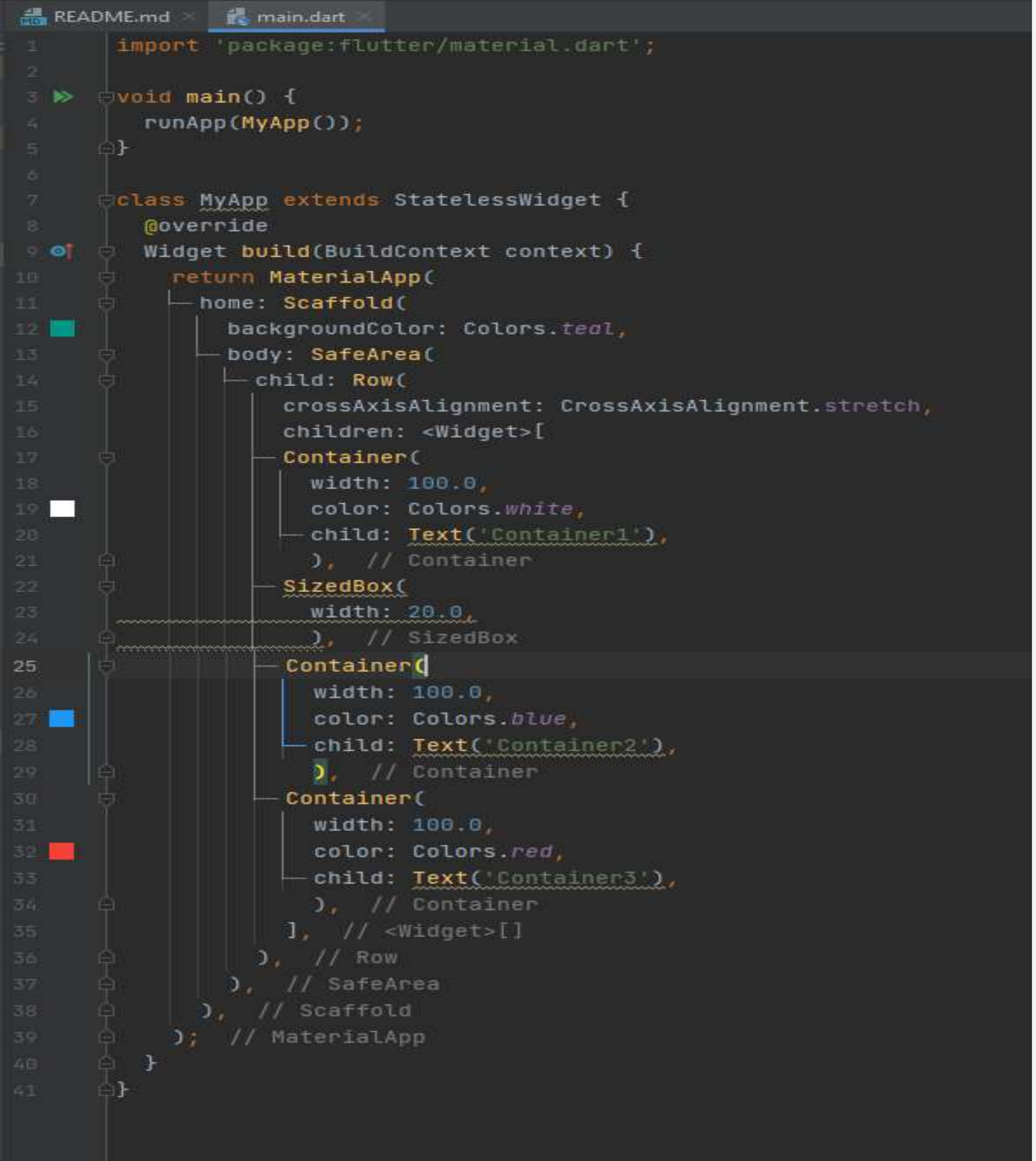


Container1

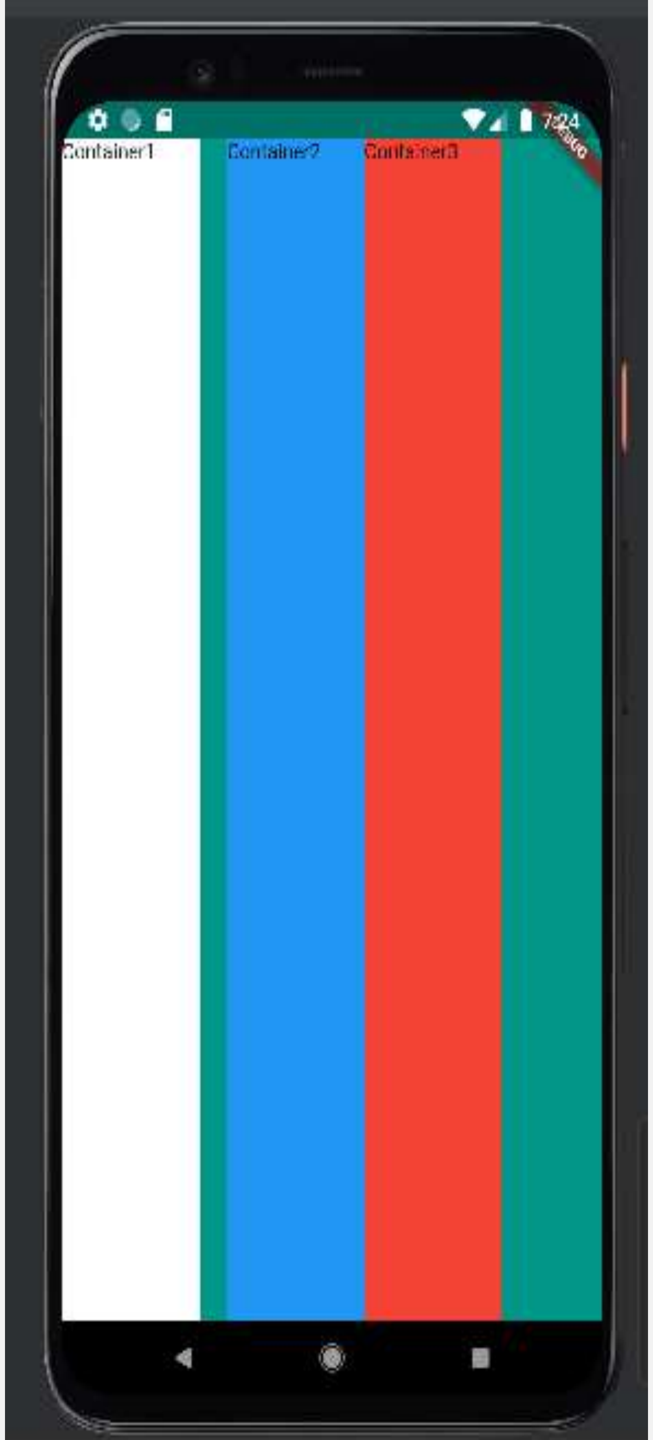
Container2

Container3

Contoh Penggunaan Kode Pada Row



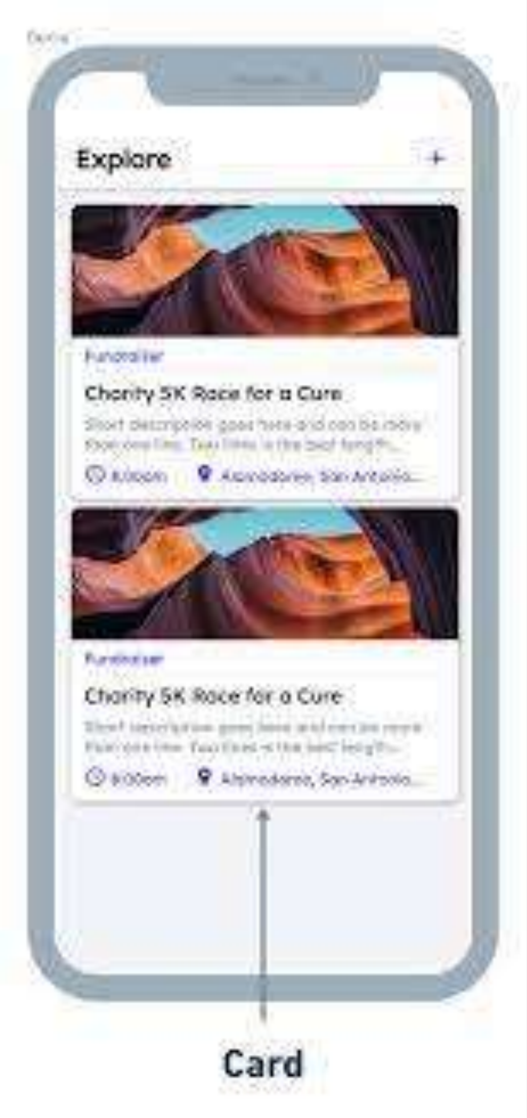
```dart
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        backgroundColor: Colors.teal,
        body: SafeArea(
          child: Row(
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: <Widget>[
            Container(
              width: 100.0,
              color: Colors.white,
              child: Text('Container1'),
              ), // Container
            SizedBox(
              width: 20.0,
              ), // SizedBox
             Container(
              width: 100.0,
              color: Colors.blue,
              child: Text('Container2'),
              ), // Container
             Container(
              width: 100.0,
              color: Colors.red,
              child: Text('Container3'),
              ), // Container
             ], // <Widget>[]
          ), // Row
        ), // SafeArea
      ), // Scaffold
    ); // MaterialApp
  }
}
```



Card Widget

Card merujuk pada widget yang digunakan untuk menampilkan konten dalam bentuk kartu dengan latar belakang dan bayangan.

Widget Card biasanya digunakan untuk mengatur konten dalam tata letak. Secara sederhana, sebuah card terdiri dari beberapa elemen, seperti latar belakang, judul, gambar, teks, dan sebagainya.



Fundraiser

Charity 5K Race for a Cure

Short description goes here and can be more than one line. Two lines is the best length...

8:00am Alamodome, San Antonio...

Fundraiser

Charity 5K Race for a Cure

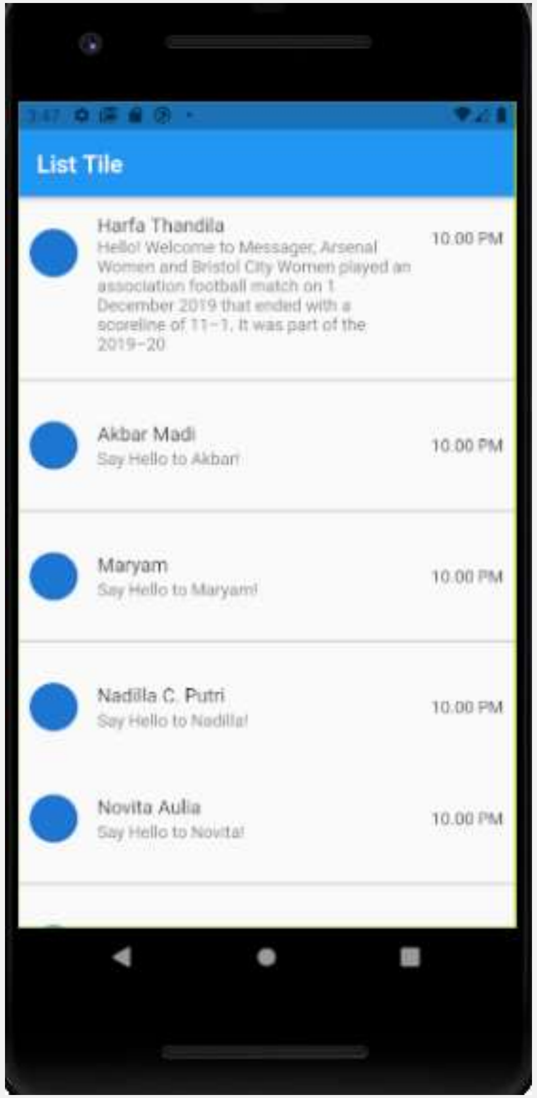
Short description goes here and can be more than one line. Two lines is the best length...

8:00am Alamodome, San Antonio...

three

ListTile Widget

List Tile adalah sebuah widget yang digunakan untuk menampilkan data dalam bentuk daftar (list) dengan tampilan yang sudah dirancang secara default. Widget ini digunakan untuk membuat daftar item yang terstruktur dengan mudah.



List Tile

Harfa Thandila
Hello! Welcome to Messager, Arsenal Women and Bristol City Women played an association football match on 1 December 2019 that ended with a scoreline of 11–1. It was part of the 2019–20
10.00 PM

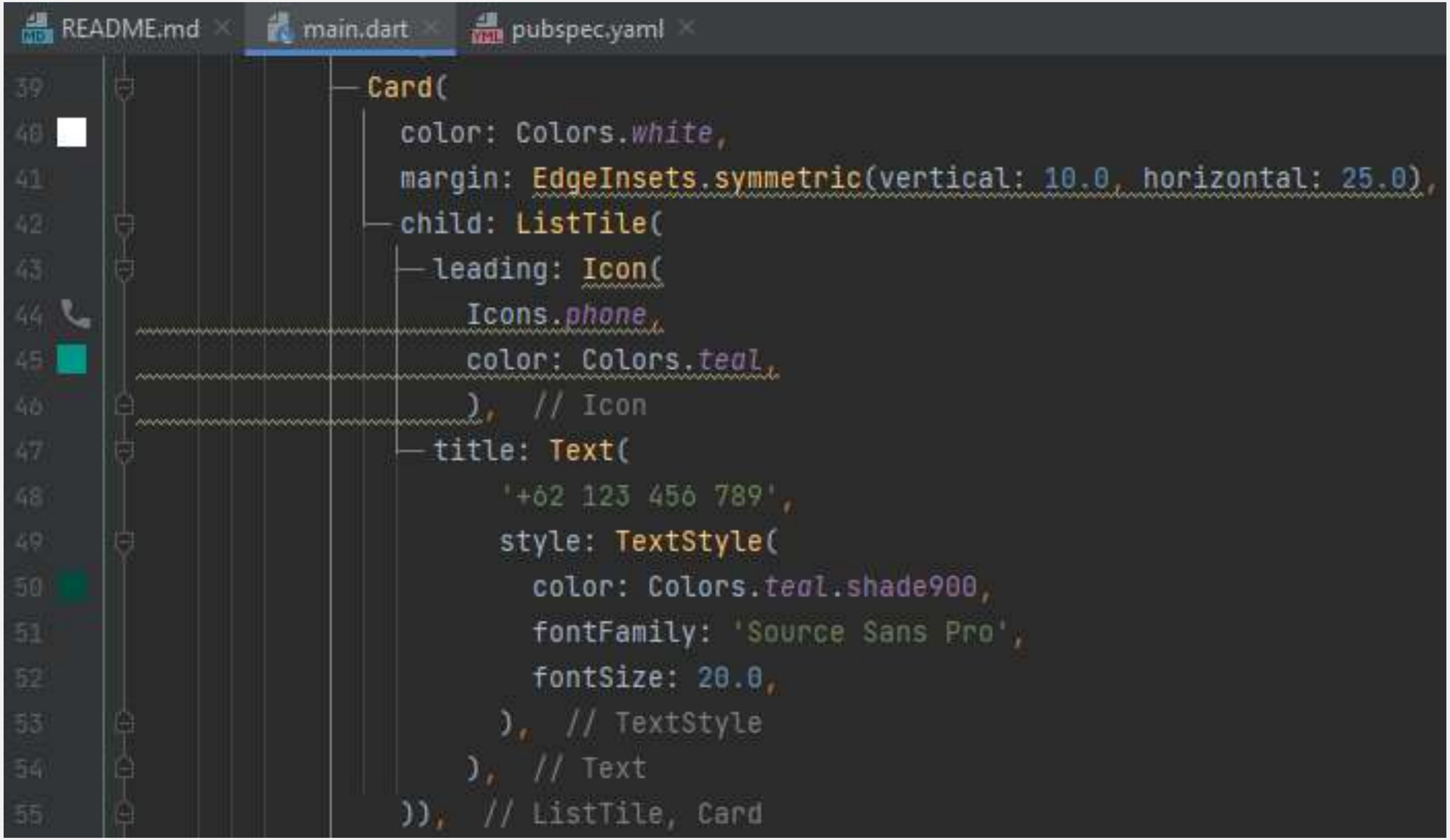
Akbar Madi
Say Hello to Akbar!
10.00 PM

Maryam
Say Hello to Maryam!
10.00 PM

Nadilla C. Putri
Say Hello to Nadilla!
10.00 PM

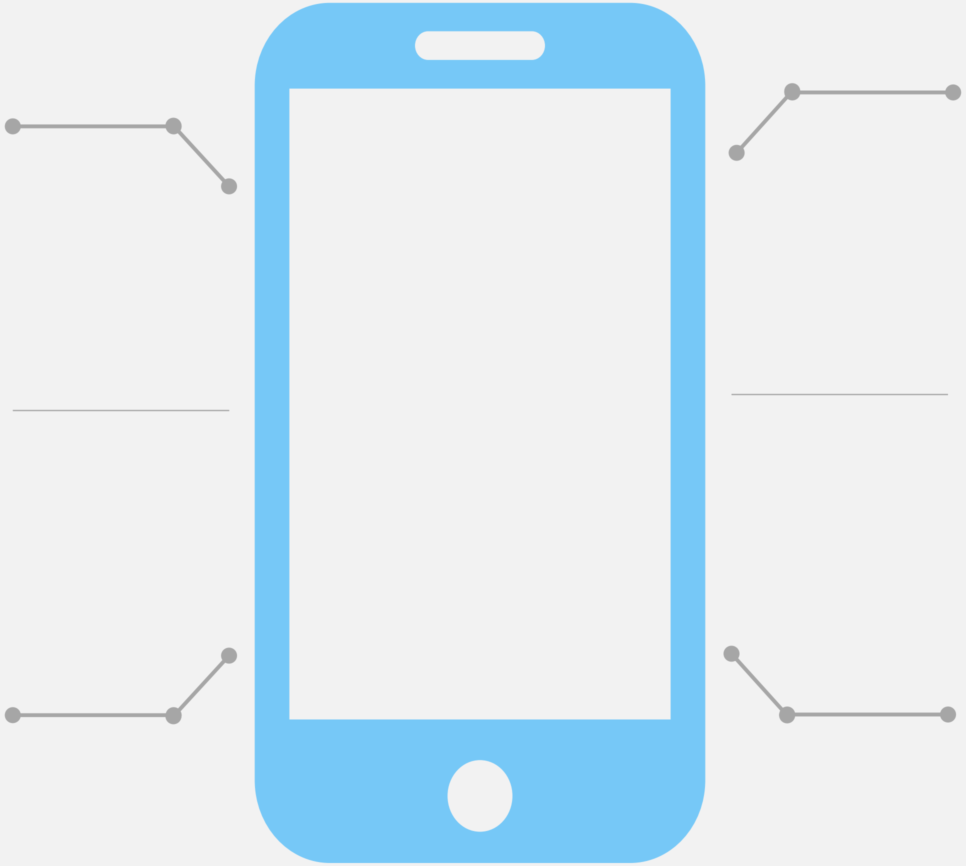
Novita Aulia
Say Hello to Novita!
10.00 PM

Contoh Penggunaan Kode Pada Card & ListTile



```
Card(
  color: Colors.white,
  margin: EdgeInsets.symmetric(vertical: 10.0, horizontal: 25.0),
  child: ListTile(
    leading: Icon(
      Icons.phone,
      color: Colors.teal,
      ), // Icon
    title: Text(
        '+62 123 456 789',
        style: TextStyle(
          color: Colors.teal.shade900,
          fontFamily: 'Source Sans Pro',
          fontSize: 20.0,
        ), // TextStyle
      ), // Text
  )), // ListTile, Card
```

Properti yang digunakan di Card



Child: untuk menentukan widget yang akan menjadi konten utama dari card.

Color: untuk menentukan latar belakang warna card. Dapat berupa objek Color atau ColorScheme.

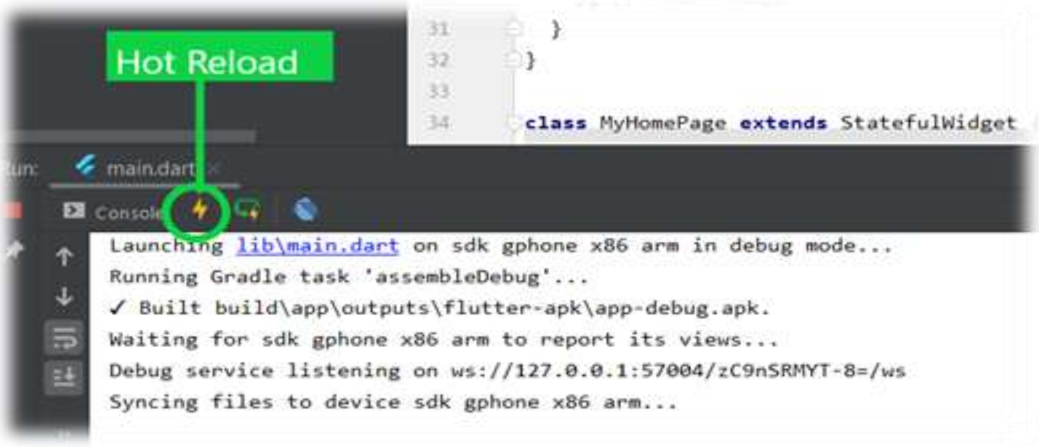
Elevation: untuk menentukan tingkat bayangan atau "elevation" card.

Shape: untuk menentukan bentuk card. Dapat berupa objek ShapeBorder, seperti RoundedRectangleBorder untuk card dengan sudut melengkung, atau BeveledRectangleBorder untuk card dengan sudut miring.

borderOnForeground: untuk menentukan apakah tepi card akan ditempatkan di atas atau di bawah konten card.

Margin: untuk mengatur margin card. Nilai properti ini dapat berupa objek EdgeInsets.

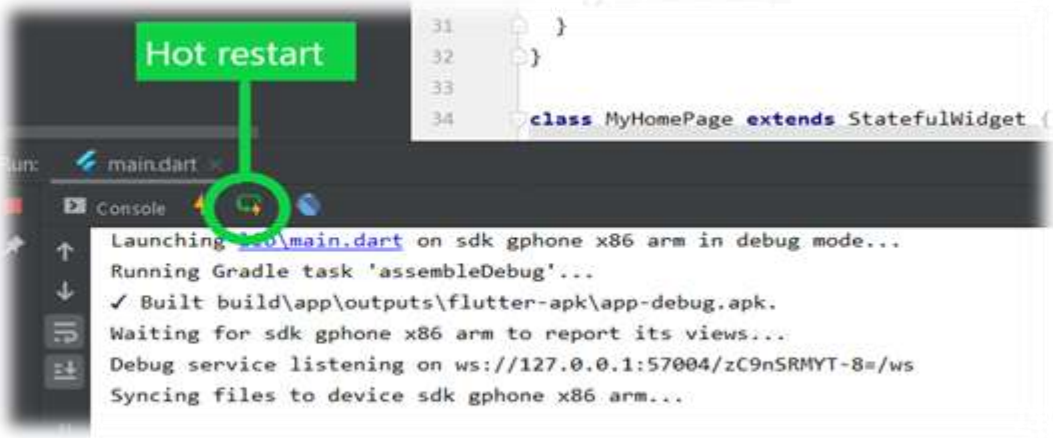
Perbedaan Hot Reload dan Hot Restart



```
    }
}

class MyHomePage extends StatefulWidget
```

```
Launching lib\main.dart on sdk gphone x86 arm in debug mode...
Running Gradle task 'assembleDebug'...
✓ Built build\app\outputs\flutter-apk\app-debug.apk.
Waiting for sdk gphone x86 arm to report its views...
Debug service listening on ws://127.0.0.1:57004/zC9nSRMYT-8=/ws
Syncing files to device sdk gphone x86 arm...
```

Hot reload memungkinkan pengembang untuk melihat perubahan yang dilakukan pada kode secara langsung pada aplikasi yang sedang berjalan.



```
    }
}

class MyHomePage extends StatefulWidget {
```

```
Launching lib\main.dart on sdk gphone x86 arm in debug mode...
Running Gradle task 'assembleDebug'...
✓ Built build\app\outputs\flutter-apk\app-debug.apk.
Waiting for sdk gphone x86 arm to report its views...
Debug service listening on ws://127.0.0.1:57004/zC9nSRMYT-8=/ws
Syncing files to device sdk gphone x86 arm...
```

Hot Restart akan memulai kembali aplikasi dari awal dengan menonaktifkan semua status yang ada.
Ini berarti bahwa semua keadaan aplikasi akan hilang.

four

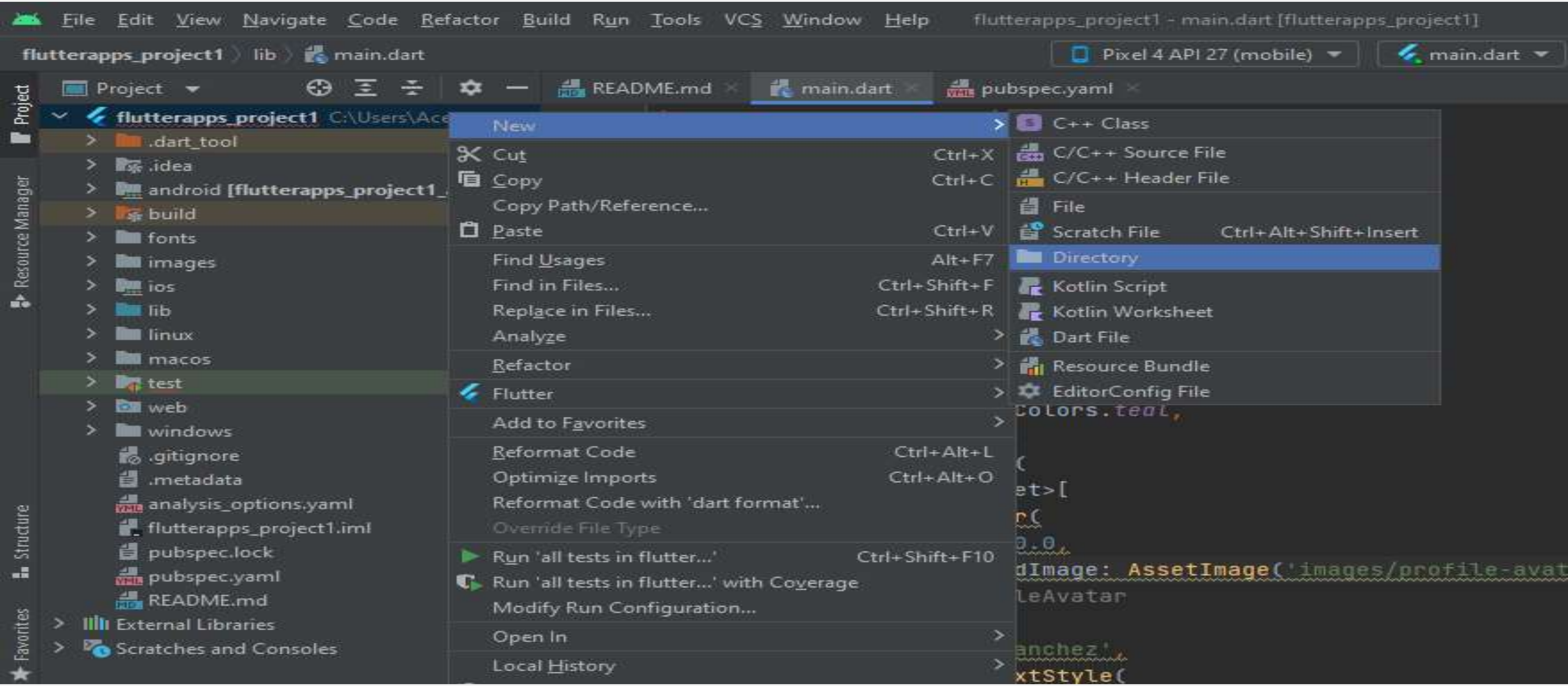
Mempelajari cara menggunakan file pubspec.yaml

Memasukkan Asset Image dan Fonts

four

Memasukkan Asset Image

MOJADIPRO

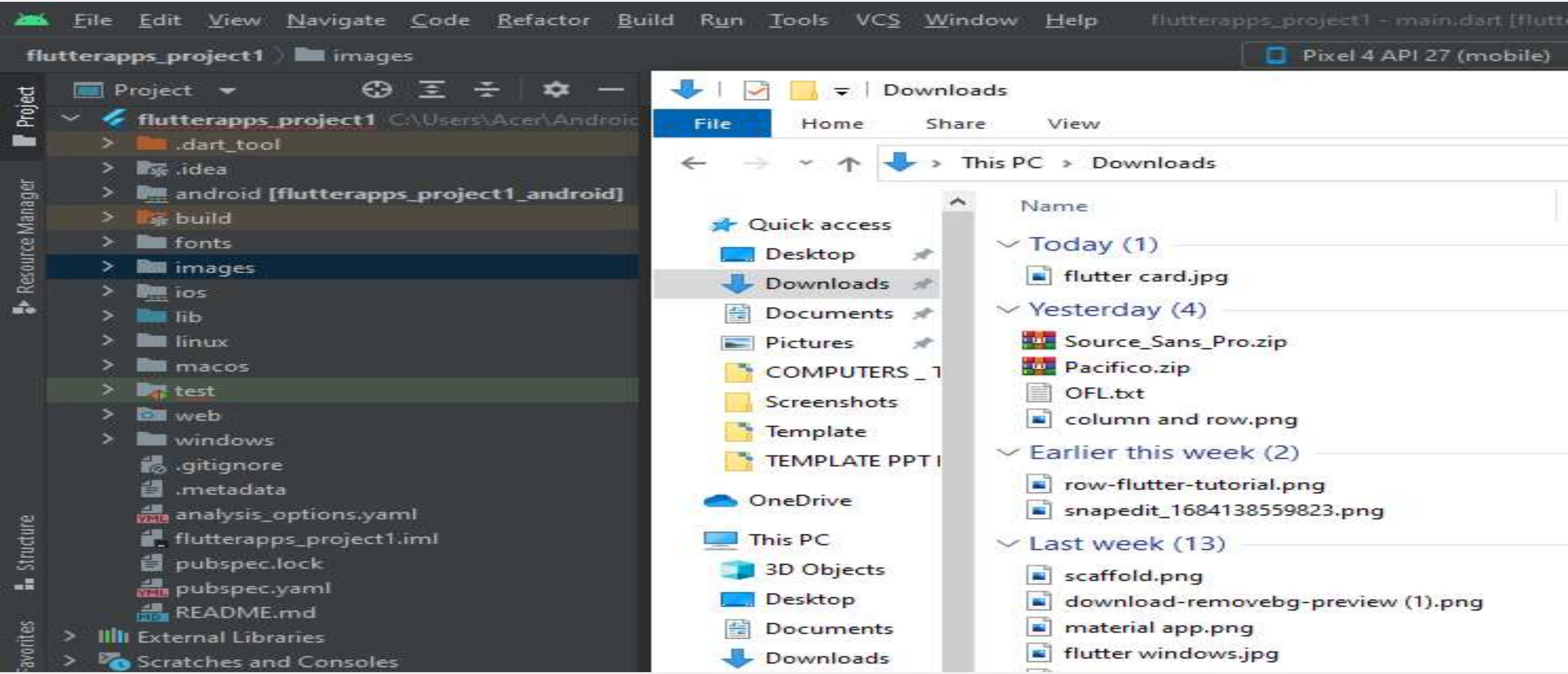


```
colors.teal,
(
et>[
r(
0.0,
dImage: AssetImage('images/profile-avat
leAvatar
anchez',
xtStyle(
```

Klik kanan pada flutter project → New → Directory → Beri nama images

four

MOJADIPRO



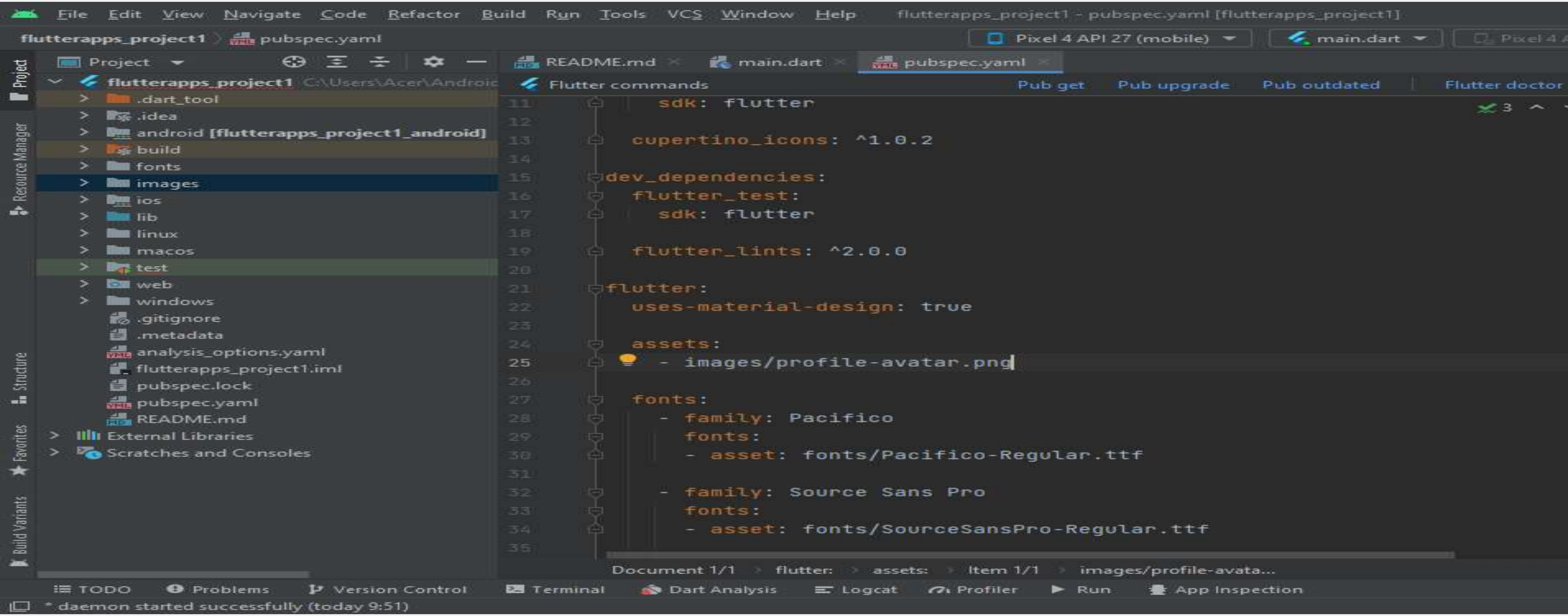
Pilih foto pada document → lalu tarik foto kedalam folder images

four

Memasukkan Asset Image Pada File Pubspec.yaml

MOJADIPRO

Flutter menggunakan file "pubspec.yaml" untuk mengelola dependensi dan asset pada aplikasi.



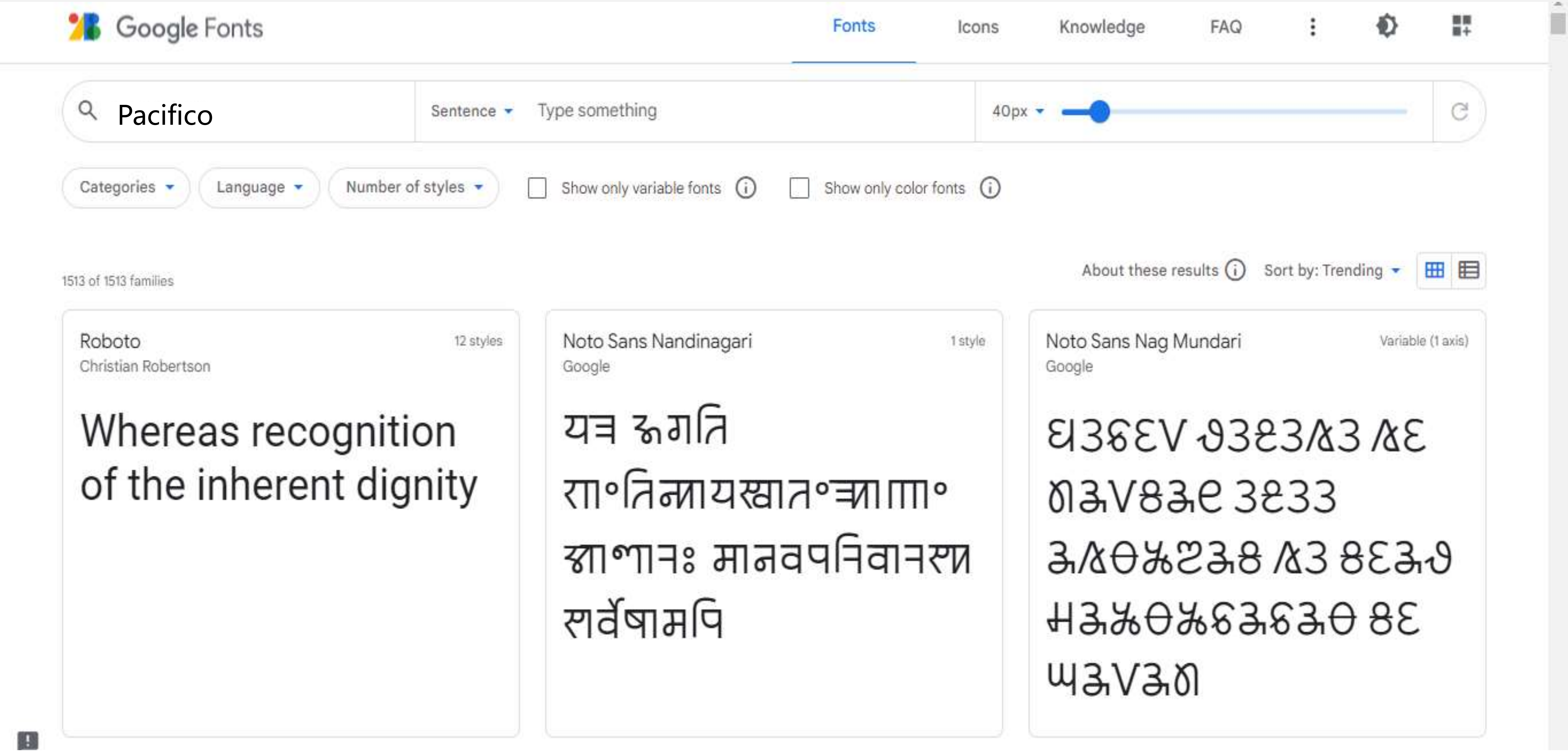
```yaml
    sdk: flutter

  cupertino_icons: ^1.0.2

dev_dependencies:
  flutter_test:
    sdk: flutter

  flutter_lints: ^2.0.0

flutter:
  uses-material-design: true

  assets:
    - images/profile-avatar.png

  fonts:
    - family: Pacifico
      fonts:
        - asset: fonts/Pacifico-Regular.ttf

    - family: Source Sans Pro
      fonts:
        - asset: fonts/SourceSansPro-Regular.ttf
```

```
* daemon started successfully (today 9:51)
```

Ketik nama foto yang ada di folder images pada assets → save → klik pub get → run emulator

four

Memasukkan Fonts Pada Project Flutter



Pacifico

Sentence

Type something

40px

□ Show only variable fonts

□ Show only color fonts

1513 of 1513 families

Sort by: Trending

Roboto — 12 styles
Christian Robertson

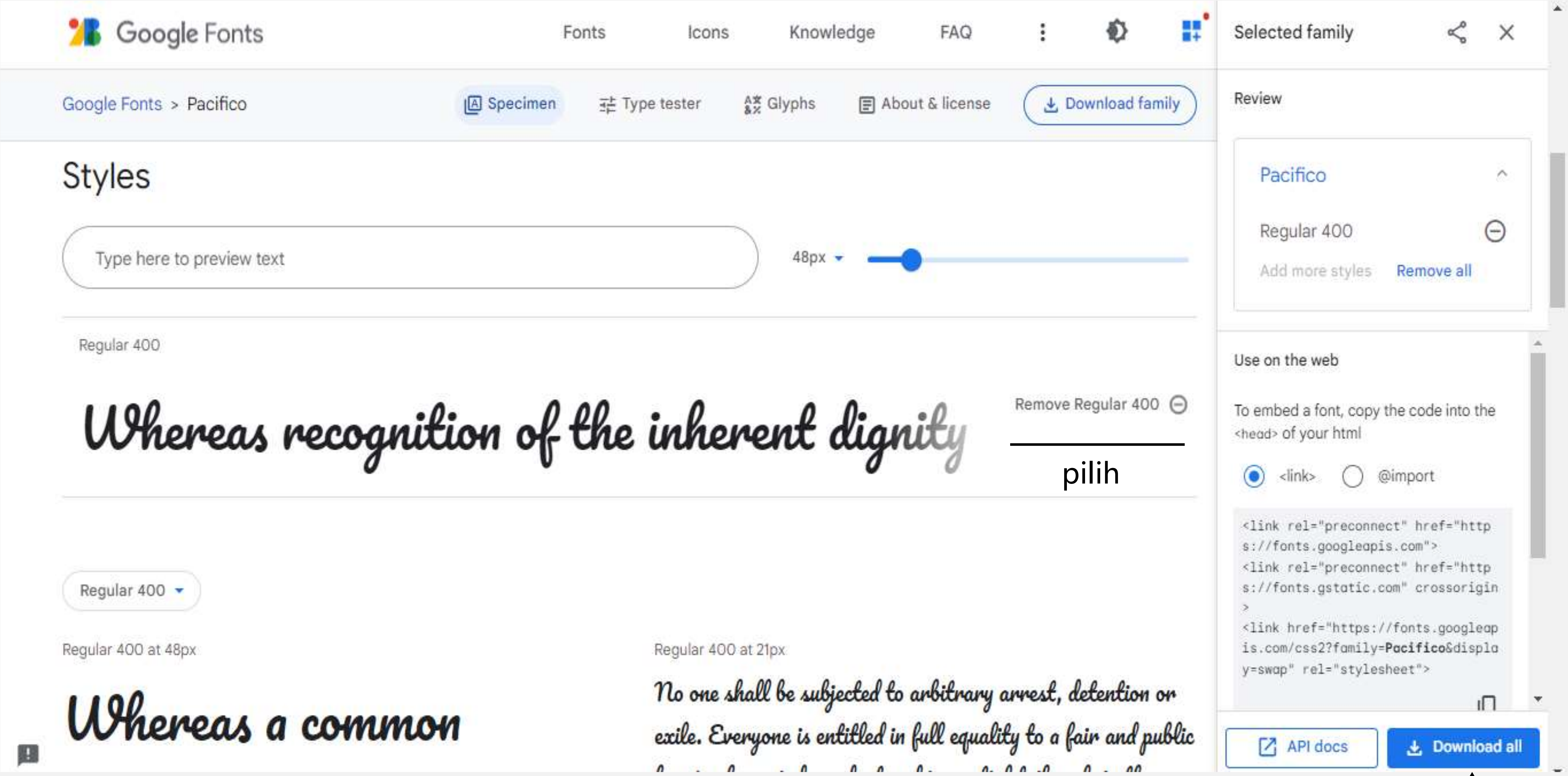
Whereas recognition of the inherent dignity

Noto Sans Nandinagari — 1 style
Google

Noto Sans Nag Mundari — Variable (1 axis)
Google

four

MOJADIPRO



Google Fonts Fonts Icons Knowledge FAQ

Google Fonts > Pacifico

Specimen Type tester Glyphs About & license Download family

Styles

Type here to preview text

48px

Regular 400

Whereas recognition of the inherent dignity

Remove Regular 400 ⊖

pilih

Regular 400

Regular 400 at 48px

Whereas a common

Regular 400 at 21px

No one shall be subjected to arbitrary arrest, detention or exile. Everyone is entitled in full equality to a fair and public

Selected family

Review

Pacifico

Regular 400 ⊖

Add more styles Remove all

Use on the web

To embed a font, copy the code into the <head> of your html

● <link> ○ @import

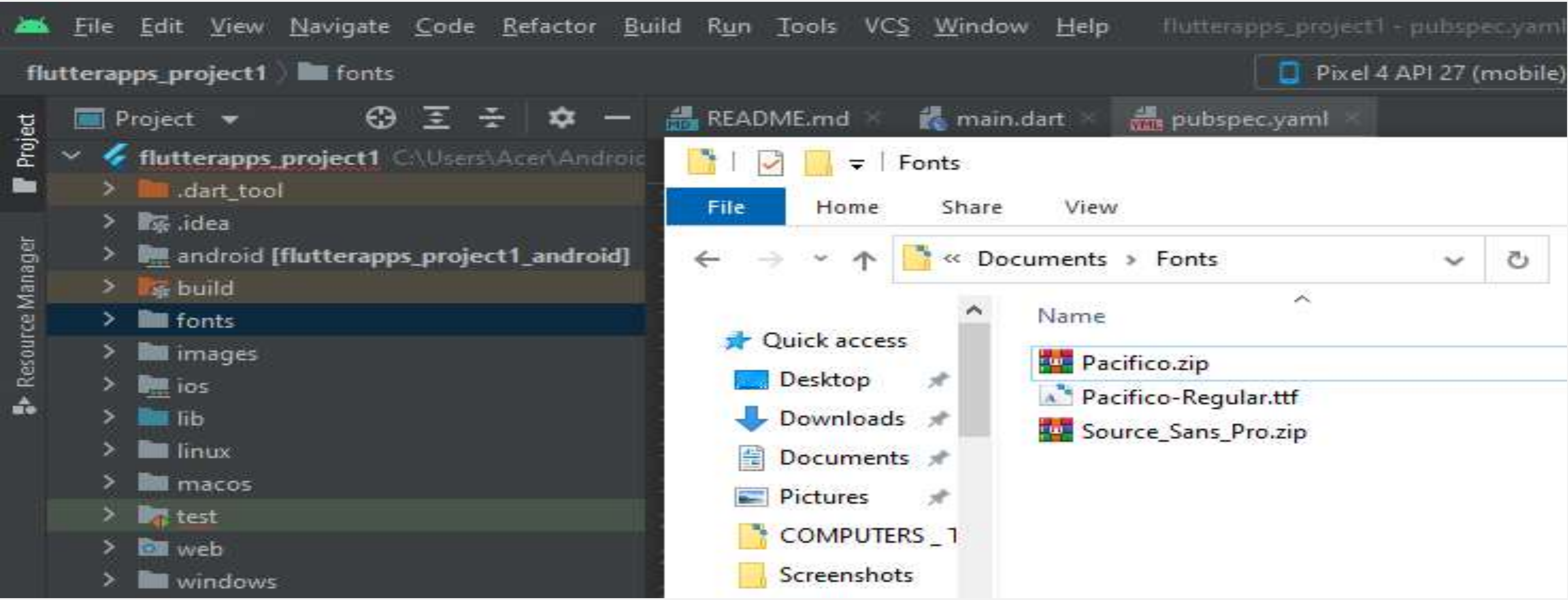
```
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Pacifico&display=swap" rel="stylesheet">
```

API docs Download all

download

four

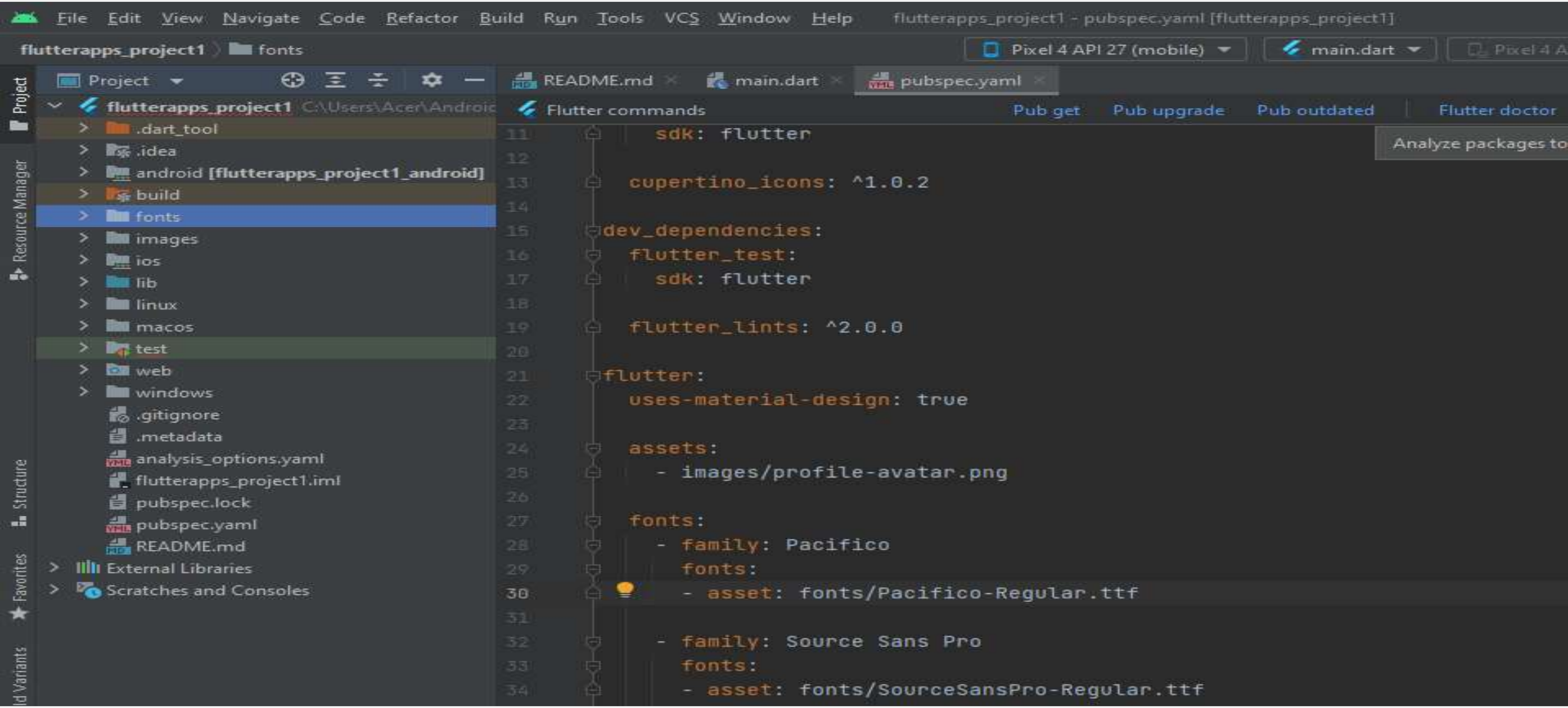
MOJADIPRO



- Buka file zip yang sudah di download
- Ekstract file
- Pindahkan fonts file kedalam fonts folder

four

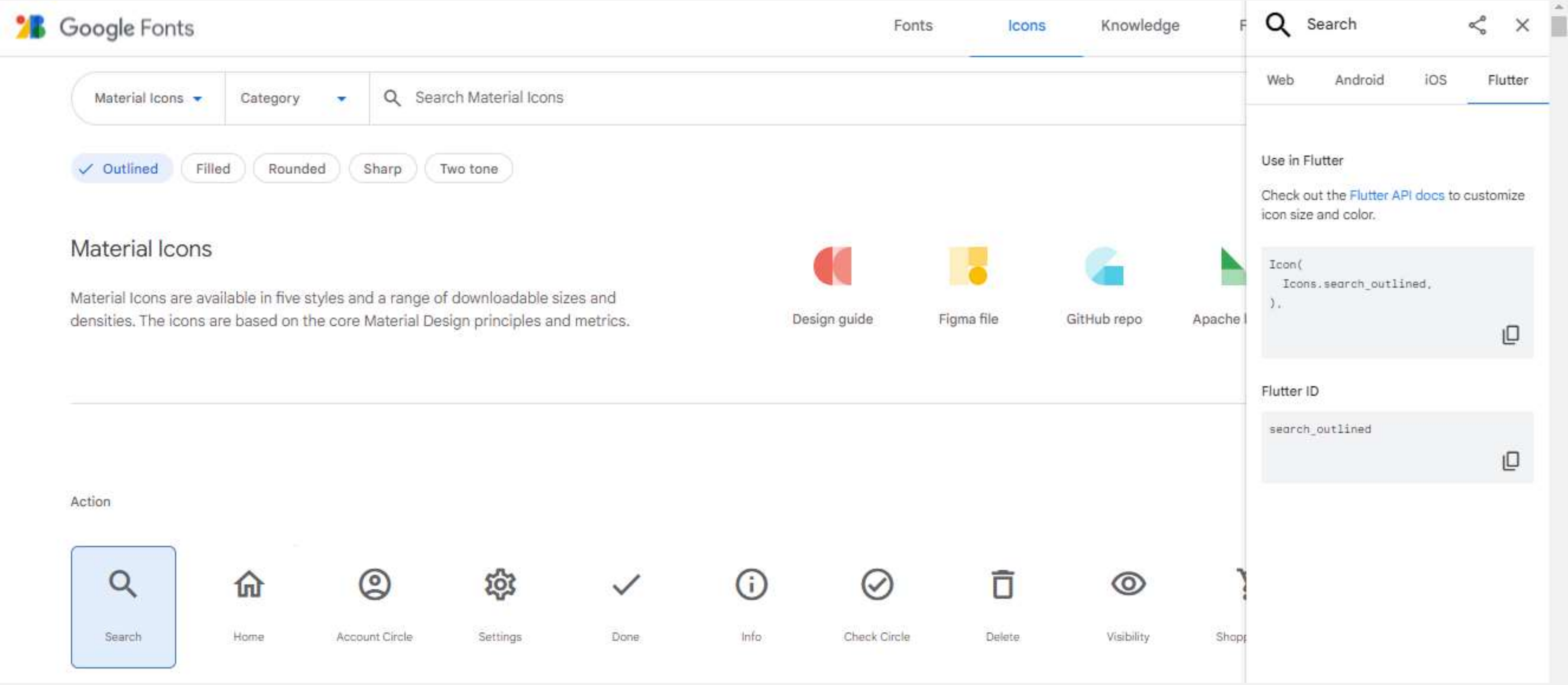
Memasukkan Fonts Pada File Pubspec.yaml



```yaml
    sdk: flutter

  cupertino_icons: ^1.0.2

dev_dependencies:
  flutter_test:
    sdk: flutter

  flutter_lints: ^2.0.0

flutter:
  uses-material-design: true

  assets:
    - images/profile-avatar.png

  fonts:
    - family: Pacifico
      fonts:
        - asset: fonts/Pacifico-Regular.ttf

    - family: Source Sans Pro
      fonts:
        - asset: fonts/SourceSansPro-Regular.ttf
```

Ketik nama fonts pada assets → save → klik pub get → run emulator

Memasukkan Materials Icons



Material Icons

Material Icons are available in five styles and a range of downloadable sizes and densities. The icons are based on the core Material Design principles and metrics.

Use in Flutter

Check out the Flutter API docs to customize icon size and color.

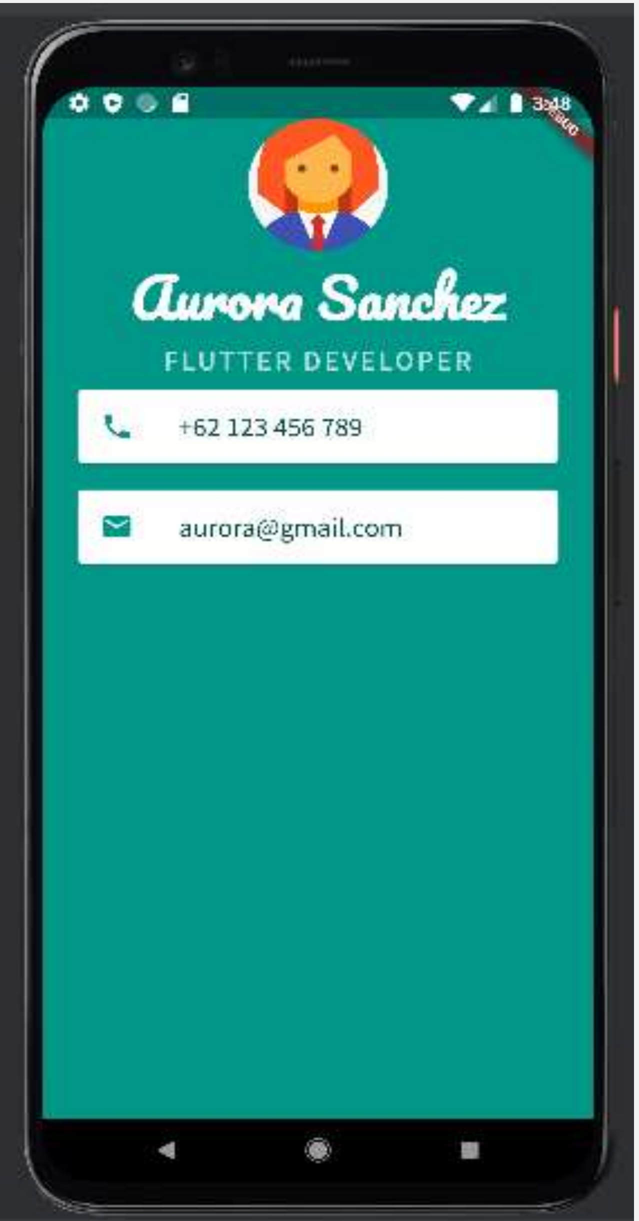
```
Icon(
  Icons.search_outlined,
),
```

Flutter ID

```
search_outlined
```

Buka Google Fonts → pilih Icons → klik icon yang akan dipakai → Pilih Flutter → Salin kode pada Android Studio

Hasil
User Interface

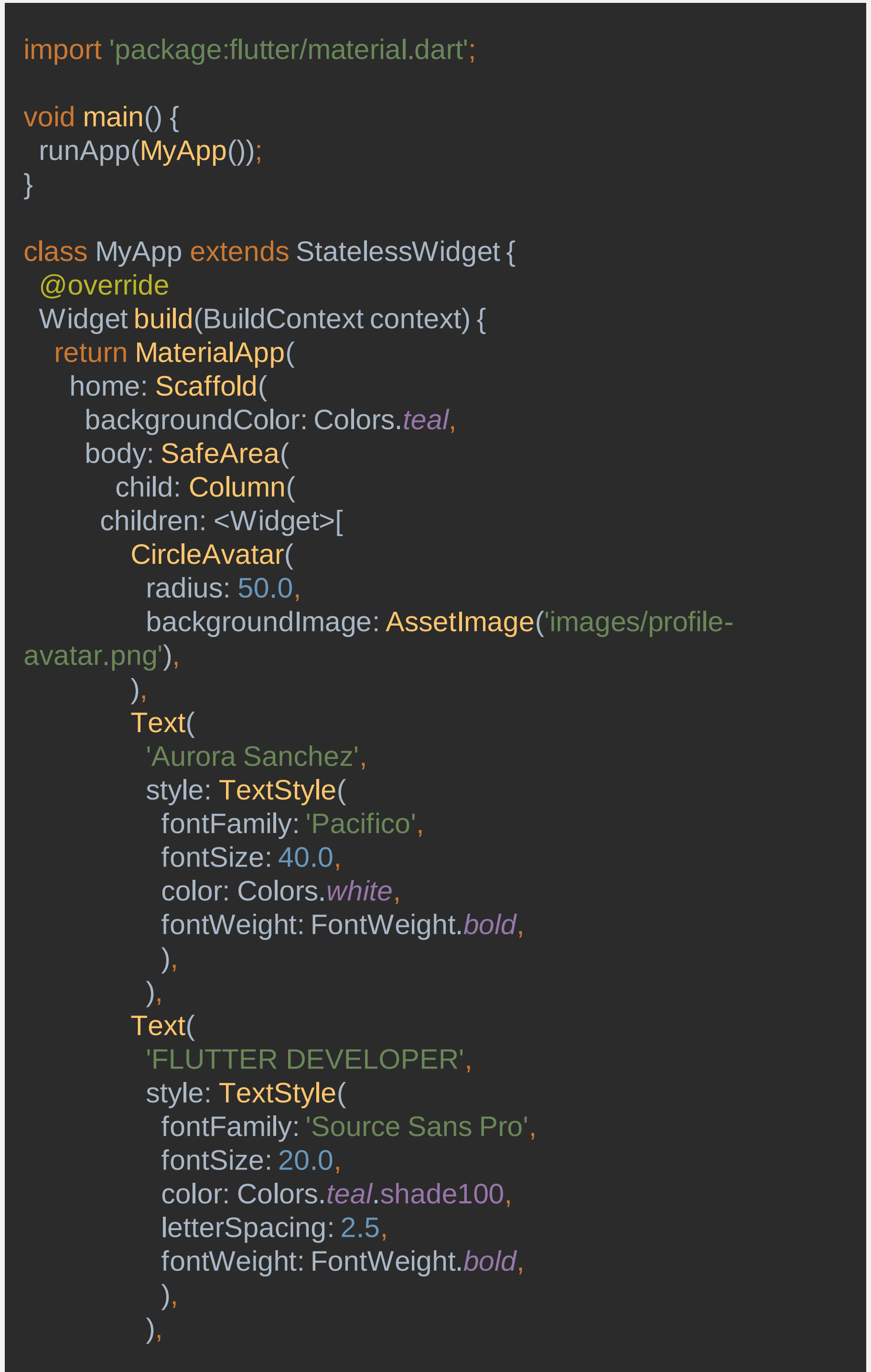


Aurora Sanchez

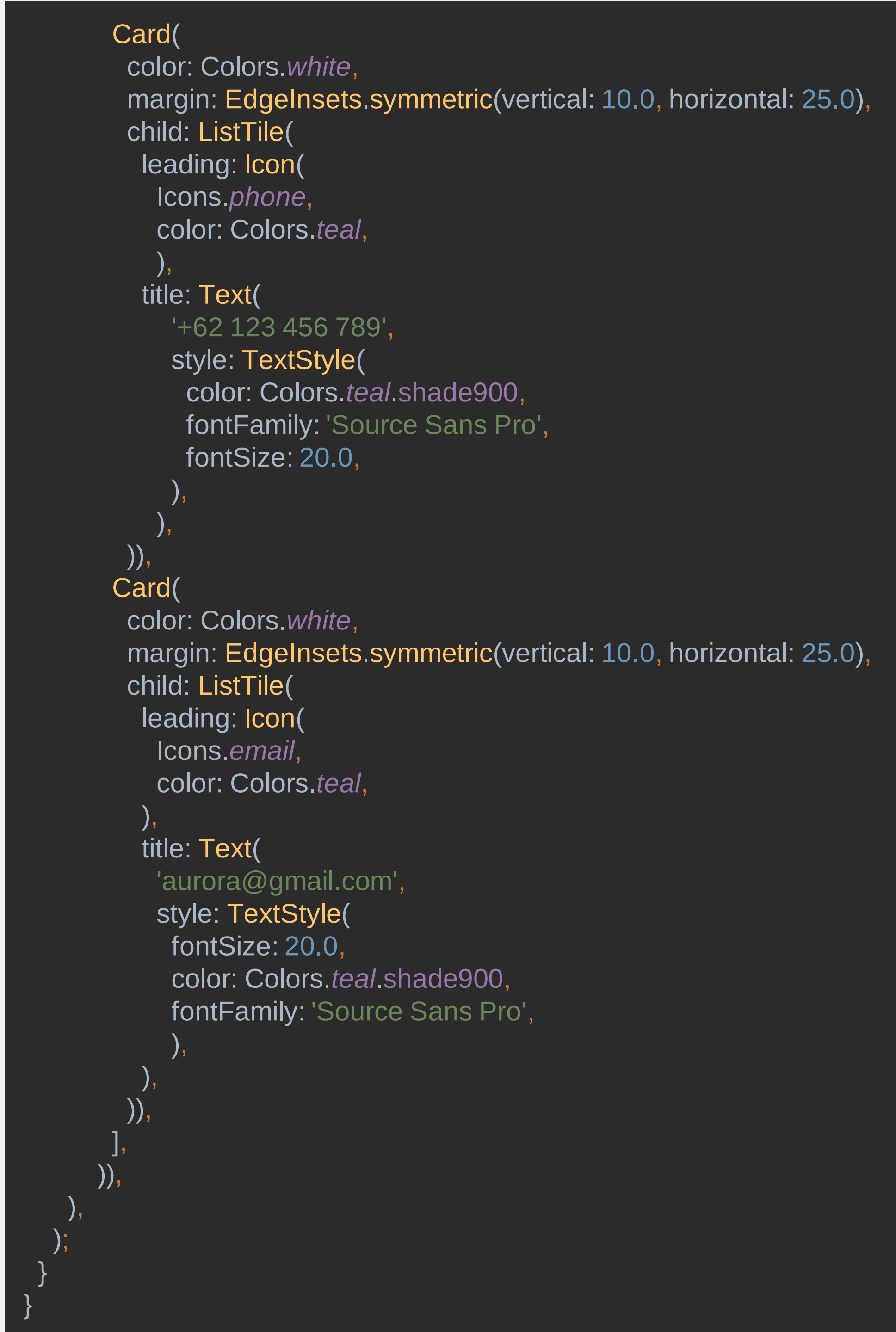
FLUTTER DEVELOPER

☎ +62 123 456 789

✉ aurora@gmail.com



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

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        backgroundColor: Colors.teal,
        body: SafeArea(
            child: Column(
          children: <Widget>[
            CircleAvatar(
              radius: 50.0,
              backgroundImage: AssetImage('images/profile-avatar.png'),
            ),
            Text(
              'Aurora Sanchez',
              style: TextStyle(
                fontFamily: 'Pacifico',
                fontSize: 40.0,
                color: Colors.white,
                fontWeight: FontWeight.bold,
                ),
              ),
            Text(
              'FLUTTER DEVELOPER',
              style: TextStyle(
                fontFamily: 'Source Sans Pro',
                fontSize: 20.0,
                color: Colors.teal.shade100,
                letterSpacing: 2.5,
                fontWeight: FontWeight.bold,
                ),
              ),
```

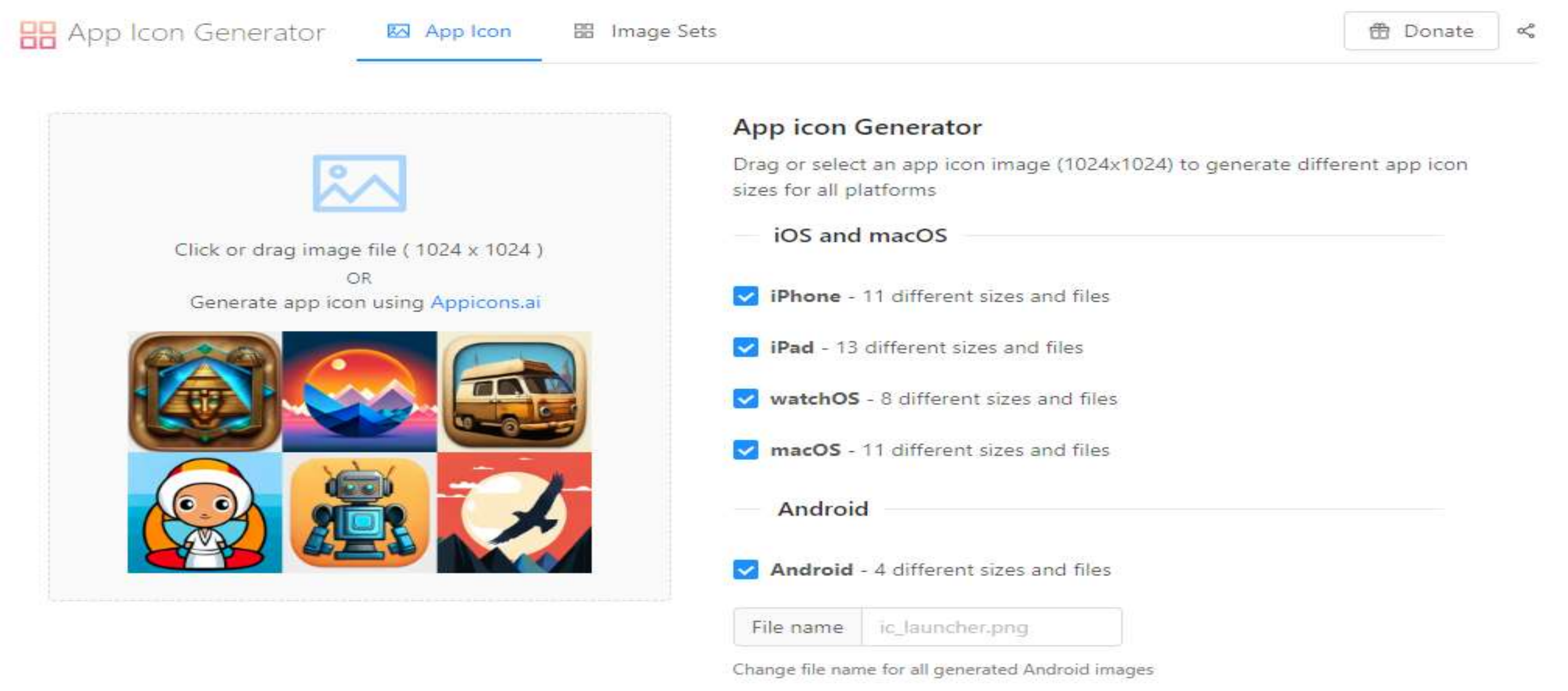


```
            Card(
              color: Colors.white,
              margin: EdgeInsets.symmetric(vertical: 10.0, horizontal: 25.0),
              child: ListTile(
                leading: Icon(
                  Icons.phone,
                  color: Colors.teal,
                  ),
                title: Text(
                    '+62 123 456 789',
                    style: TextStyle(
                      color: Colors.teal.shade900,
                      fontFamily: 'Source Sans Pro',
                      fontSize: 20.0,
                    ),
                  ),
              )),
            Card(
              color: Colors.white,
              margin: EdgeInsets.symmetric(vertical: 10.0, horizontal: 25.0),
              child: ListTile(
                leading: Icon(
                  Icons.email,
                  color: Colors.teal,
                ),
                title: Text(
                  'aurora@gmail.com',
                  style: TextStyle(
                    fontSize: 20.0,
                    color: Colors.teal.shade900,
                    fontFamily: 'Source Sans Pro',
                    ),
                ),
              )),
          ],
        )),
      ),
    );
  }
}
```

five

Mempelajari Cara Memasukkan App Icon

Memasukkan App Icon



Click or drag image file (1024 x 1024)
OR
Generate app icon using Appicons.ai

App icon Generator

Drag or select an app icon image (1024x1024) to generate different app icon sizes for all platforms

iOS and macOS

☑ **iPhone** - 11 different sizes and files

☑ **iPad** - 13 different sizes and files

☑ **watchOS** - 8 different sizes and files

☑ **macOS** - 11 different sizes and files

Android

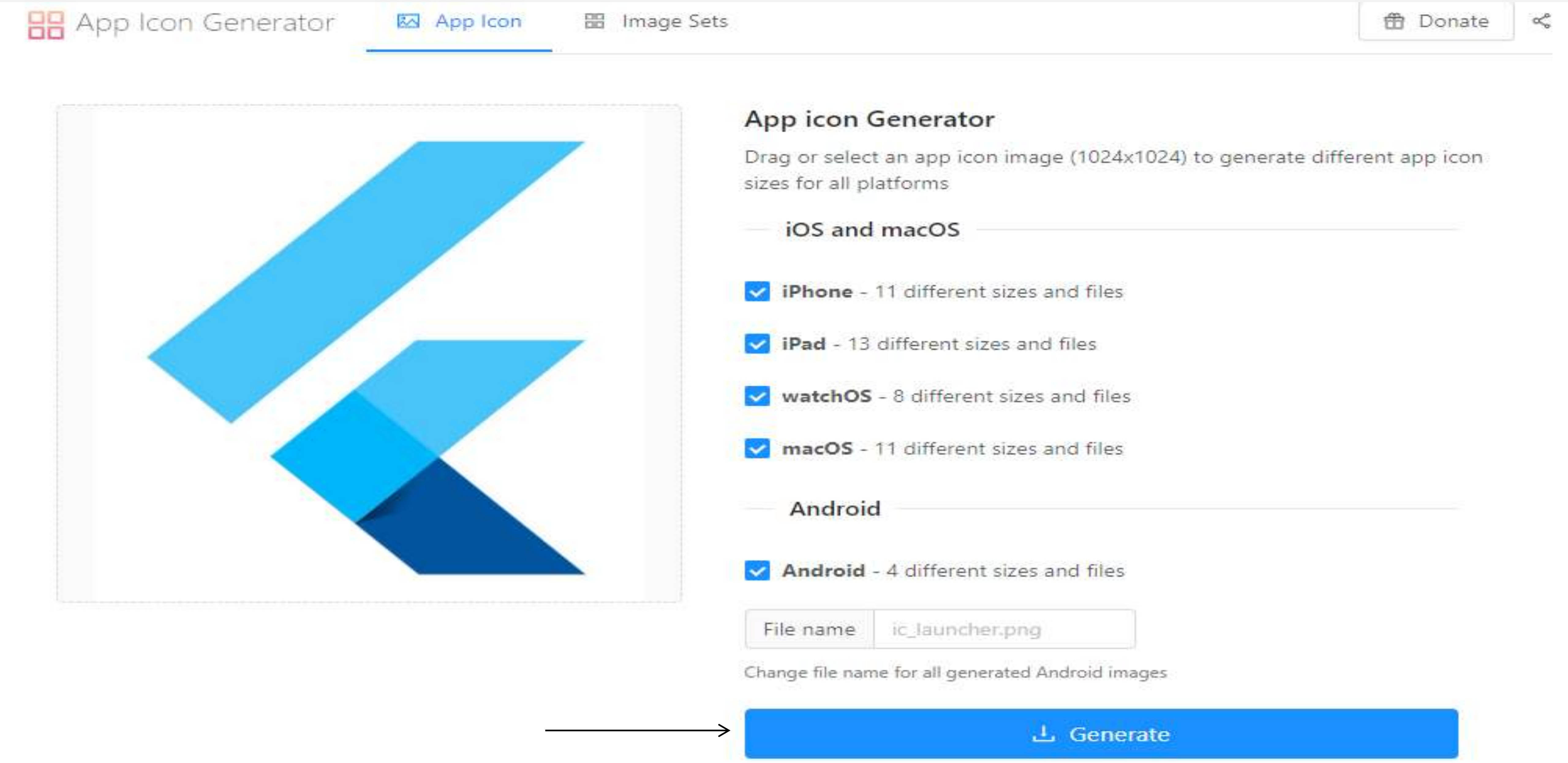
☑ **Android** - 4 different sizes and files

File name: ic_launcher.png

Change file name for all generated Android images

Buka App Icon Generator → Klik image file

Memasukkan App Icon



App icon Generator

Drag or select an app icon image (1024x1024) to generate different app icon sizes for all platforms

iOS and macOS

- ☑ **iPhone** - 11 different sizes and files
- ☑ **iPad** - 13 different sizes and files
- ☑ **watchOS** - 8 different sizes and files
- ☑ **macOS** - 11 different sizes and files

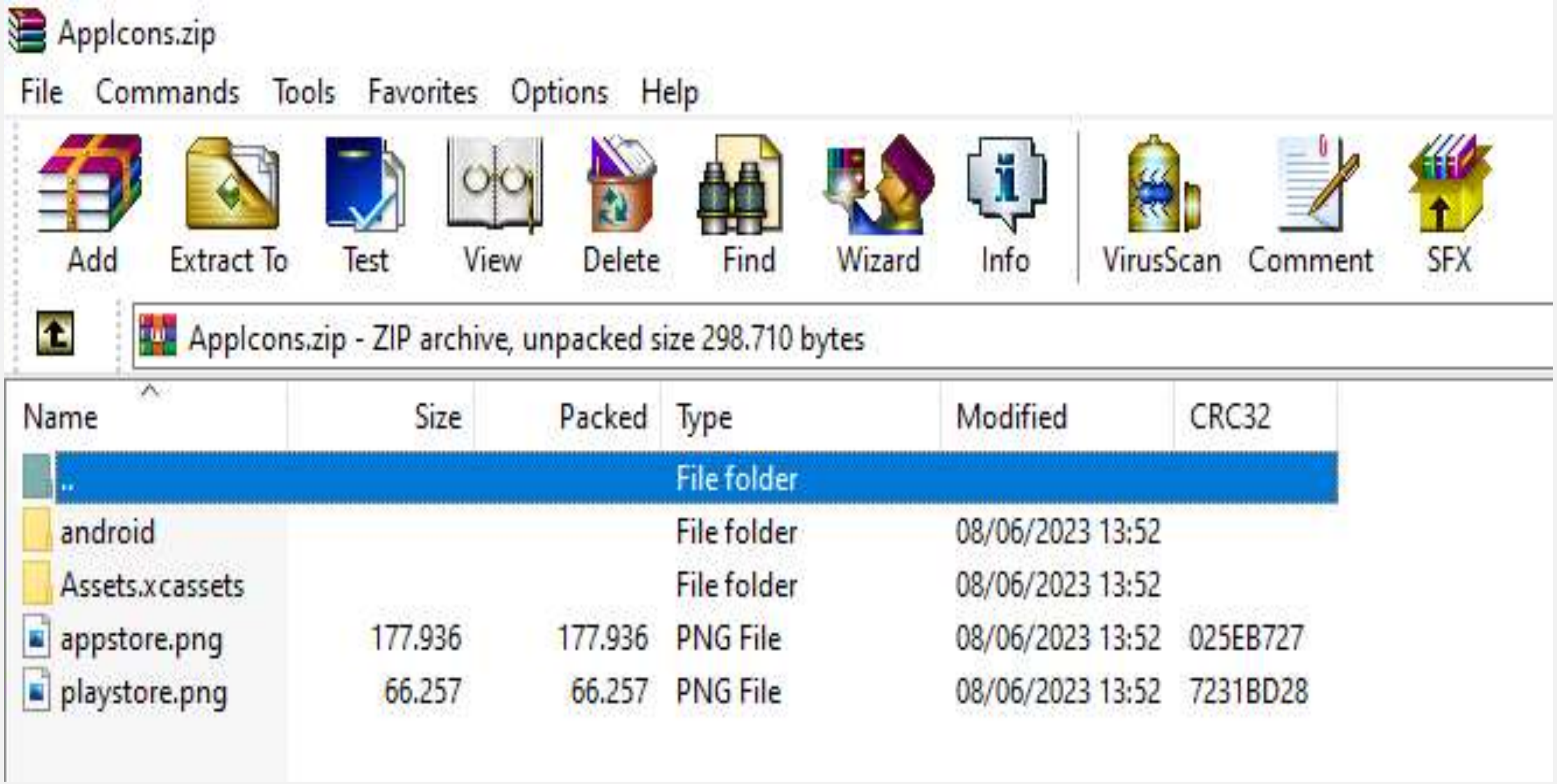
Android

- ☑ **Android** - 4 different sizes and files

File name: ic_launcher.png

Change file name for all generated Android images

Memasukkan App Icon

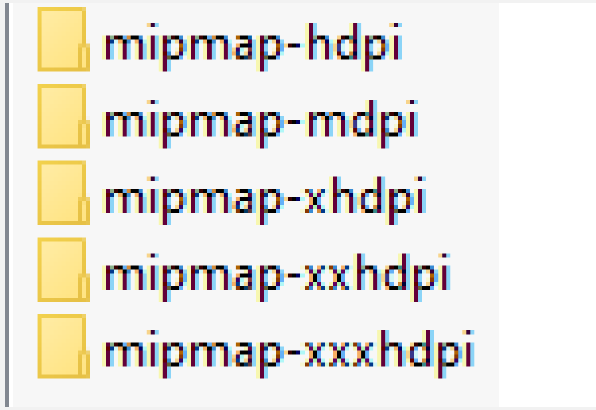


AppIcons.zip - ZIP archive, unpacked size 298.710 bytes

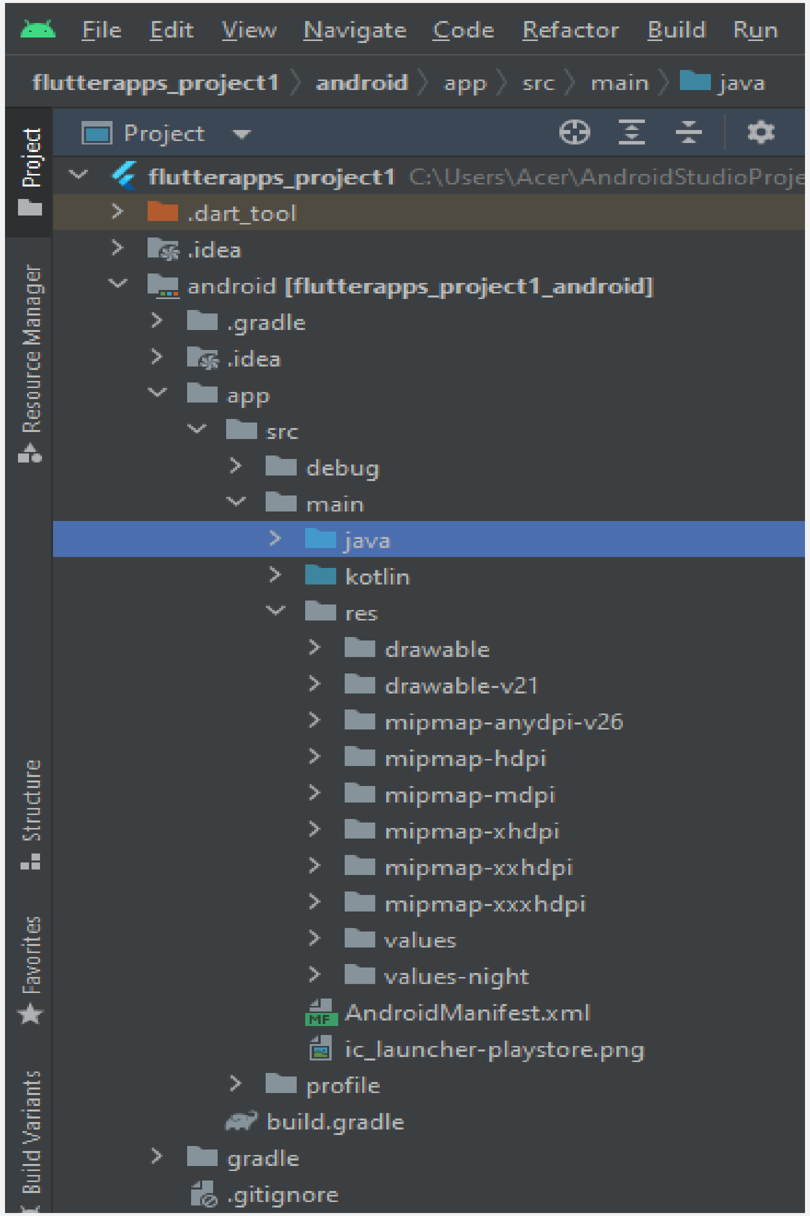
Name	Size	Packed	Type	Modified	CRC32
..			File folder		
android			File folder	08/06/2023 13:52	
Assets.xcassets			File folder	08/06/2023 13:52	
appstore.png	177.936	177.936	PNG File	08/06/2023 13:52	025EB727
playstore.png	66.257	66.257	PNG File	08/06/2023 13:52	7231BD28

Buka file zip yang telah di download → Pilih Android → Ekstrak file five

Memasukkan App Icon



- mipmap-hdpi
- mipmap-mdpi
- mipmap-xhdpi
- mipmap-xxhdpi
- mipmap-xxxhdpi

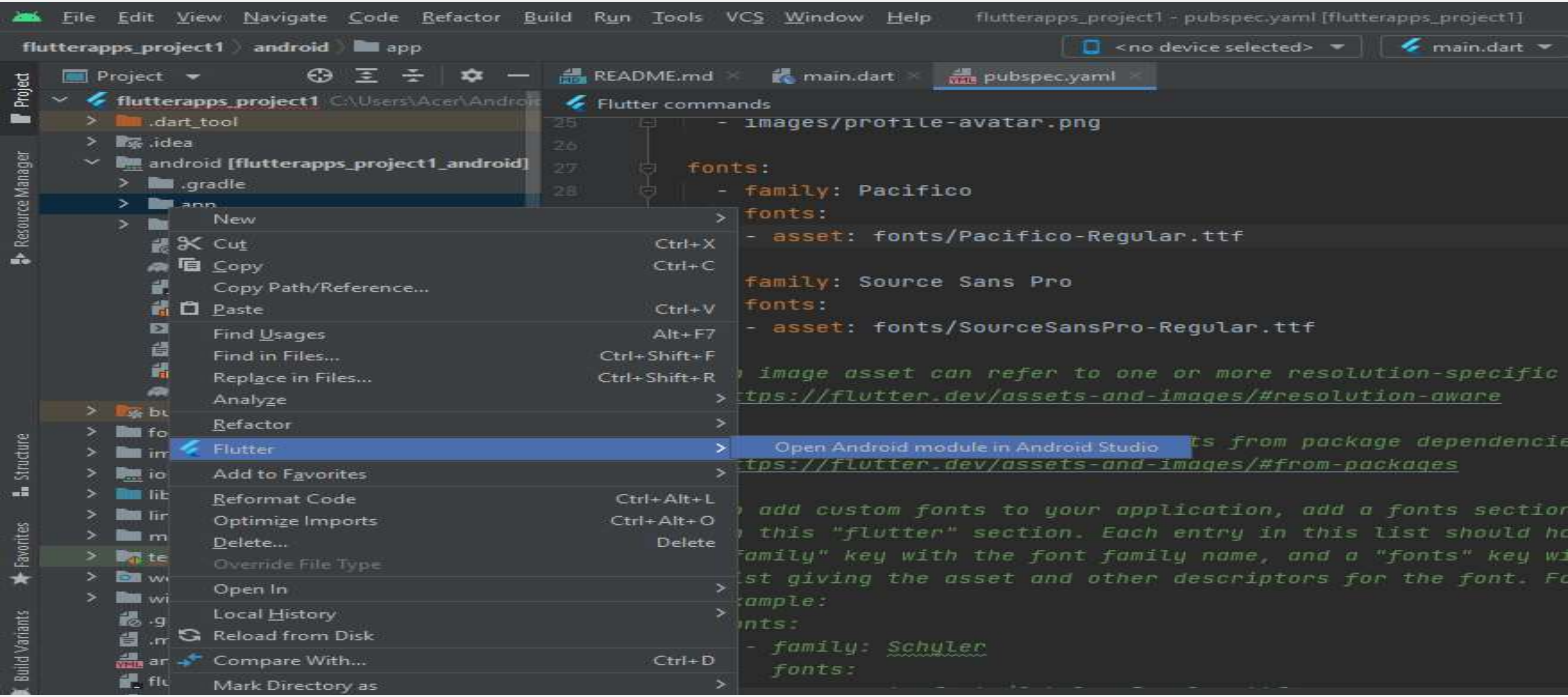


flutterapps_project1 〉 android 〉 app 〉 src 〉 main 〉 java

```
flutterapps_project1  C:\Users\Acer\AndroidStudioProje
  .dart_tool
  .idea
  android [flutterapps_project1_android]
    .gradle
    .idea
    app
      src
        debug
        main
          java
          kotlin
          res
            drawable
            drawable-v21
            mipmap-anydpi-v26
            mipmap-hdpi
            mipmap-mdpi
            mipmap-xhdpi
            mipmap-xxhdpi
            mipmap-xxxhdpi
            values
            values-night
          AndroidManifest.xml
          ic_launcher-playstore.png
        profile
      build.gradle
    gradle
    .gitignore
```

Masukkan folder ke dalam file res

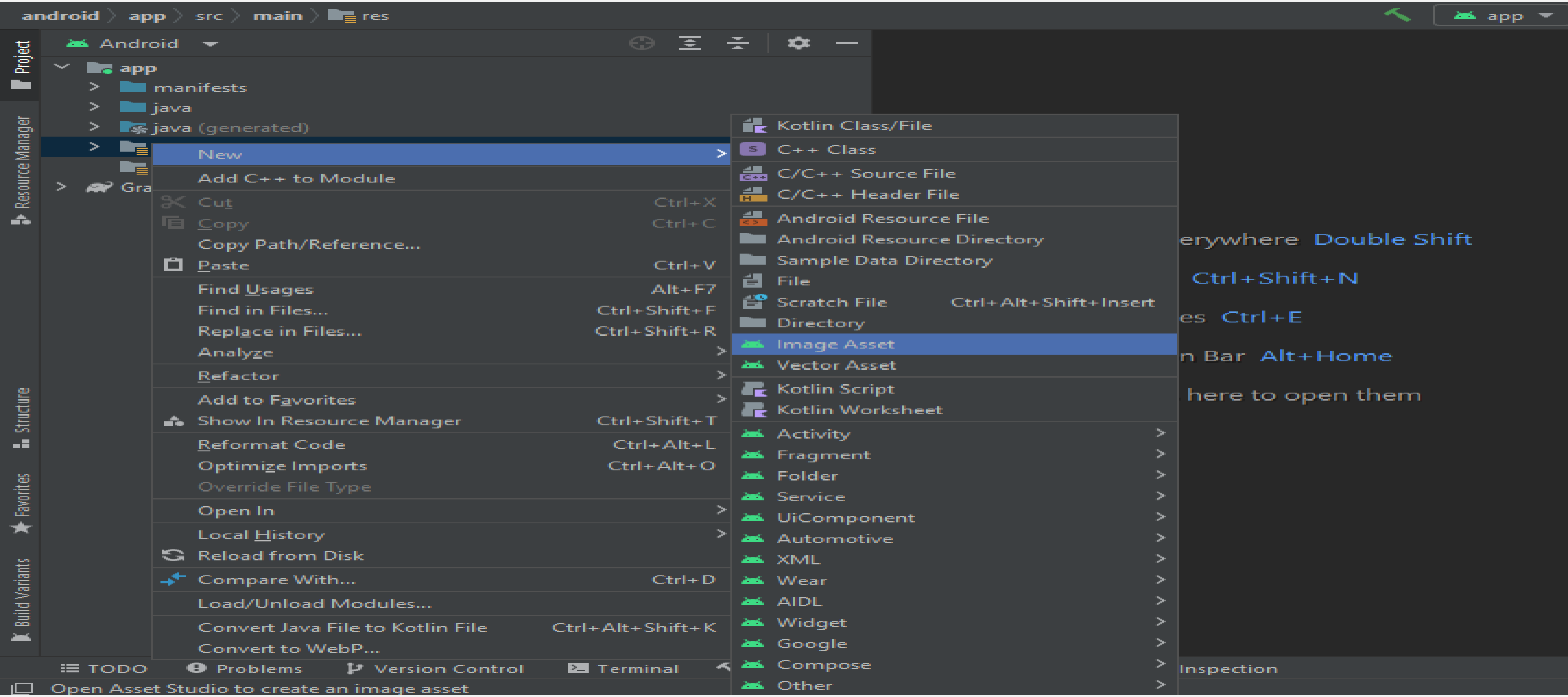
Memasukkan App Icon



```
    - images/profile-avatar.png

  fonts:
    - family: Pacifico
      fonts:
        - asset: fonts/Pacifico-Regular.ttf

      family: Source Sans Pro
      fonts:
        - asset: fonts/SourceSansPro-Regular.ttf

  image asset can refer to one or more resolution-specific
  tps://flutter.dev/assets-and-images/#resolution-aware

                                ts from package dependencie
  tps://flutter.dev/assets-and-images/#from-packages

  add custom fonts to your application, add a fonts sectio
  this "flutter" section. Each entry in this list should ha
  amily" key with the font family name, and a "fonts" key wi
  st giving the asset and other descriptors for the font. Fo
  ample:
  nts:
    - family: Schyler
      fonts:
```

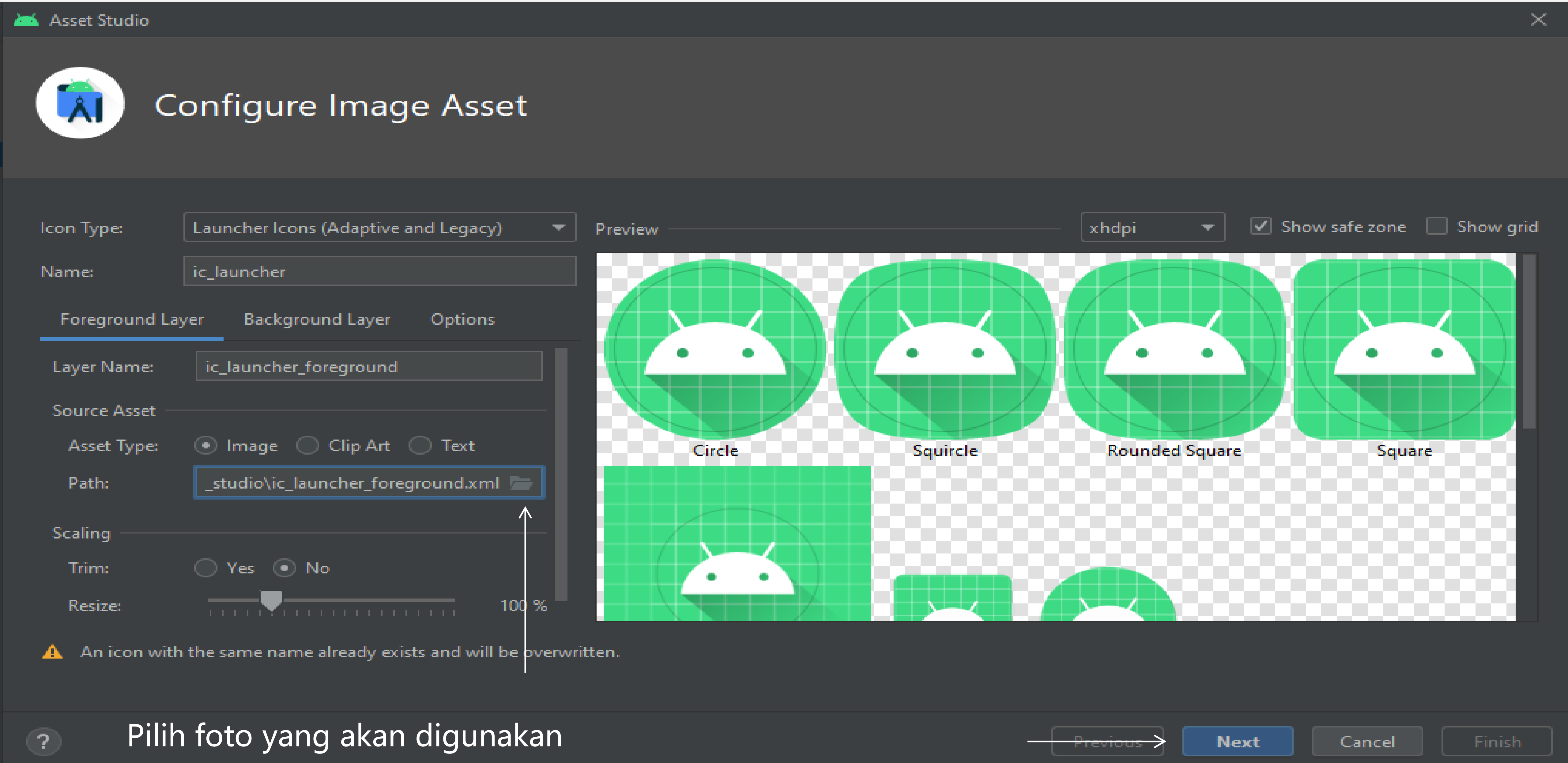
Pilih Android → Klik kanan pada App → Flutter → Open Android module

Memasukkan App Icon



Pilih Android → Klik kanan pada App → Flutter → Open Android module

Memasukkan App Icon



Asset Studio

Configure Image Asset

Icon Type: Launcher Icons (Adaptive and Legacy)

Name: ic_launcher

Foreground Layer | Background Layer | Options

Layer Name: ic_launcher_foreground

Source Asset

Asset Type: ● Image ○ Clip Art ○ Text

Path: _studio\ic_launcher_foreground.xml

Scaling

Trim: ○ Yes ● No

Resize: 100 %

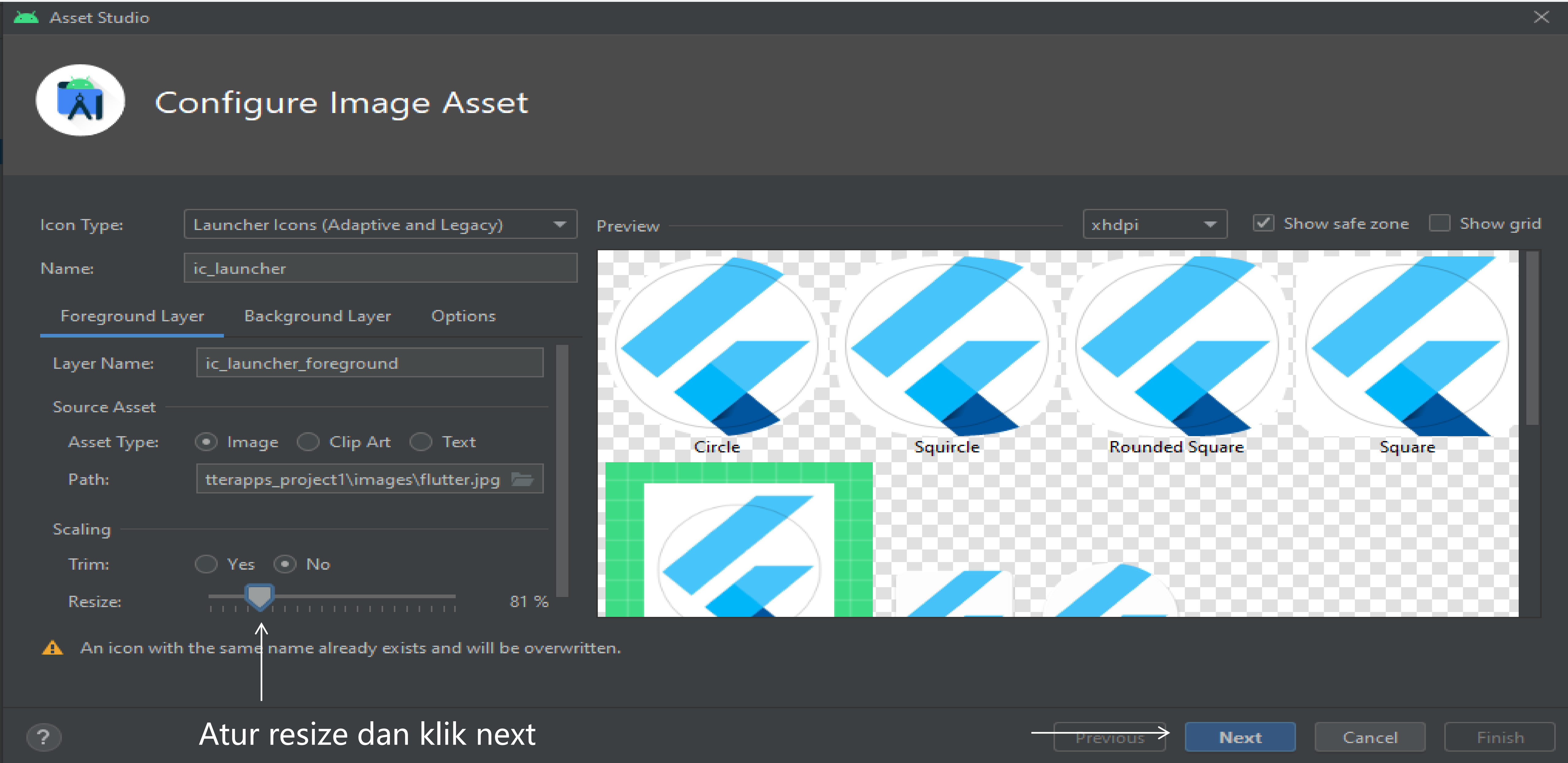
⚠ An icon with the same name already exists and will be overwritten.

Pilih foto yang akan digunakan

five

Memasukkan App Icon

MOJADIPRO



Asset Studio

Configure Image Asset

Icon Type: Launcher Icons (Adaptive and Legacy)

Name: ic_launcher

Layer Name: ic_launcher_foreground

Source Asset

Asset Type: ● Image ○ Clip Art ○ Text

Path: tterapps_project1\images\flutter.jpg

Scaling

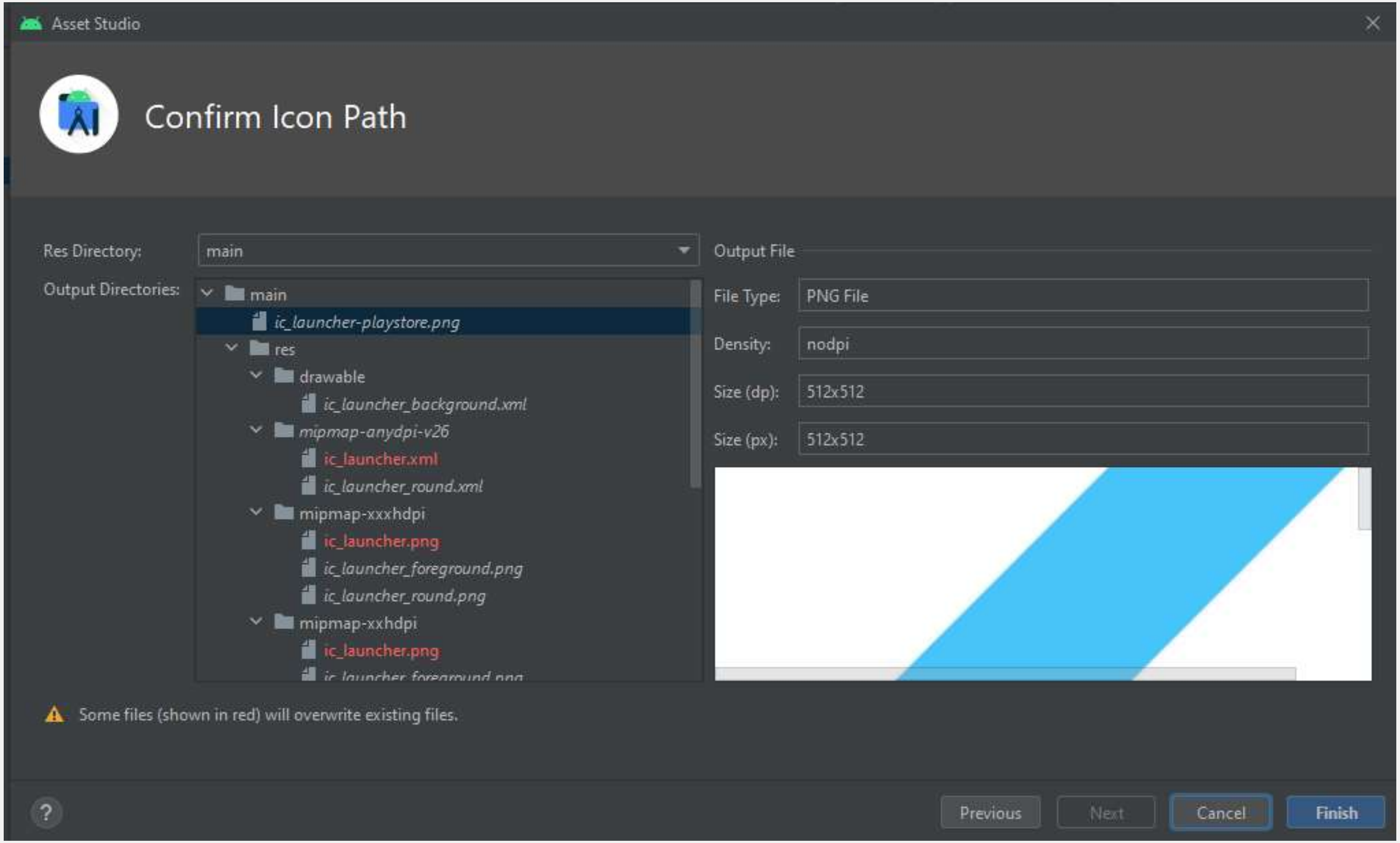
Trim: ○ Yes ● No

Resize: 81 %

An icon with the same name already exists and will be overwritten.

Atur resize dan klik next

Memasukkan App Icon



Asset Studio

Confirm Icon Path

Res Directory: main

Output Directories:

```
main
  ic_launcher-playstore.png
  res
    drawable
      ic_launcher_background.xml
    mipmap-anydpi-v26
      ic_launcher.xml
      ic_launcher_round.xml
    mipmap-xxxhdpi
      ic_launcher.png
      ic_launcher_foreground.png
      ic_launcher_round.png
    mipmap-xxhdpi
      ic_launcher.png
      ic_launcher_foreground.png
```

Output File

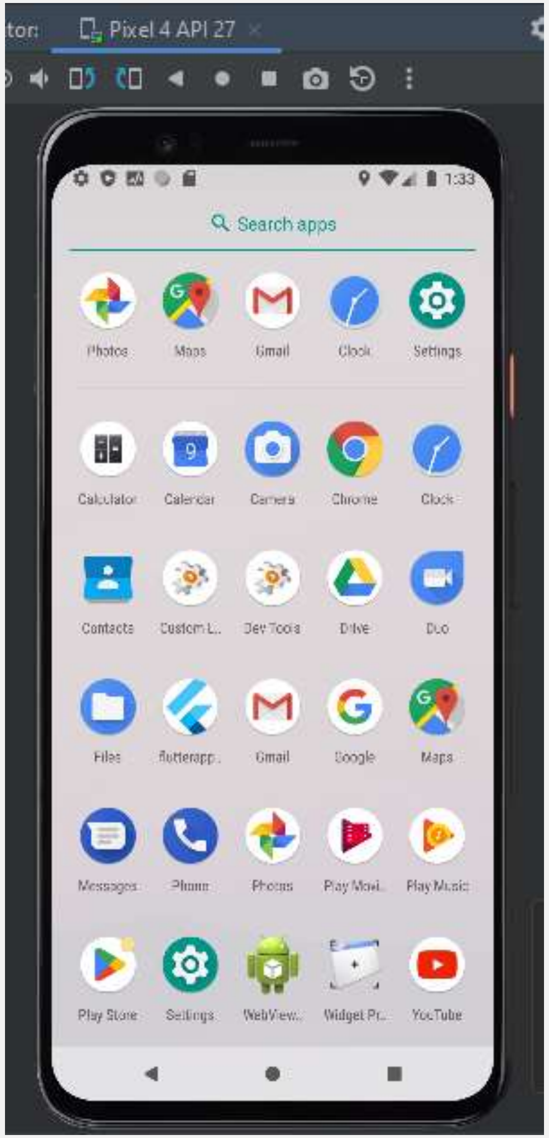
File Type: PNG File

Density: nodpi

Size (dp): 512x512

Size (px): 512x512

⚠ Some files (shown in red) will overwrite existing files.



Klik finish → lalu buka di emulator

FLUTTER

MOJADIPRO

TERIMA KASIH