

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.

## Scaffold



## AppBar



## SafeArea



## Container



## Text

```
onPanUpdate:  
DragUpdateDetails(Offset(0.3, 0.0))
```

## RichText



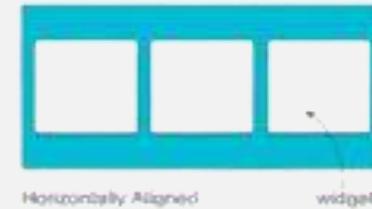
Flutter World for Mobile

## Column



widgets

## Row

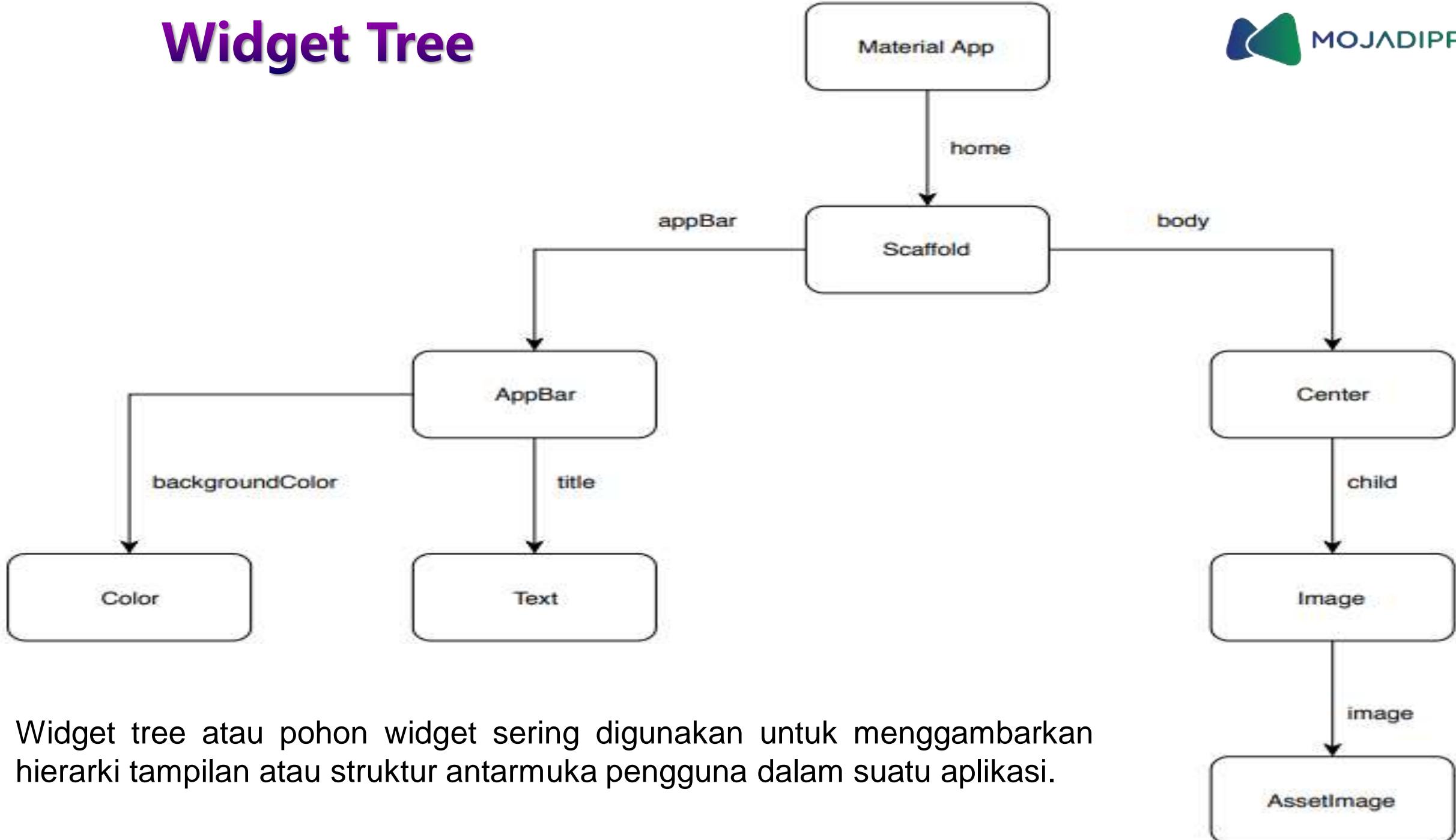


widgets

## Button



# Widget Tree

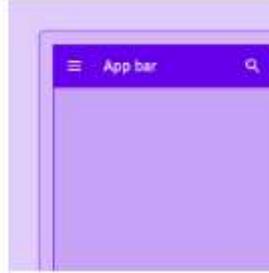


Widget tree atau pohon widget sering digunakan untuk menggambarkan hierarki tampilan atau struktur antarmuka pengguna dalam suatu aplikasi.



# Material App

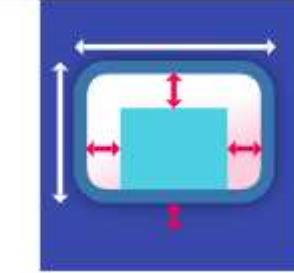
MaterialApp adalah kelas atau widget yang menjadi komponen utama atau inti dari aplikasi flutter. Widget yang dapat diakses menggunakan kelas MaterialApp :

**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**

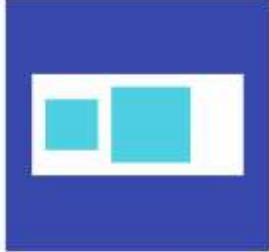
A convenience widget that combines common painting, positioning, and sizing widgets.

**Image**

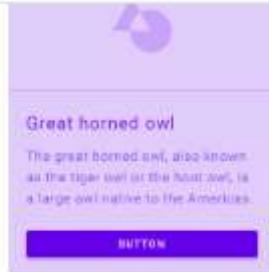
A widget that displays an image.

**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.

**Great horned owl**

The great horned owl, also known as the tiger owl or the horned owl, is a large owl native to the Americas.

**BUTTON****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.

**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

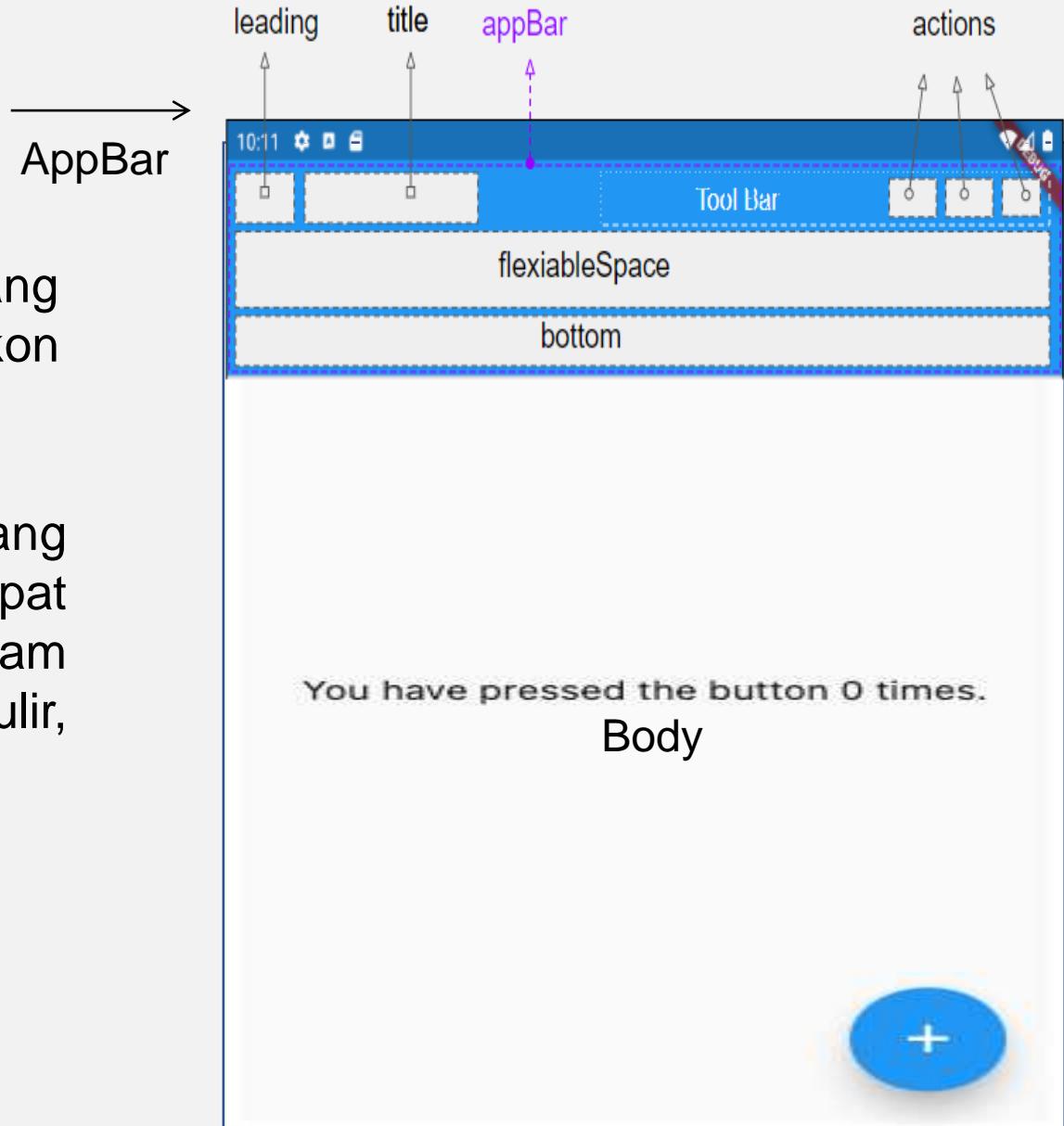
Di dalam Scaffold terdapat AppBar dan Body

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.

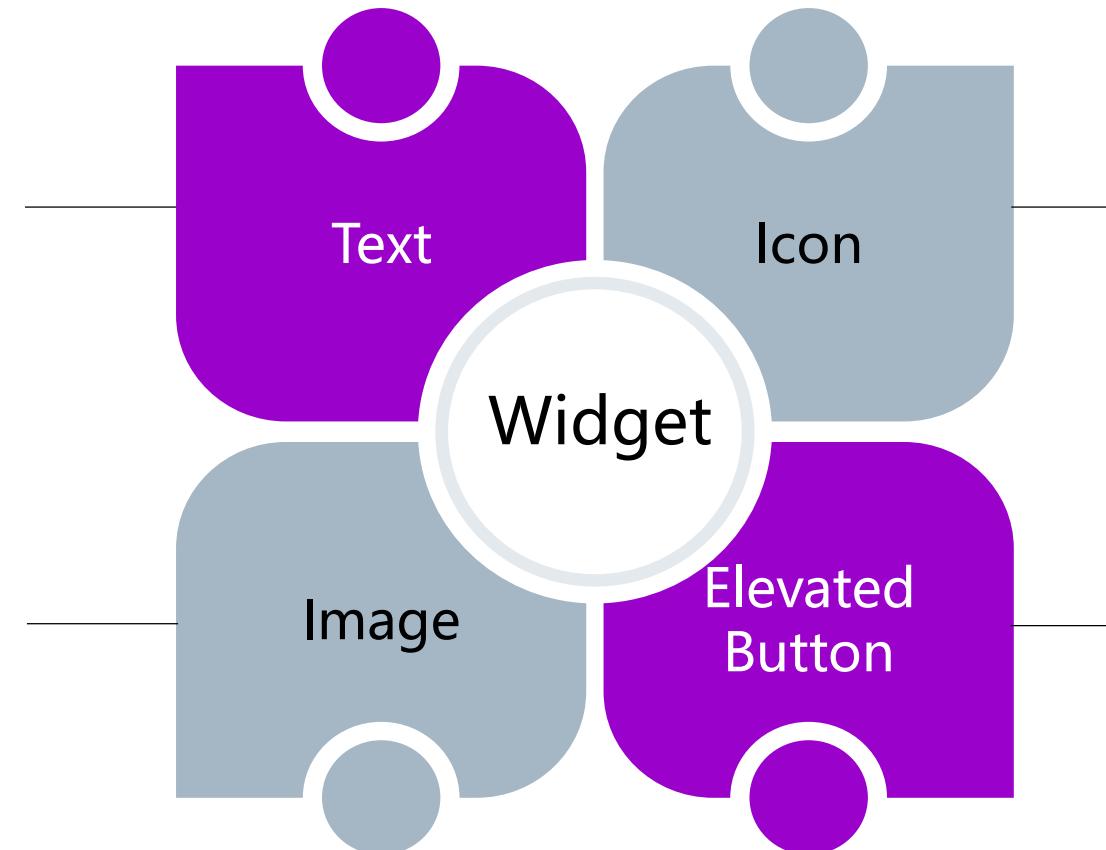




## Berikut adalah beberapa widget dasar yang perlu diketahui

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

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



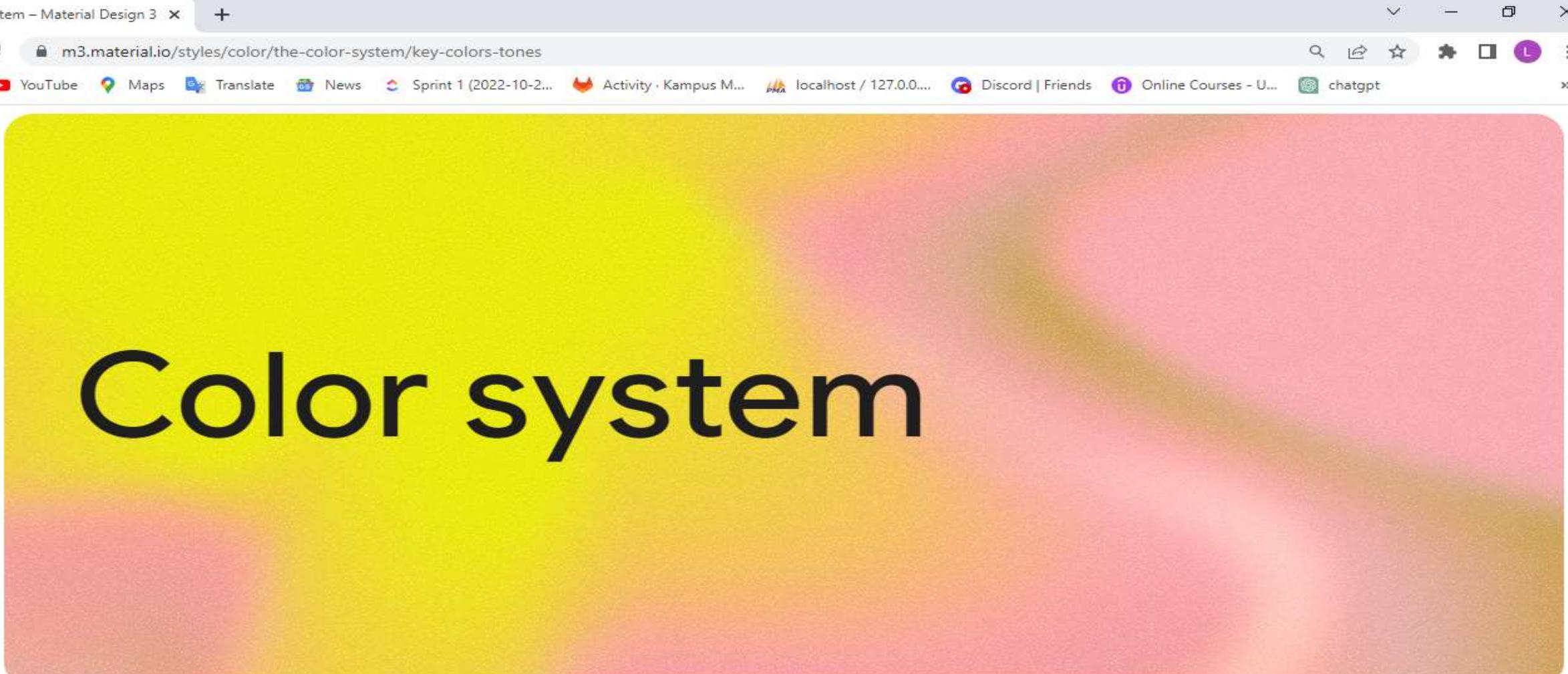
Suatu simbol atau gambar kecil yang digunakan untuk mewakili suatu objek, aksi, atau konsep tertentu.

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

one



# Penggunaan Custom Warna pada Material Design



The screenshot shows a web browser window with the URL [m3.material.io/styles/color/the-color-system/key-colors-tones](https://m3.material.io/styles/color/the-color-system/key-colors-tones). The page title is "Color system". On the left, there is a sidebar with icons and labels: Home, Get started, Develop, Foundations, Styles (which is selected), Components, and Blog. At the bottom, there are tabs for "Key colors & tones" (selected), Color roles, Tokens, Custom colors, and Accessibility.

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



Home



Get started



Develop



Foundations



Styles



Components



Blog

## Key colors &amp; tones

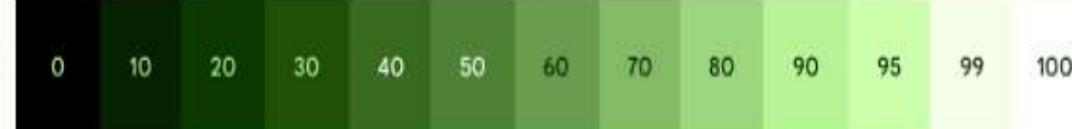
## Color roles

## Tokens

## Custom colors

## Accessibility

Primary Key Color



On this page

## Color system

Key colors

Tonal palettes

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



Home



Get started



Develop



Foundations



Styles



Components



Blog



Key colors &amp; tones

Color roles

Tokens

Custom colors

Accessibility



On this page

## Color system

Adding custom colors to a scheme



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

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(
5     MaterialApp(
6       home: Scaffold(
7         backgroundColor: green,
8         appBar: AppBar(
9           // AppBar
10          // Scaffold
11        ), // MaterialApp
12      ),
13    ),
14  );
15}
```

The screenshot shows the Android Studio interface with the main.dart file open. The code defines a main() function that runs an MaterialApp. Inside the MaterialApp, a Scaffold is defined with a green background color. An AppBar is also defined. A color palette dropdown is open over the 'green' color reference in the code, showing various shades of green: green, greenAccent, lightGreen, and lightGreenAccent. The dropdown includes a message: 'Press Ctrl+Space again for more results' and a 'Next Tip' link.

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



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
```



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

one



# Memasukkan Network Image



flutter - Google Penelusuran

flutter-logo-sharing.png (937x46)

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

Gmail YouTube Maps Translate News Sprint 1 (2022-10-2... Activity · Kampus M... localhost / 127.0.0.... Discord | Friends Online Courses - U... chatgpt



Copy URL dan masukkan ke dalam kode

```
Pixel 4 API 27 (mobile) main.dart Pixel 4 API 27 Emulator: Pixel 4 API 27 README.md × main.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/flutter-logo.png'), // Image ), // Center ), // Scaffold ), // MaterialApp ), ); }
```

The screenshot shows a Flutter application running on an Android emulator. The app's title bar says "My Apps". The main screen displays the Flutter logo and the word "Flutter". The code editor on the left shows the Dart code for the application, which defines a scaffold with a blue-grey background and a central image of the Flutter logo.



two

Mengclone Project Flutter  
dari Github



GitHub - londonappbrewery/mi\_card\_flutter

londonappbrewery / mi\_card\_flutter Public

Code Issues 25 Pull requests 6 Actions Projects Security Insights

master 1 branch 0 tags Go to file Code About

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

[www.appbrewery.co](#)

Readme 401 stars 31 watching 1.1k forks Report repository

A screenshot of a web browser showing a GitHub repository page for "londonappbrewery / mi\_card\_flutter". The repository is public and contains 1 branch and 0 tags. The master branch has 2 commits by user "TheMuellenator" dated Nov 18, 2021. The commits are: "updating gradle wrapper", "updating project for android", "updating project for android", "updating project for android", and "updating project for android". The repository has 401 stars, 31 watching, and 1.1k forks. A green arrow points from the "Code" button in the top navigation bar to the "About" section on the right, which describes the repository as "Starter code for the Mi Card Project from the Complete Flutter Development Bootcamp".

two



GitHub - londonappbrewery/mi\_card\_flutter

Gmail YouTube Maps Translate News Sprint 1 (2022-10-2... Activity · Kampus M... localhost / 127.0.0.... Discord | Friends Online Courses - U... chatgpt



Product Solutions Open Source Pricing

Search

Sign in

Sign up

londonappbrewery / mi\_card\_flutter

Public

Notifications

Fork 1.1k

Star 401

Code Issues 25 Pull requests 6 Actions Projects Security Insights

master

1 branch

0 tags

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

Go to file

Code

Local

Codespaces

Clone

HTTPS GitHub CLI

[https://github.com/londonappbrewery/mi\\_card\\_flutter](https://github.com/londonappbrewery/mi_card_flutter)

Copied!



Open with GitHub Desktop

Download ZIP

## About

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

[www.appbrewery.co](http://www.appbrewery.co)

Readme

401 stars

31 watching

1.1k forks

Report repository

two



Welcome to Android Studio

Android Studio  
Chipmunk | 2021.2.1 Pat...

Projects

Customize

Plugins

Learn Android Studio

Search projects

New Project

Open

New Flutter Project

Get from Version Control...

Virtual Device Manager

Profile or Debug APK

SDK Manager

Import an Android Code Sample

- 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

two



Welcome to Android Studio

Android Studio  
Chipmunk | 2021.2.1 Pat...

Projects

Customize

Plugins

Learn Android Studio

Search projects Get from Version Control

New Project X Open New Flutter Project ...

Repository URL

Version control: Git

URL: [https://github.com/londonappbrewery/mi\\_card\\_flutter.git](https://github.com/londonappbrewery/mi_card_flutter.git)

Directory: C:\Users\Acer\AndroidStudioProjects\mi\_card\_flutter

Paste URL

Clone Cancel

two



File Edit View Navigate Code Refactor Build Run Tools Git Window Help mi\_card\_flutter - main.dart

mi\_card\_flutter lib main.dart

Project README.md main.dart

Dart SDK is not configured Download Dart SDK Open Dart settings

```
import 'package:flutter/material.dart';
void main() {
  runApp(
    MaterialApp(
      home: Scaffold(
        backgroundColor: Colors.teal,
        body: Container(),
      ),
    ),
  );
}
```

Commit Pull Requests Scratches and Consoles Structure Favorites

Device Manager Flutter Outline Flutter Inspector Flutter Performance Emulator



two

File Edit View Navigate Settings

mi\_card\_flutter > lib > main.dart

Project .idea android ios lib main.dart pubspec.yaml README.md External Libraries Scratches and Consoles

Appearance & Behavior Keymap Editor Plugins Version Control Build, Execution, Deployment Languages & Frameworks C/C++ Schemas and DTDs Dart Flutter (selected) Kotlin Markdown Template Data Languages Tools Advanced Settings Experimental

SDK

Flutter SDK path: C:\src\flutter

Version:

General

Report usage information to Google Analytics [www.google.com/policies/privacy >](http://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

OK Cancel Apply

Start SDK Open Dart settings Device Manager Flutter Outline Flutter Inspector Flutter Performance Emulator Event Log

two



File Edit View Navigate Code Refactor Build Run Tools Git Window Help mi\_card\_flutter - main.dart

mi\_card\_flutter lib main.dart

Project README.md main.dart

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

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(
5     MaterialApp(
6       home: Scaffold(
7         backgroundColor: Colors.teal,
8         body: Container(),
9       ),
10    ),
11  );
12}
13
```

Device Manager Flutter Outline Flutter Inspector Flutter Performance Emulator

The screenshot shows the Android Studio interface with a Flutter project named "mi\_card\_flutter". The "main.dart" file is open in the code editor. The code imports the Material package and defines a main function that runs an application with a teal background color. A yellow status bar at the top indicates "Pub get' has not been run". The "lib" folder in the project structure is selected. The bottom right corner of the screen displays several Flutter-related tools: Device Manager, Flutter Outline, Flutter Inspector, Flutter Performance, and Emulator.



three

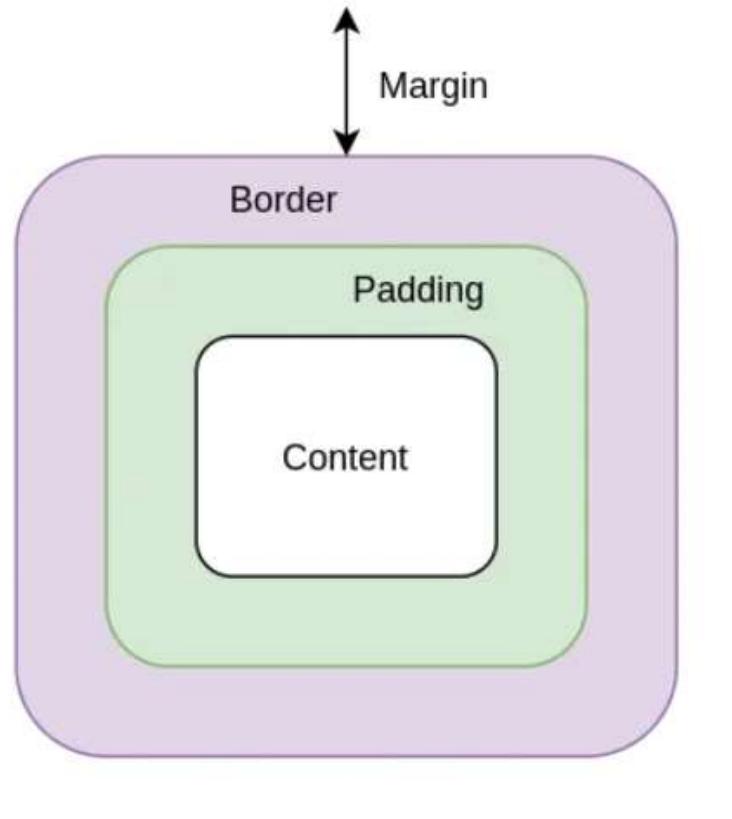
# Mempelajari cara menggunakan layout widgets

Container, Column, Row, dan Card



# Container Widget

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.





# Properti yang digunakan di Container

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

01

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

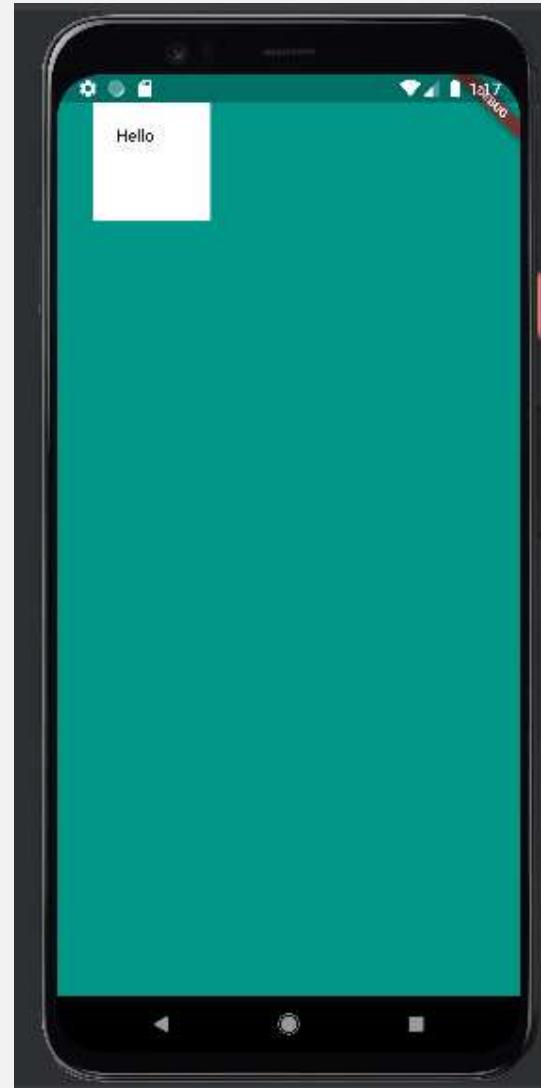
02

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

03

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

04



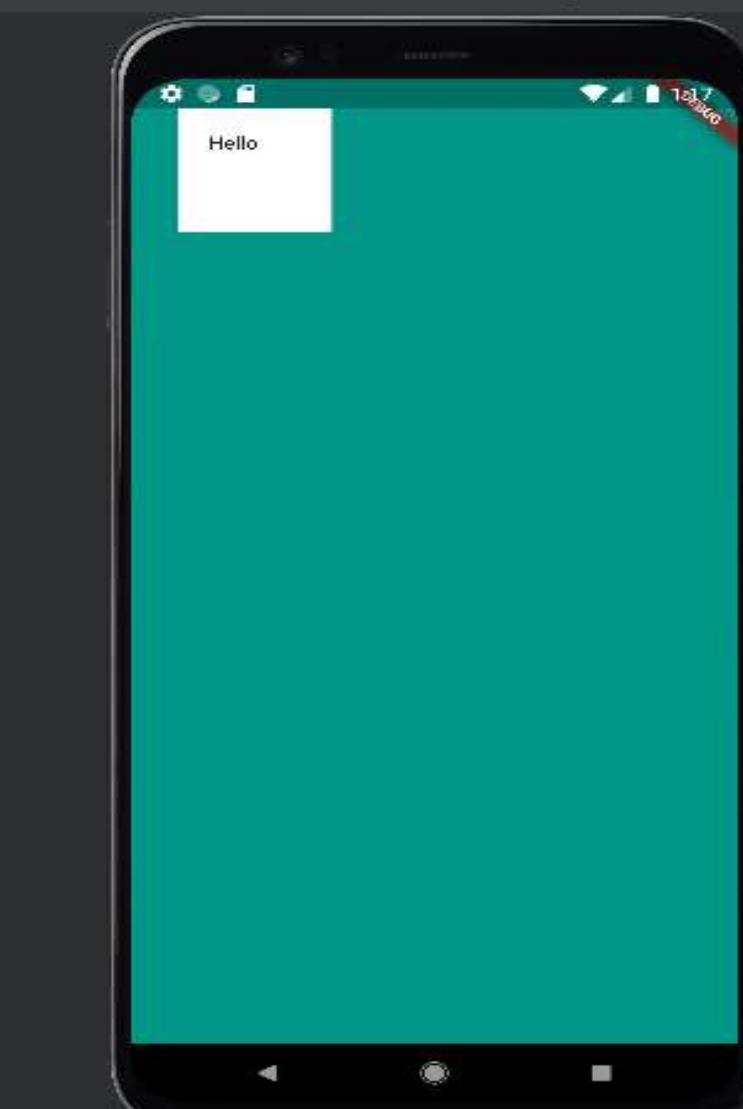
05

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



The image shows a Flutter application running in an Android emulator. The app has a teal background color. In the center, there is a white `Container` widget with a height and width of 100.0. Inside this container, the word "Hello" is displayed as text. The code for this application is visible in the main.dart file.

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10     return MaterialApp(
11       home: Scaffold(
12         backgroundColor: Colors.teal,
13         body: SafeArea(
14           child: Container(
15             height: 100.0,
16             width: 100.0,
17             margin: EdgeInsets.only(left: 30.0),
18             padding: EdgeInsets.all(20.0),
19             color: Colors.white,
20             child: Text('Hello'),
21           ), // Container
22         ), // SafeArea
23       ), // Scaffold
24     ); // MaterialApp
25   }
26 }
27
```



# Column dan Row Widget

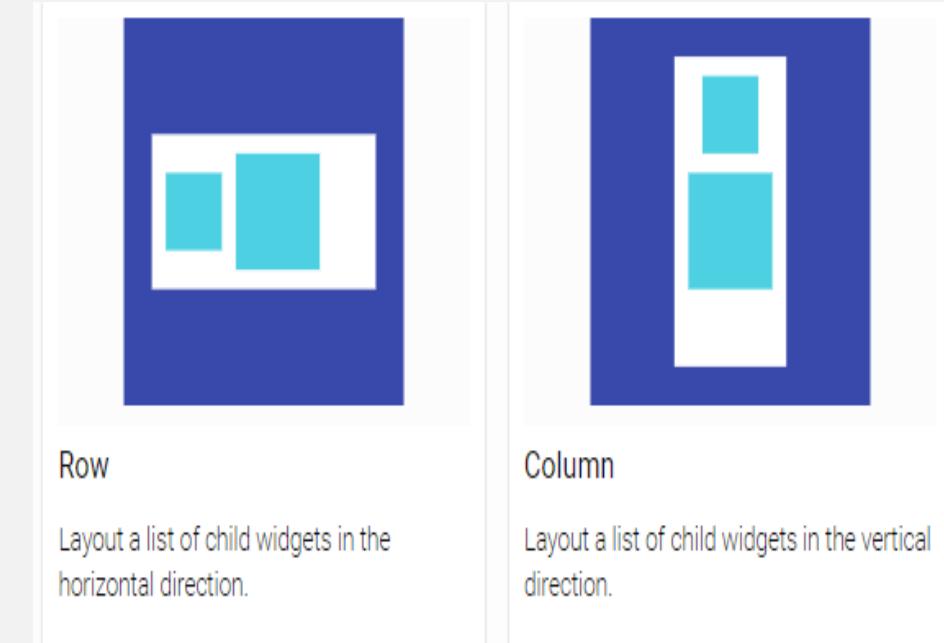
Multi-child layout widgets adalah widget yang memungkinkan penempatan lebih dari satu widget di dalamnya. Widget-widget ini memungkinkan Anda untuk mengatur tata letak dan tampilan dari beberapa widget secara bersamaan.

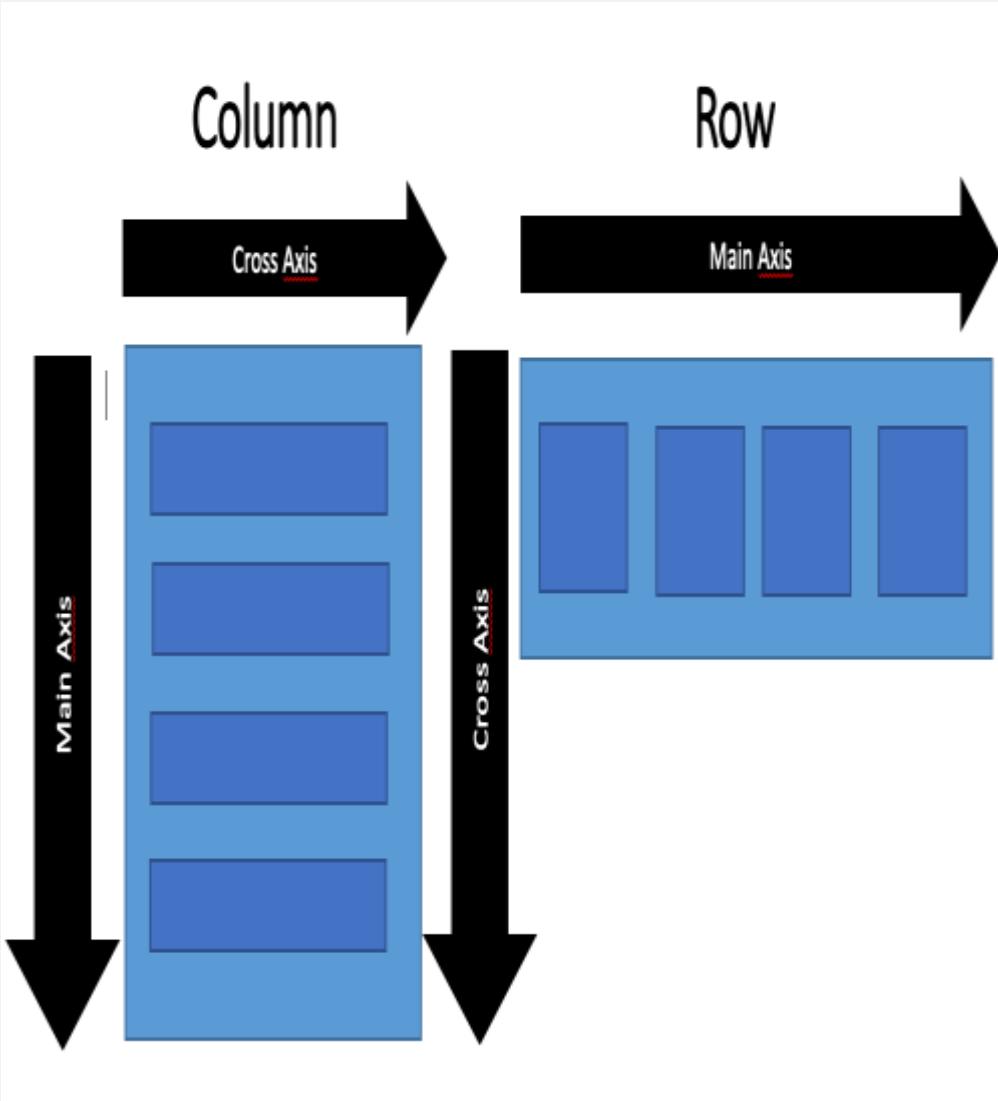
01

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.





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.

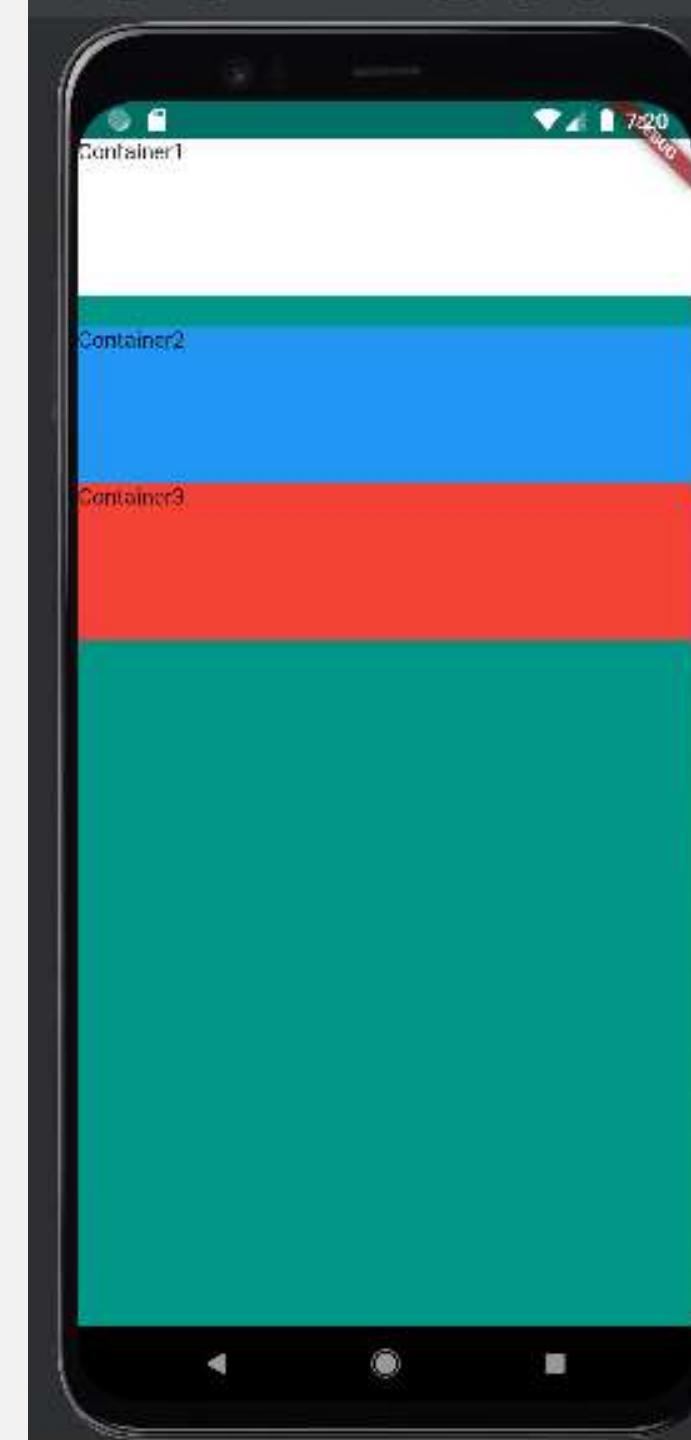


crossAxisAlignment: Properti ini digunakan untuk mengatur tata letak horizontal pada kolom, dan vertical pada baris.



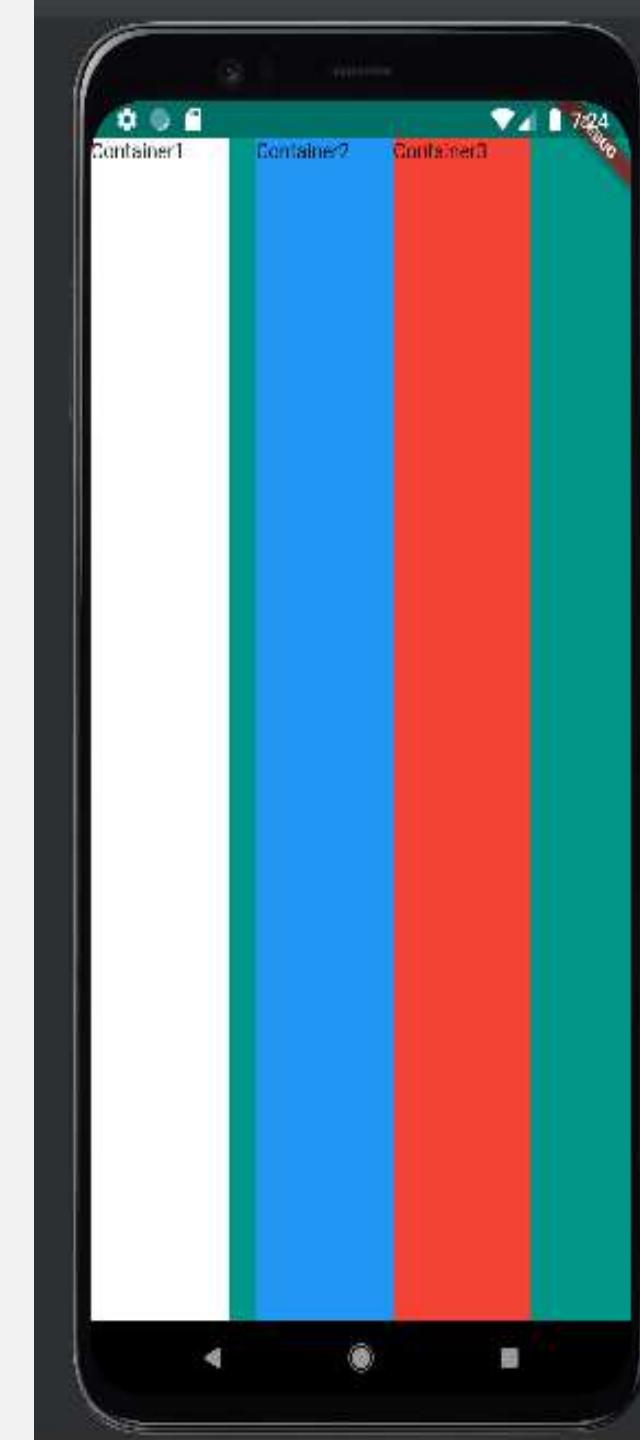
mainAxisSize: Properti ini digunakan untuk mengatur ukuran utama. Jika sebagai MainAxisSize.max, maka akan memperluas ukuran. Jika sebagai MainAxisSize.min, maka akan mengambil ukuran minimum.

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10     return MaterialApp(
11       home: Scaffold(
12         backgroundColor: Colors.teal,
13         body: SafeArea(
14           child: Column(
15             mainAxisAlignment: MainAxisAlignment.spaceEvenly,
16             children: <Widget>[
17               Container(
18                 height: 100.0,
19                 color: Colors.white,
20                 child: Text('Container1'),
21               ), // Container
22               SizedBox(
23                 height: 20.0,
24               ), // SizedBox
25               Container(
26                 height: 100.0,
27                 color: Colors.blue,
28                 child: Text('Container2'),
29               ), // Container
30               Container(
31                 height: 100.0,
32                 color: Colors.red,
33                 child: Text('Container3'),
34               ), // Container
35             ],
36           ), // Column
37         ), // SafeArea
38       ), // Scaffold
39     ); // MaterialApp
40 }
41 }
```



Contoh  
Penggunaan  
Kode Pada  
Column

```
README.md × main.dart ×  
1 import 'package:flutter/material.dart';  
2  
3 void main() {  
4   runApp(MyApp());  
5 }  
6  
7 class MyApp extends StatelessWidget {  
8   @override  
9   Widget build(BuildContext context) {  
10     return MaterialApp(  
11       home: Scaffold(  
12         backgroundColor: Colors.teal,  
13         body: SafeArea(  
14           child: Row(  
15             mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
16             children: <Widget>[  
17               Container(  
18                 width: 100.0,  
19                 color: Colors.white,  
20                 child: Text('Container1'),  
21               ), // Container  
22               SizedBox(  
23                 width: 20.0,  
24               ), // SizedBox  
25               Container(  
26                 width: 100.0,  
27                 color: Colors.blue,  
28                 child: Text('Container2'),  
29               ), // Container  
30               Container(  
31                 width: 100.0,  
32                 color: Colors.red,  
33                 child: Text('Container3'),  
34               ), // Container  
35             ], // <Widget>[]  
36           ), // Row  
37         ), // SafeArea  
38       ), // Scaffold  
39     ); // MaterialApp  
40   }  
41 }
```



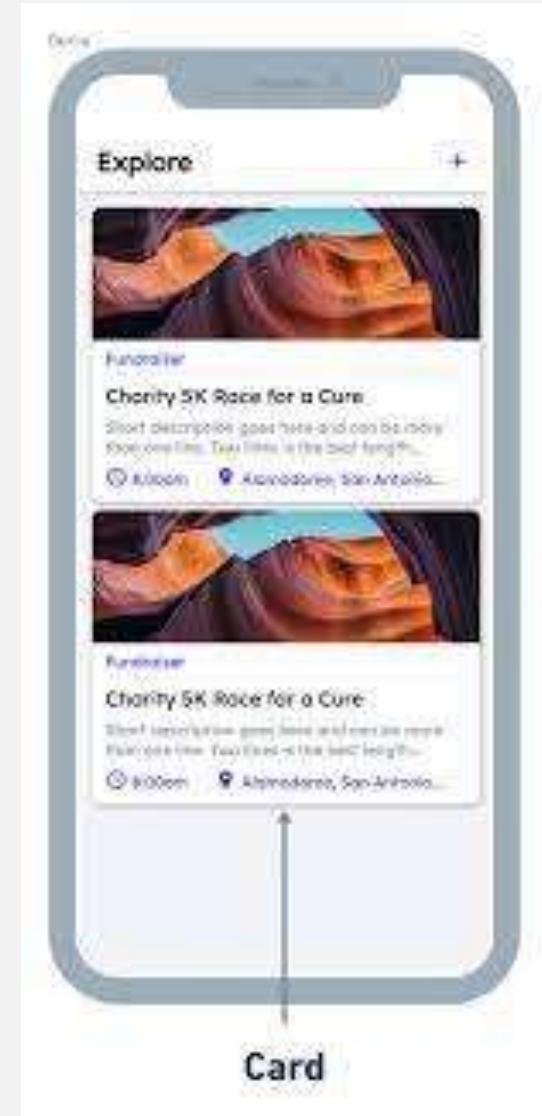
# Contoh Penggunaan Kode Pada Row



# Card Widget

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

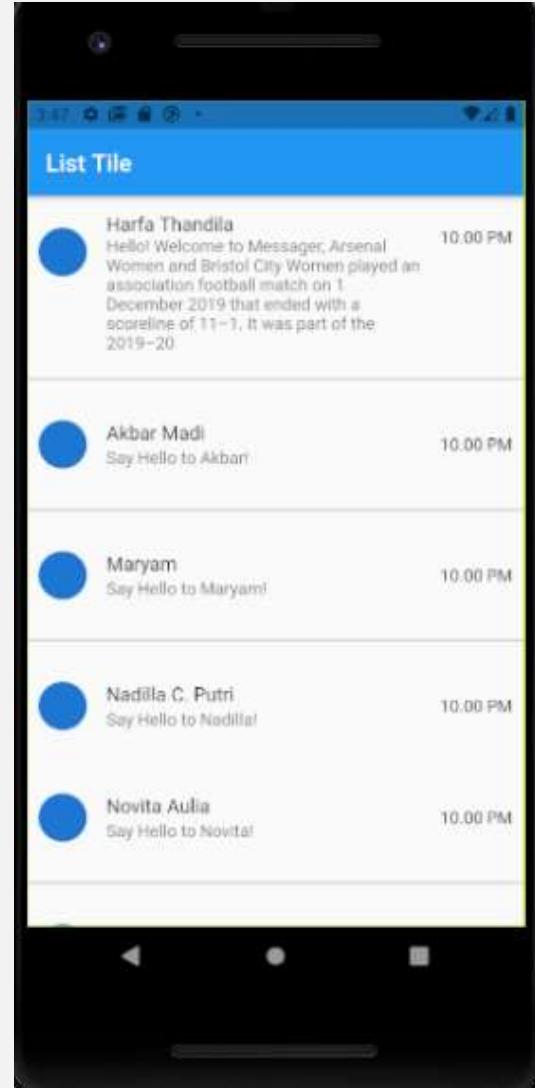
Widget Card biasanya digunakan untuk mengatur konten dalam tata letak. Secara sederhana, sebuah card terdiri dari beberapa elemen, seperti latar belakang, judul, gambar, teks, dan sebagainya.





# 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.



# Contoh Penggunaan Kode Pada Card & ListTile



A screenshot of a code editor showing a Dart file named `main.dart`. The code defines a `Card` widget containing a `ListTile`. The `ListTile` has a leading icon of a phone and a title text '+' followed by a 10-digit phone number. The code uses `Color` objects from the `Colors` class and `EdgeInsets` from the `EdgeInsets` class. The code editor interface shows tabs for `README.md`, `main.dart`, and `pubspec.yaml`. On the left, there are vertical icons for file navigation and a search bar.

```
39   Card(
40     color: Colors.white,
41     margin: EdgeInsets.symmetric(vertical: 10.0, horizontal: 25.0),
42     child: ListTile(
43       leading: Icon(
44         Icons.phone,
45         color: Colors.teal,
46       ), // Icon
47       title: Text(
48         '+62 123 456 789',
49         style: TextStyle(
50           color: Colors.teal.shade900,
51           fontFamily: 'Source Sans Pro',
52           fontSize: 20.0,
53         ), // TextStyle
54       ), // Text
55     ), // ListTile, Card
```



# Properti yang digunakan di Card

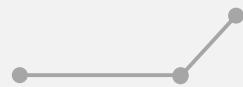
**Child:** untuk menentukan widget yang akan menjadi konten utama dari card.



**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

```
31 }  
32 }  
33 }  
34 }  
  
class MyHomePage extends StatefulWidget {
```

Run: main.dart

Console:

```
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.

```
31 }  
32 }  
33 }  
34 }
```

Run: main.dart

Console:

```
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

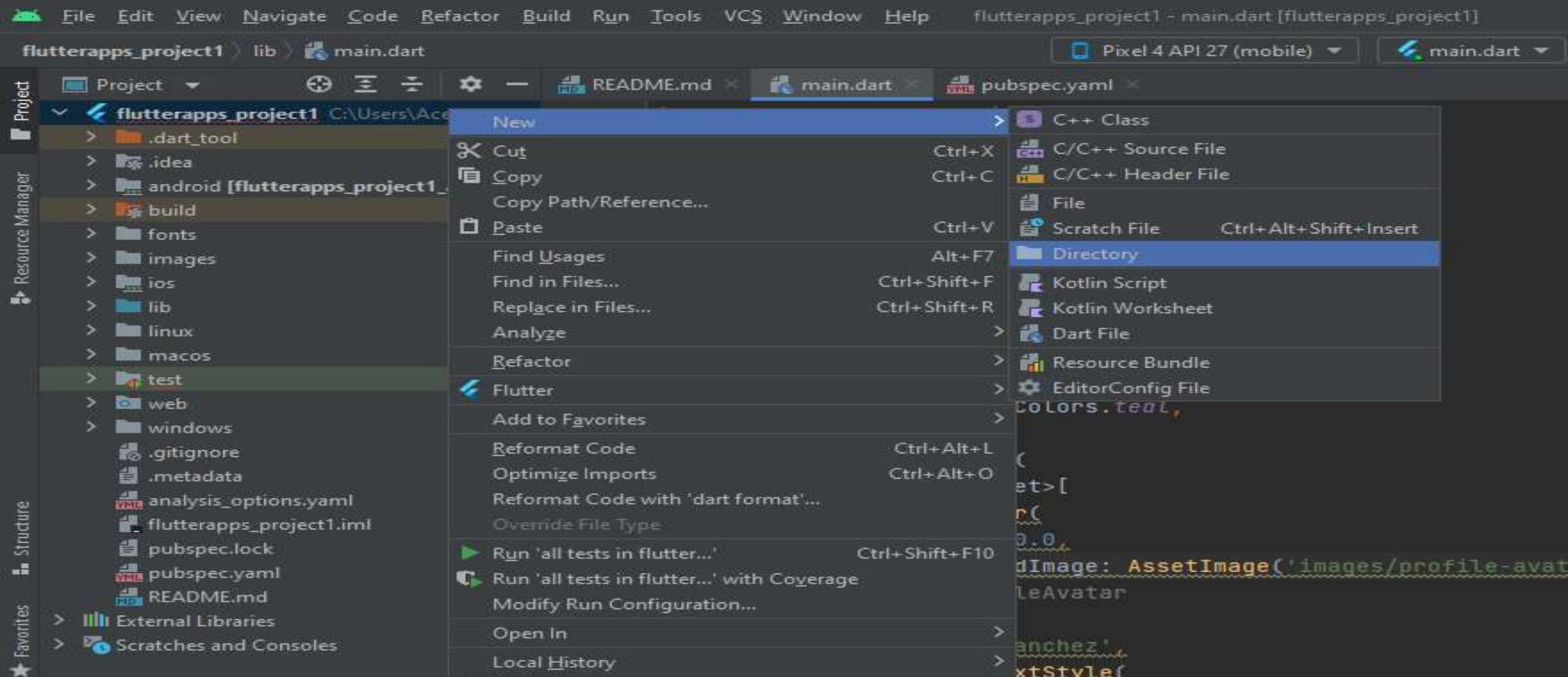
## Mempelajari cara menggunakan file pubspec.yaml

Memasukkan Asset Image dan Fonts

four

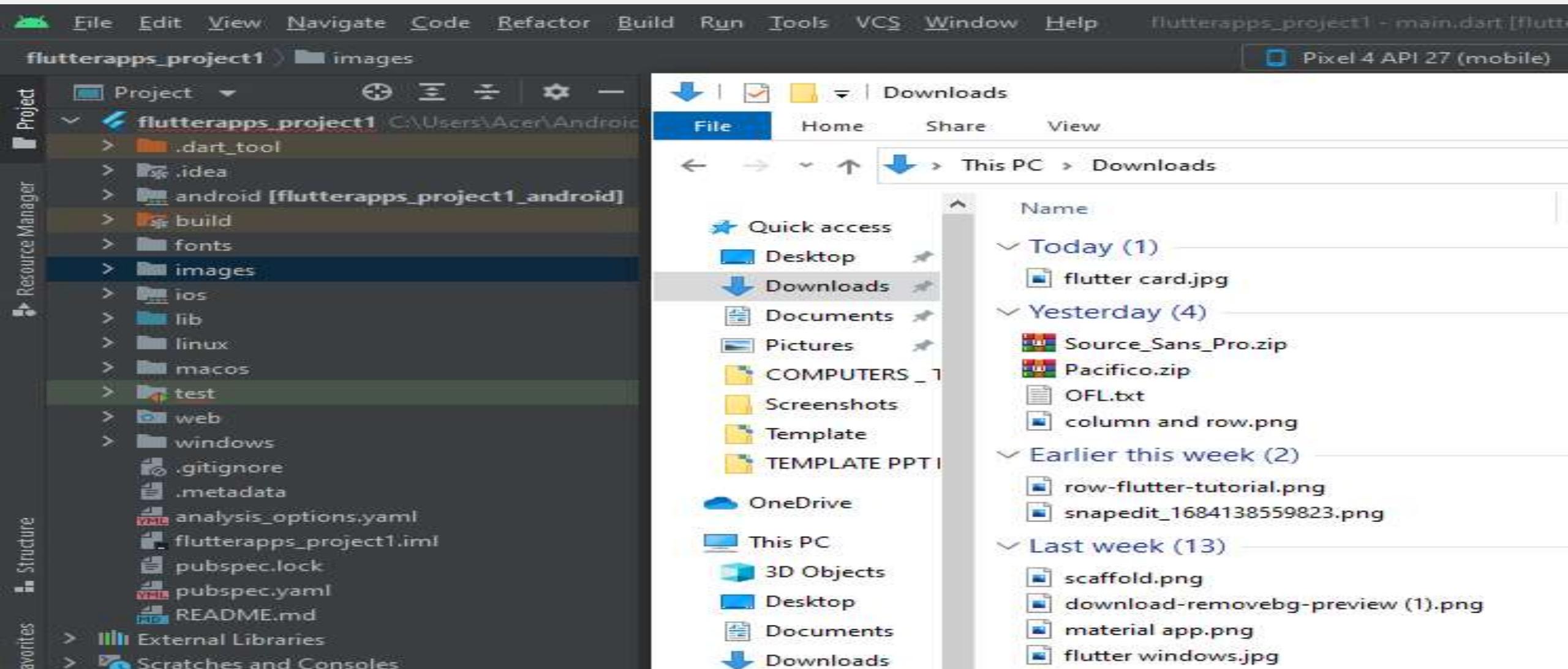


# Memasukkan Asset Image



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

four



Pilih foto pada document → lalu tarik foto kedalam folder images

four



## Memasukkan Asset Image Pada File Pubspec.yaml



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

The screenshot shows the Android Studio interface with the project 'flutterapps\_project1' open. The pubspec.yaml file is selected in the code editor. The code in the file is as follows:

```
11:   sdk: flutter
12:
13:   cupertino_icons: ^1.0.2
14:
15:   dev_dependencies:
16:     flutter_test:
17:       sdk: flutter
18:
19:     flutter_lints: ^2.0.0
20:
21:   flutter:
22:     uses-material-design: true
23:
24:   assets:
25:     - images/profile-avatar.png
26:
27:   fonts:
28:     - family: Pacifico
29:       fonts:
30:         - asset: fonts/Pacifico-Regular.ttf
31:
32:     - family: Source Sans Pro
33:       fonts:
34:         - asset: fonts/SourceSansPro-Regular.ttf
35:
```

The 'assets' section contains a single entry: '- images/profile-avatar.png'. The 'fonts' section contains two entries, each with a font family and a corresponding TTF file asset.

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



Google Fonts

Fonts

Icons

Knowledge

FAQ



Pacifico

Sentence ▾ Type something

40px ▾



Categories ▾

Language ▾

Number of styles ▾

 Show only variable fonts  Show only color fonts 

1513 of 1513 families

About these results

Sort by: Trending ▾

Roboto  
Christian Robertson

12 styles

Whereas recognition  
of the inherent dignity

Noto Sans Nandinagari  
Google

1 style

যত্র ছাগতি  
রা°তিন্নাযস্বাত°আাা°  
আাাৰঃ মানবপনিবানৰা  
সাৰ্বেষামপি

Noto Sans Nag Mundari  
Google

Variable (1 axis)

ଧିଦେବ ହିଦେବାଳା ଲେ  
ଶକ୍ତିବ୍ୟକ୍ତି ଶକ୍ତି  
କୁରୁକୁରୁକୁରୁକୁରୁକୁରୁକୁରୁ  
ମନୁଷୁକୁରୁକୁରୁକୁରୁକୁରୁକୁରୁ  
ଯକୁନିକୁ

four



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

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

Whereas a common

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

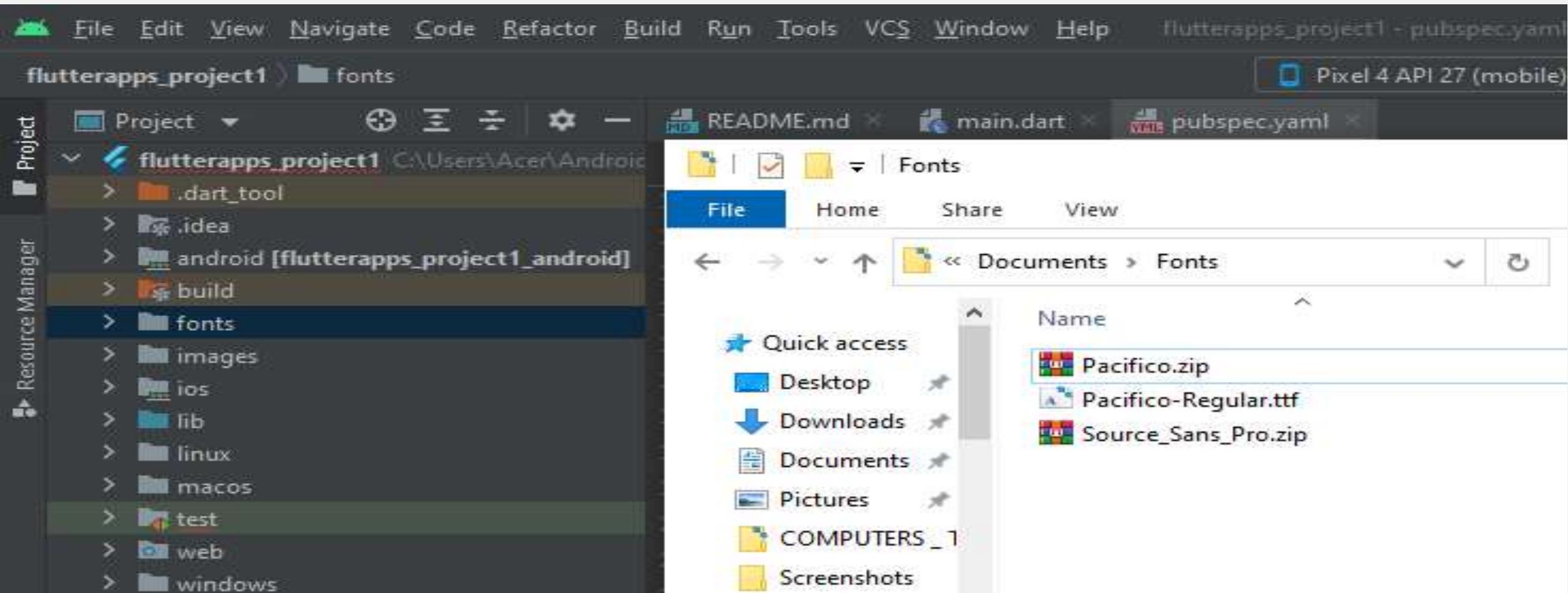
<link>  @import

```
<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



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

four



# Memasukkan Fonts Pada File Pubspec.yaml

```
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

four



# Memasukkan Materials Icons



Google Fonts

Material Icons Category Search Material Icons

Outlined Filled Rounded Sharp Two tone

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.

Design guide Figma file GitHub repo Apache

Action

Search Home Account Circle Settings Done Info Check Circle Delete Visibility Shopping Cart

Icons Knowledge

Web Android iOS Flutter

Use in Flutter

Check out the [Flutter API docs](#) to customize icon size and color.

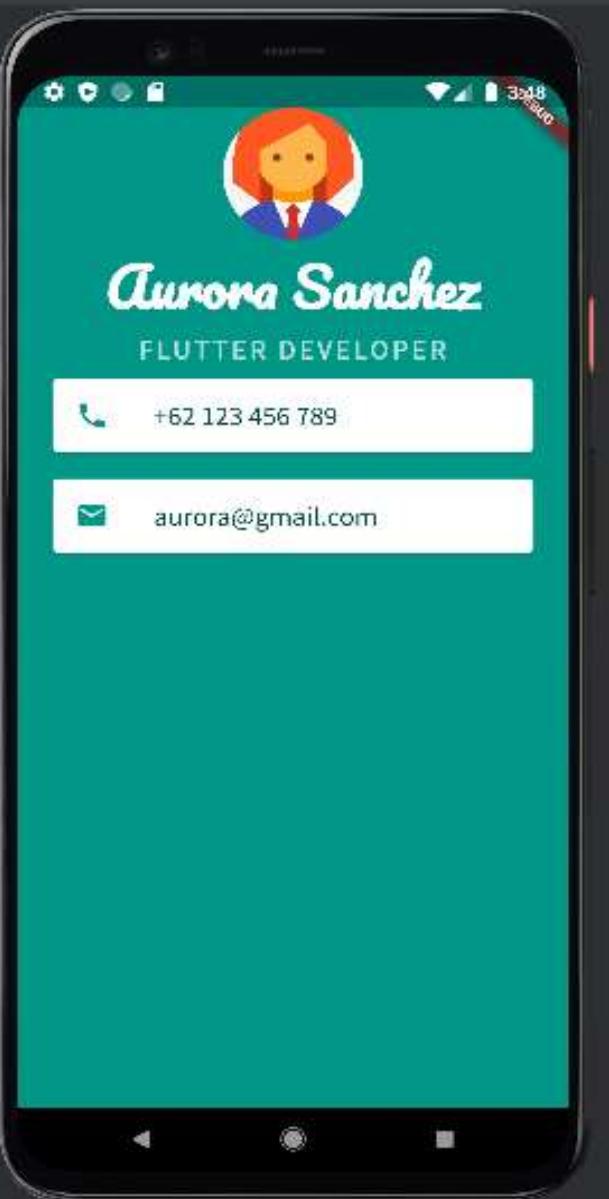
```
Icon(  
  Icons.search_outlined,  
)
```

Flutter ID

```
search_outlined
```

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

# Hasil User Interface



```
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**

five



# Memasukkan App Icon



App Icon Generator

App Icon

Image Sets

Donate



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

five



# Memasukkan App Icon



App Icon Generator

App Icon

Image Sets

Donate



## 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

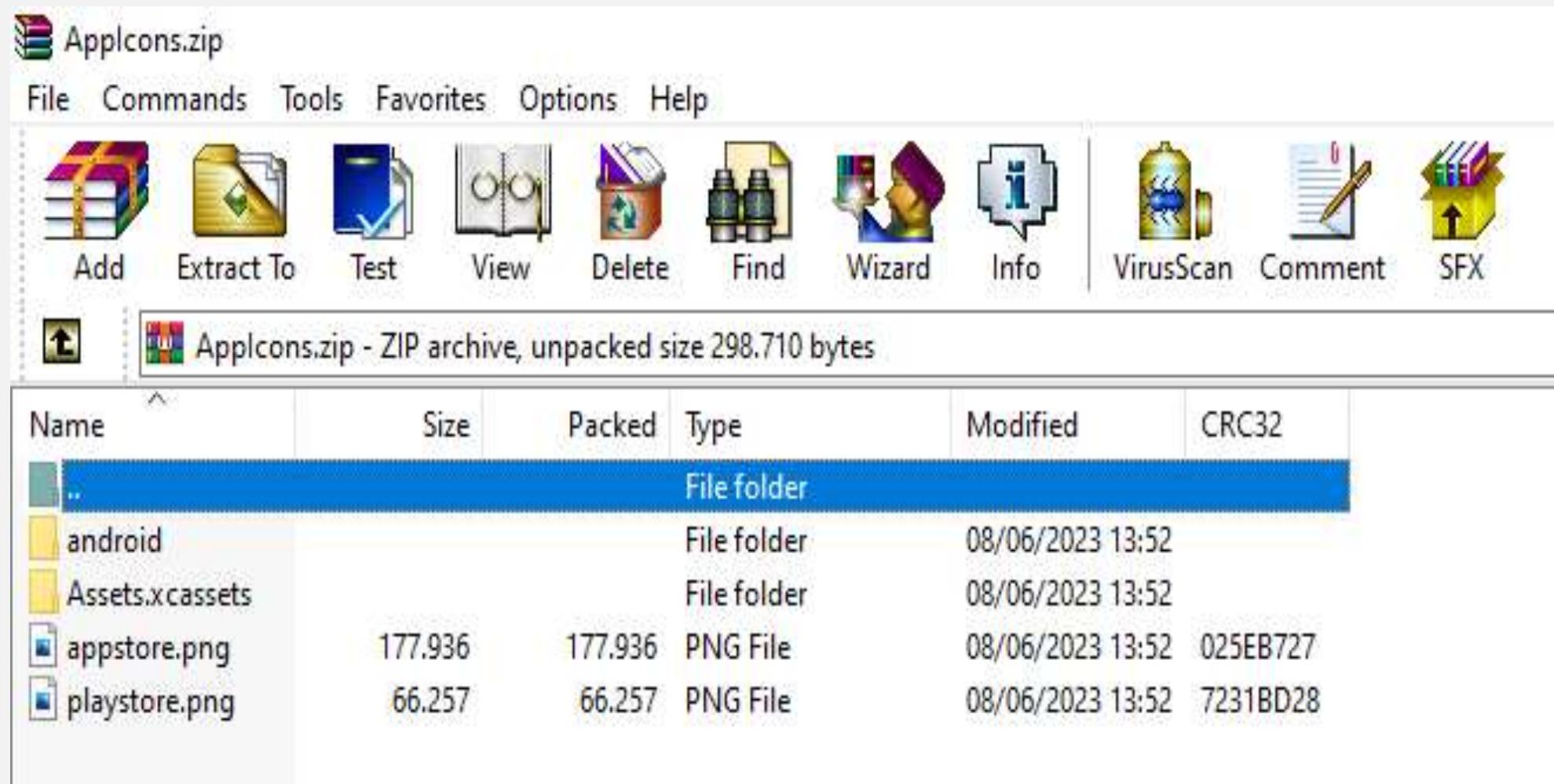
Change file name for all generated Android images



Generate



# Memasukkan App Icon

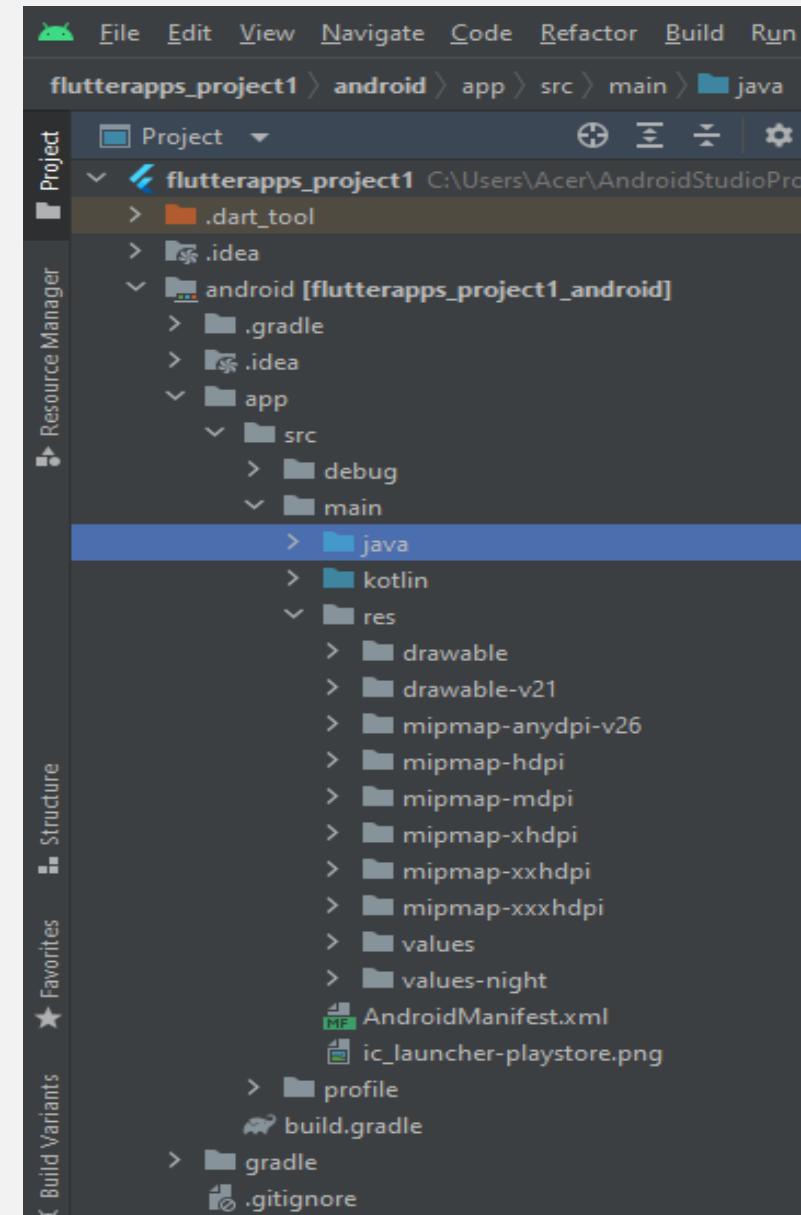


Buka file zip yang telah di download → Pilih Android → Ekstrak file



# Memasukkan App Icon

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



Masukkan folder ke  
dalam file res



# Memasukkan App Icon

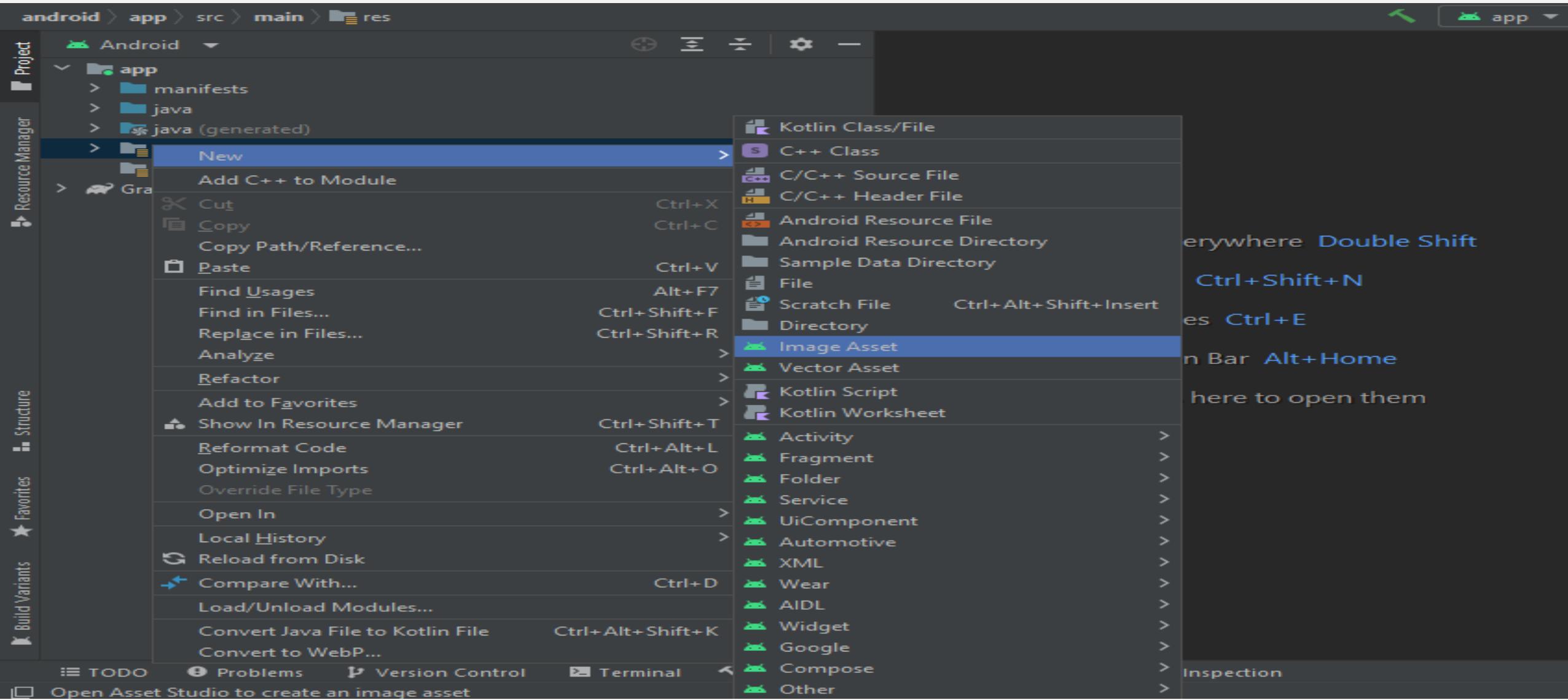
The screenshot shows the Android Studio interface with the project 'flutterapps\_project1' selected. In the center, the 'pubspec.yaml' file is open, displaying configuration for fonts and images. A context menu is open over the 'android' folder in the Project tree, with the 'Flutter' option highlighted. The 'Flutter' option has a tooltip explaining it opens the Android module in Android Studio. The code in the pubspec.yaml file includes sections for fonts and images, with examples of asset paths like 'assets/fonts/Pacifico-Regular.ttf'.

```
flutterapps_project1 - pubspec.yaml [flutterapps_project1]
flutterapps_project1 > android > app
Project README.md main.dart pubspec.yaml
flutterapps_project1 .dart_tool .idea android [flutterapps_project1_android] .gradle app New Cut Copy Copy Path/Reference... Paste Find Usages Find in Files... Replace in Files... Analyze Refactor Add to Favorites Reformat Code Optimize Imports Delete... Override File Type Open In Local History Reload from Disk Compare With... Mark Directory as
Flutter commands
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
  https://flutter.dev/assets-and-images/#resolution-aware
  Open Android module in Android Studio
  https://flutter.dev/assets-and-images/#from-packages
  add custom fonts to your application, add a fonts section
  to this "flutter" section. Each entry in this list should have
  a "family" key with the font family name, and a "fonts" key with
  a list giving the asset and other descriptors for the font. For
  example:
  fonts:
    - 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 %

Preview

xhdpi  Show safe zone  Show grid

Circle      Squircle      Rounded Square      Square

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

Pilih foto yang akan digunakan

Previous → Next Cancel Finish

five



# 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: tterapps\_project1\images\flutter.jpg

Scaling

Trim:  Yes  No

Resize:  81 %

Preview

xhdpi  Show safe zone  Show grid

Circle      Squircle      Rounded Square      Square

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

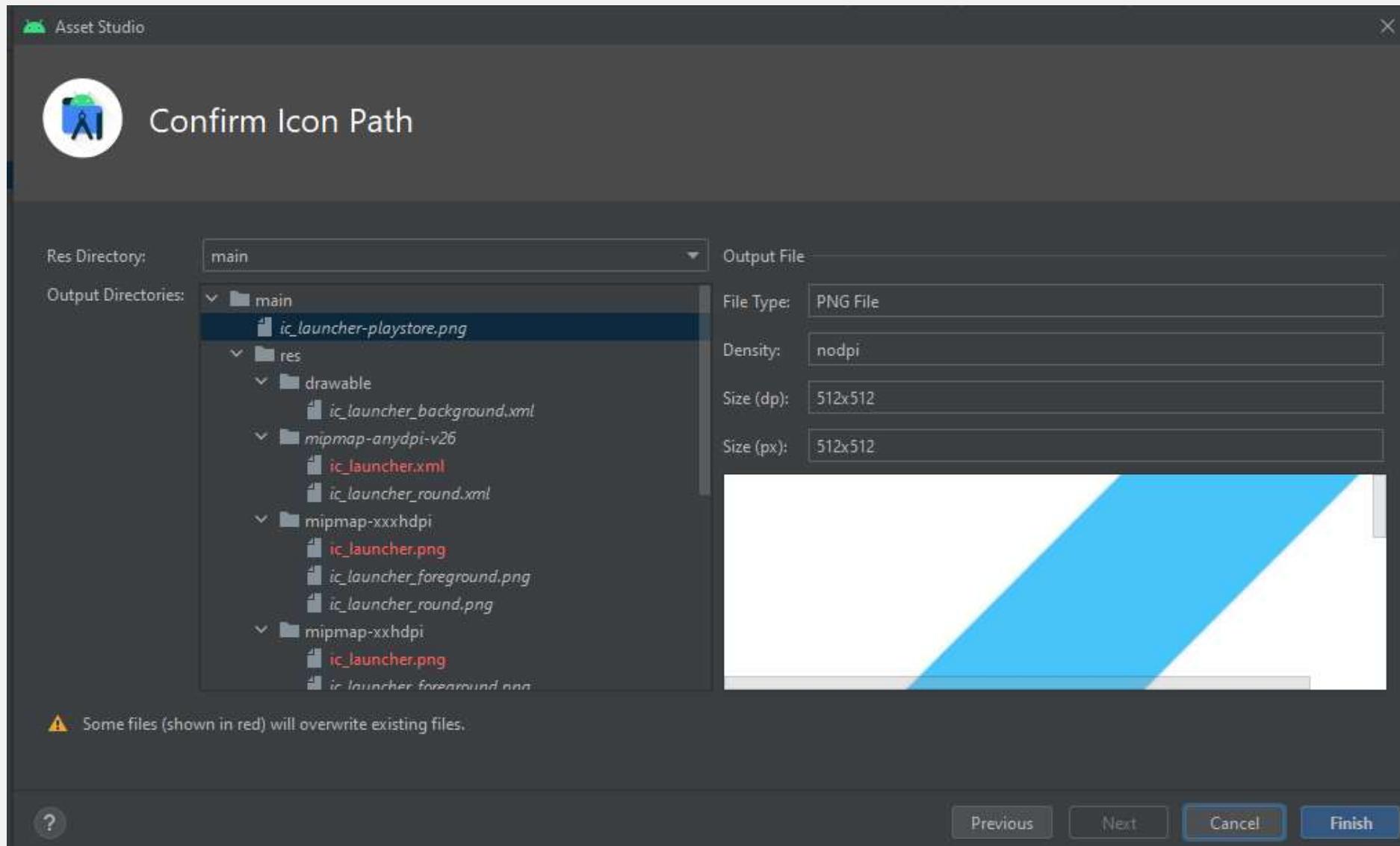
Atur resize dan klik next

Previous Next Cancel Finish

five



# Memasukkan App Icon



Klik finish → lalu buka di emulator

FLUTTER



**TERIMA KASIH**