

# **LAPORAN JOBSHEET 6**

## **Layout & Navigasi**

Mata Kuliah: Pemrograman Mobile

Dosen Pengampu: Ade Ismail, S.Kom., M.TI.



Disusun Oleh:

Nama: Syava Aprilia P

NIM: 2241760129

Absen: 24

**PROGRAM STUDI SISTEM INFORMASI BISNIS**

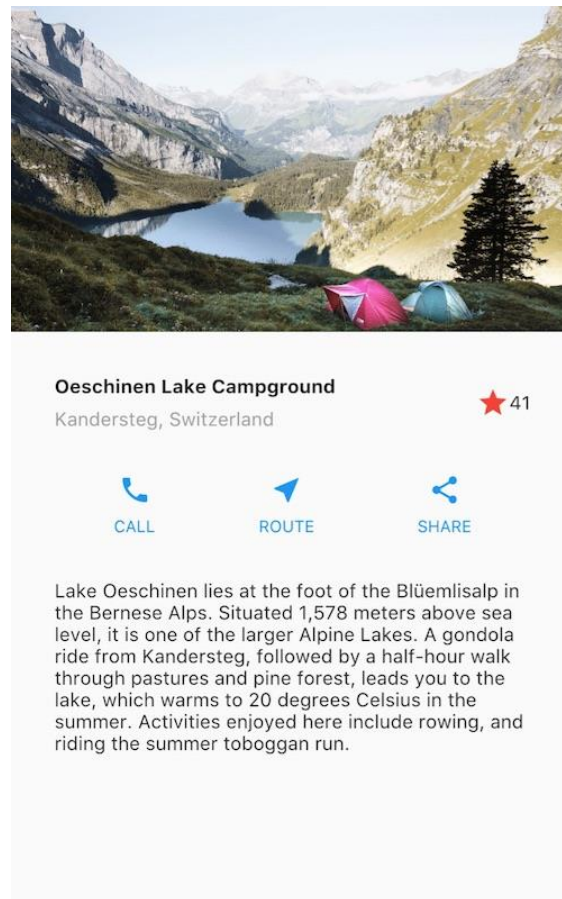
**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2024**

# Praktikum 1: Membangun Layout di Flutter

Tampilan akhir yang akan Anda buat



## Langkah 1: Buat Project Baru

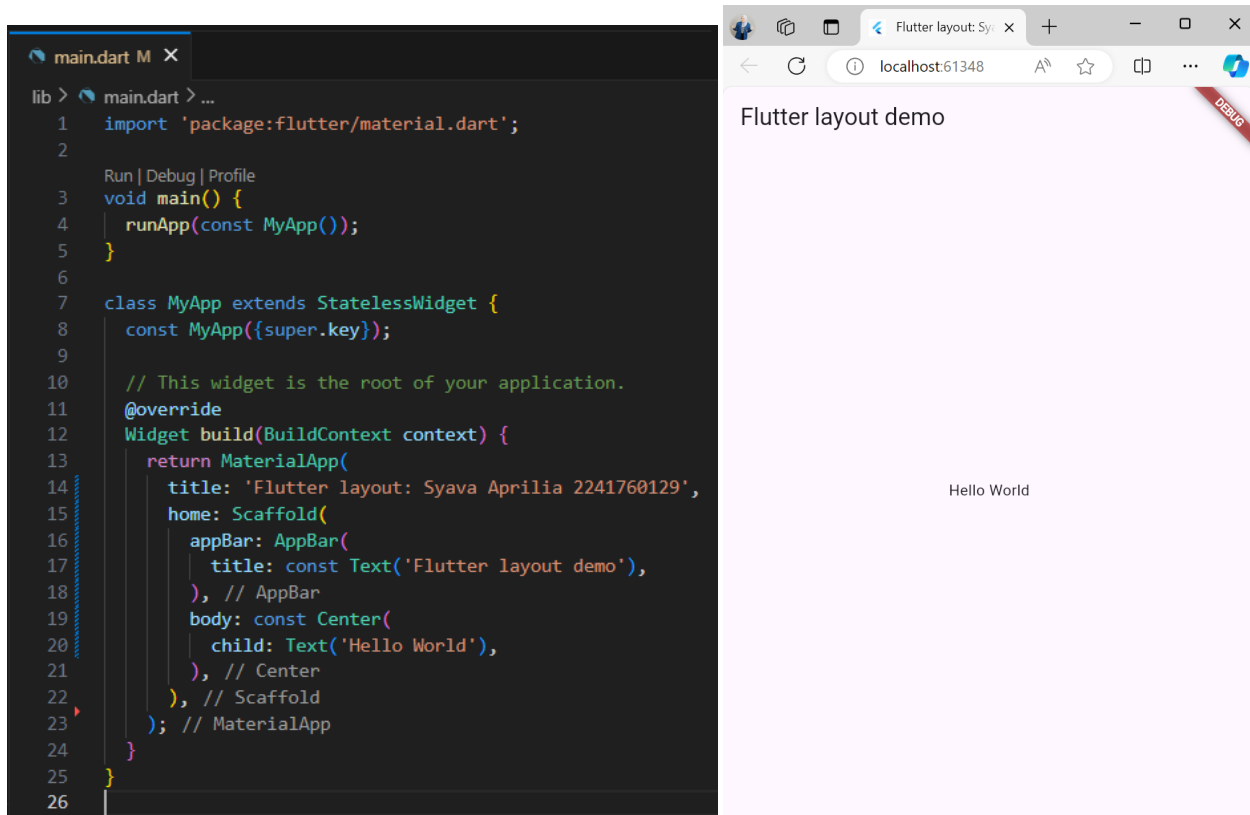
Buatlah sebuah project flutter baru dengan nama **layout\_flutter**. Atau sesuaikan style laporan praktikum yang Anda buat.

```
EXPLORER  ...  main.dart U X
/ LAYOUT_FLUTTER
> .dart_tool
> .idea
> android
> ios
> lib
  main.dart U
> linux
> macos
> test
> web
> windows
> .gitignore
> .metadata
! analysis_options... U
! layout_flutter.iml
! pubspec.lock
! pubspec.yaml
! README.md U

lib > main.dart > ...
1  import 'package:flutter/material.dart';
2
3  Run | Debug | Profile
4  void main() {
5    runApp(const MyApp());
6  }
7
8  class MyApp extends StatelessWidget {
9    const MyApp({super.key});
10
11    // This widget is the root of your application.
12    @override
13    Widget build(BuildContext context) {
14      return MaterialApp(
15        title: 'Flutter Demo',
16        theme: ThemeData(
17          // This is the theme of your application.
18          // TRY THIS: Try running your application with "flutter run". Y
19          // the application has a purple toolbar. Then, without quitting
20          // try changing the seedColor in the colorScheme below to Color
21          // and then invoke "Hot reload" from your IDE or press the
```

## Langkah 2: Buka file lib/main.dart

Buka file main.dart lalu ganti dengan kode berikut. Isi nama dan NIM Anda di text title.

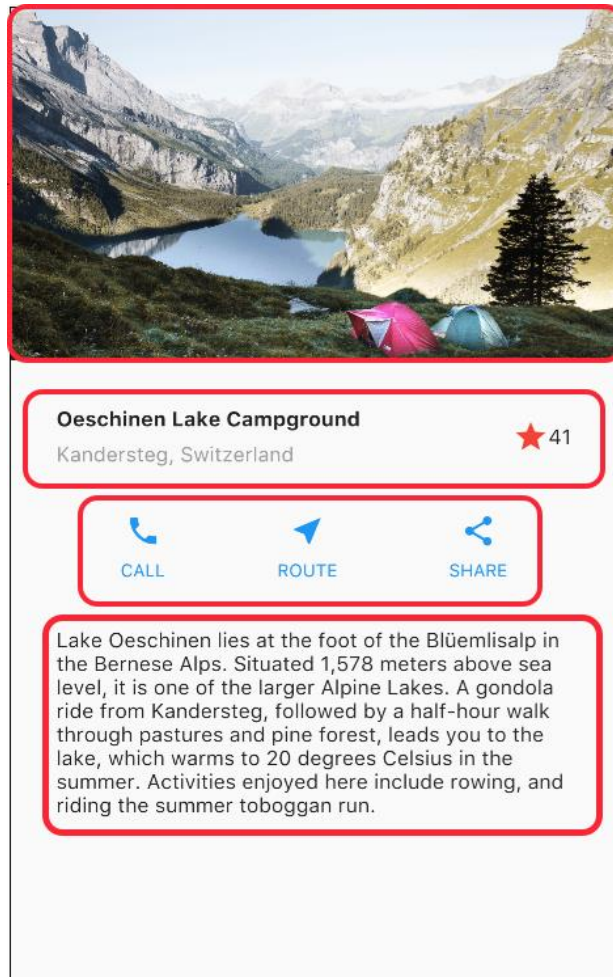


## Langkah 3: Identifikasi layout diagram

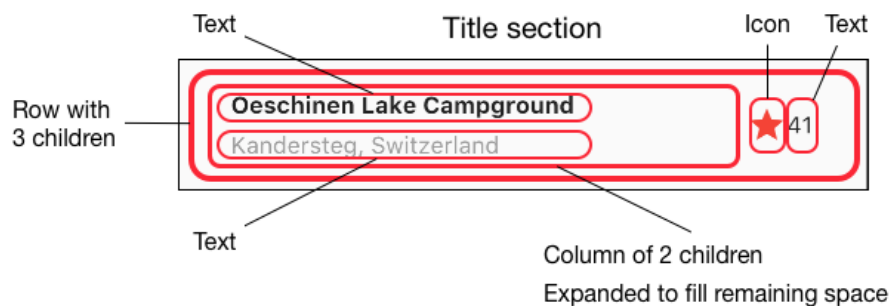
Langkah pertama adalah memecah tata letak menjadi elemen dasarnya:

- Identifikasi baris dan kolom.
- Apakah tata letaknya menyertakan kisi-kisi (grid)?
- Apakah ada elemen yang tumpang tindih?
- Apakah UI memerlukan tab?
- Perhatikan area yang memerlukan alignment, padding, atau borders.

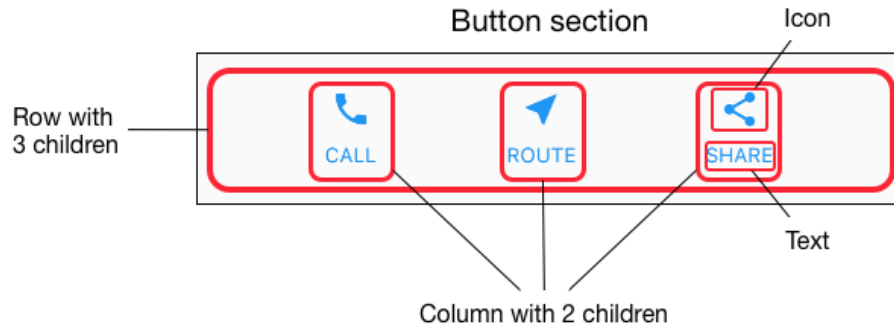
Pertama, identifikasi elemen yang lebih besar. Dalam contoh ini, empat elemen disusun menjadi sebuah kolom: sebuah gambar, dua baris, dan satu blok teks.



Selanjutnya, buat diagram setiap baris. Baris pertama, disebut bagian Judul, memiliki 3 anak: kolom teks, ikon bintang, dan angka. Anak pertamanya, kolom, berisi 2 baris teks. Kolom pertama itu memakan banyak ruang, sehingga harus dibungkus dengan widget yang Diperluas.



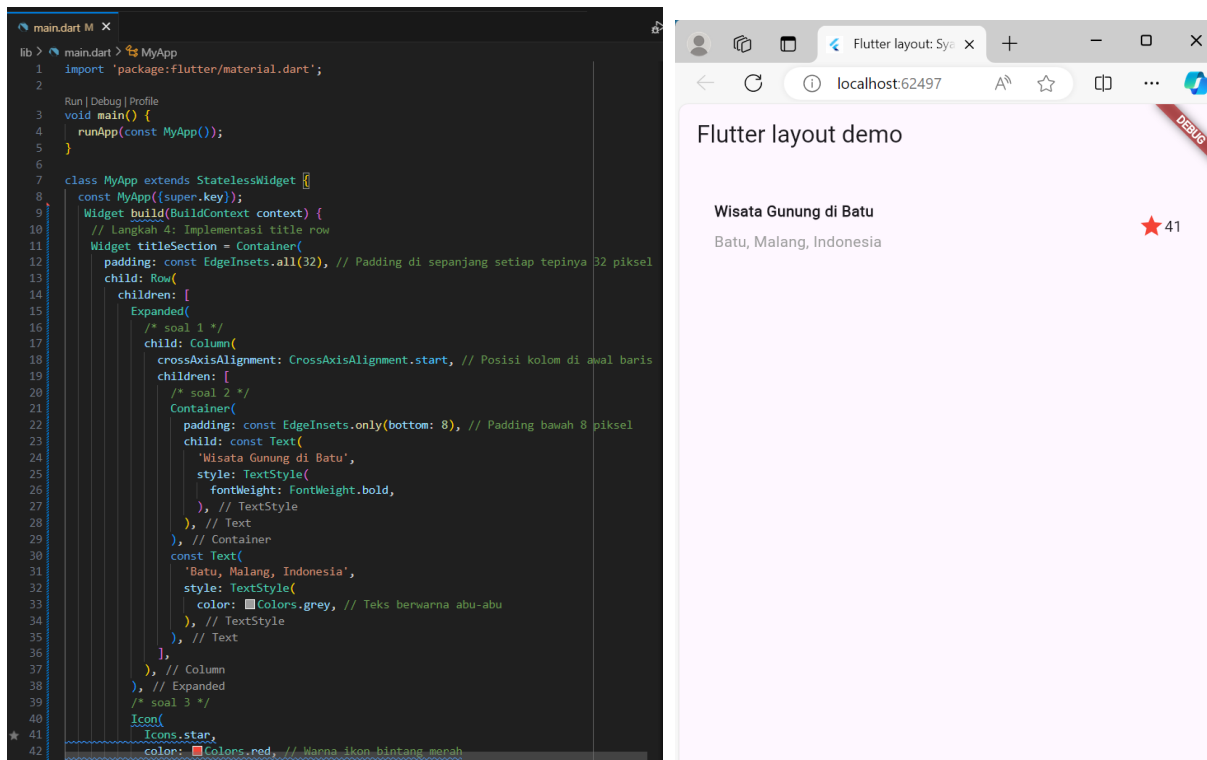
Baris kedua, disebut bagian Tombol, juga memiliki 3 anak: setiap anak merupakan kolom yang berisi ikon dan teks.



Setelah tata letak telah dibuat diagramnya, cara termudah adalah dengan menerapkan pendekatan bottom-up. Untuk meminimalkan kebingungan visual dari kode tata letak yang banyak bertumpuk, tempatkan beberapa implementasi dalam variabel dan fungsi.

#### Langkah 4: Implementasi title row

Pertama, Anda akan membuat kolom bagian kiri pada judul. Tambahkan kode berikut di bagian atas metode build() di dalam kelas MyApp:



## Praktikum 2: Implementasi button row

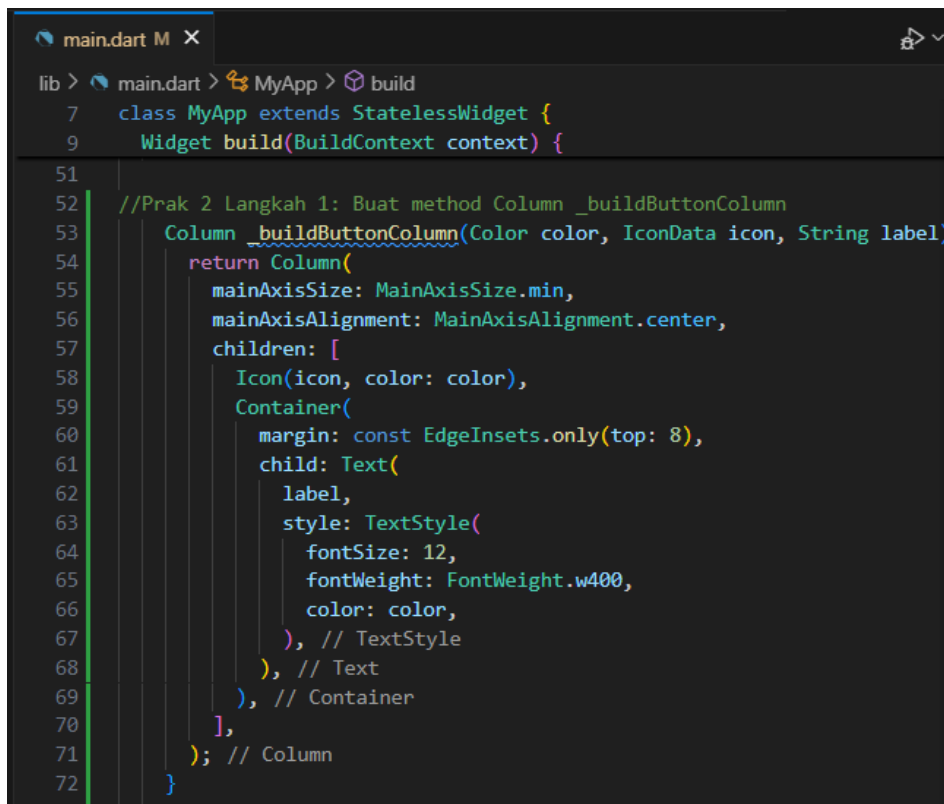
Selesaikan langkah-langkah praktikum berikut ini dengan melanjutkan dari praktikum sebelumnya.

#### Langkah 1: Buat method `Column _buildButtonColumn`

Bagian tombol berisi 3 kolom yang menggunakan tata letak yang sama—sebuah ikon di atas baris teks. Kolom pada baris ini diberi jarak yang sama, dan teks serta ikon diberi warna primer.

Karena kode untuk membangun setiap kolom hampir sama, buatlah metode pembantu pribadi bernama `buildButtonColumn()`, yang mempunyai parameter warna, Icon dan Text, sehingga dapat mengembalikan kolom dengan widgetnya sesuai dengan warna tertentu.

#### lib/main.dart (`_buildButtonColumn`)



```
lib > main.dart > MyApp > build
7  class MyApp extends StatelessWidget {
9    Widget build(BuildContext context) {

51
52    //Prak 2 Langkah 1: Buat method Column _buildButtonColumn
53    Column _buildButtonColumn(Color color, IconData icon, String label)
54    {
55      return Column(
56        mainAxisAlignment: MainAxisAlignment.min,
57        mainAxisAlignment: MainAxisAlignment.center,
58        children: [
59          Icon(icon, color: color),
60          Container(
61            margin: const EdgeInsets.only(top: 8),
62            child: Text(
63              label,
64              style: TextStyle(
65                fontSize: 12,
66                fontWeight: FontWeight.w400,
67                color: color,
68              ), // TextStyle
69            ), // Text
70          ), // Container
71        ],
72      ); // Column
73    }
74  }
```

#### Langkah 2: Buat widget `buttonSection`

Buat Fungsi untuk menambahkan ikon langsung ke kolom. Teks berada di dalam Container dengan margin hanya di bagian atas, yang memisahkan teks dari ikon.

Bangun baris yang berisi kolom-kolom ini dengan memanggil fungsi dan set warna, Icon, dan teks khusus melalui parameter ke kolom tersebut. Sejajarkan kolom di sepanjang sumbu utama menggunakan `MainAxisAlignment.spaceEvenly` untuk mengatur ruang kosong secara merata sebelum, di antara, dan setelah setiap kolom. Tambahkan kode berikut tepat di bawah deklarasi `titleSection` di dalam metode `build()`:

#### lib/main.dart (`buttonSection`)

#### Langkah 3: Tambah `button section` ke `body`

Tambahkan variabel `buttonSection` ke dalam `body` seperti berikut:

```

main.dart M X
lib > main.dart > MyApp > build
7   class MyApp extends StatelessWidget {
9     Widget build(BuildContext context) {
53       Column _buildButtonColumn(Color color, IconData icon, String label
71     ); // Column
72   }
73
74   //Prak 2 langkah 2: Buat widget buttonSection
75   Color color = Theme.of(context).primaryColor;
76
77   Widget buttonSection = Row(
78     mainAxisAlignment: MainAxisAlignment.spaceEvenly,
79     children: [
80       _buildButtonColumn(color, Icons.call, 'CALL'),
81       _buildButtonColumn(color, Icons.near_me, 'ROUTE'),
82       _buildButtonColumn(color, Icons.share, 'SHARE'),
83     ],
84   );
85
86   return MaterialApp(
87     title: 'Flutter layout: Syava Aprilia - 2241760129', // Nama dan
88     home: Scaffold(

```

## Praktikum 3: Implementasi text section

Selesaikan langkah-langkah praktikum berikut ini dengan melanjutkan dari praktikum sebelumnya.

### Langkah 1: Buat widget textSection

Tentukan bagian teks sebagai variabel. Masukkan teks ke dalam Container dan tambahkan padding di sepanjang setiap tepinya. Tambahkan kode berikut tepat di bawah deklarasi buttonSection:

```

lib > main.dart > MyApp > build
7   class MyApp extends StatelessWidget {
9     Widget build(BuildContext context) {
86   //Prak 3 langkah 1: Buat widget textSection
87   Widget textSection = Container(
88     padding: const EdgeInsets.all(32),
89     child: const Text(
90       'Pura Luhur Uluwatu terletak di Desa Pecatu, Kuta Selatan, Badung, Bali. '
91       'Pura ini berdiri di atas tebing setinggi 97 meter yang menjorok ke laut dan merupakan salah satu Pura Sad Kayangan, dipercaya sebagai penyangga 9 mata angin oleh umat Hindu. '
92       'Awalnya, pura ini digunakan untuk memuja Empu Kuturan, seorang pendeta suci abad ke-11, dan kemudian Dang Hyang Nirartha yang mencapai moksa di sini, yang menginspirasi nama "Luhur Uluwatu." '
93       'Selain menjadi tempat ibadah, Pura Uluwatu juga terkenal dengan Pantai Pecatu di bawahnya, yang menjadi destinasi selancar internasional.'
94       'Syava Aprilia 2241760129',
95       softWrap: true,
96     ), // Text
97   ); // Container
98

```

Dengan memberi nilai softWrap = true, baris teks akan memenuhi lebar kolom sebelum membungkusnya pada batas kata.

### Langkah 2: Tambahkan variabel text section ke body

Tambahkan widget variabel textSection ke dalam body seperti berikut:

```

100   return MaterialApp(
101     title: 'Flutter layout: Syava Aprilia - 2241760129', // Nama da
102     home: Scaffold(
103       appBar: AppBar(
104         title: const Text('Flutter layout demo'),
105       ), // AppBar
106       body: Center(
107         child: Column(
108           children: [
109             titleSection, // Bagian judul
110             buttonSection, // Bagian tombol
111             textSection, // Bagian teks
112           ],
113         ), // Column
114       ), // Center
115     ),

```



## Praktikum 4: Implementasi image section

Selesaikan langkah-langkah praktikum berikut ini dengan melanjutkan dari praktikum sebelumnya.

### Langkah 1: Siapkan aset gambar



Anda dapat mencari gambar di internet yang ingin ditampilkan. Buatlah folder images di root project **layout\_flutter**. Masukkan file gambar tersebut ke folder images, lalu set nama file tersebut ke file pubspec.yaml seperti berikut:

```
59 uses-material-design: true
60 assets:
61 - images/bali.jpeg
62
```

## Langkah 2: Tambahkan gambar ke body

Tambahkan aset gambar ke dalam body seperti berikut:

```
lib > main.dart > MyApp > build
7 class MyApp extends StatelessWidget {
9   Widget build(BuildContext context) {
97     ); // Container
98
99     return MaterialApp(
100     title: 'Flutter layout: Syava Aprilia - 2241760129',
101     home: Scaffold(
102       appBar: AppBar(
103         title: const Text('Flutter layout demo'),
104       ), // AppBar
105       body: Center(
106         child: Column(
107           children: [
108             Image.asset(
109               'images/bali.jpeg',
110               width: 600,
111               height: 240,
112               fit: BoxFit.cover,
113             ), // Image.asset
114             titleSection, // Bagian judul
115             buttonSection, // Bagian tombol
116             textSection, // Bagian teks
117           ],
118         ), // Column
119       ), // Center
120     ), // Scaffold
121   ); // MaterialApp
122 }
123
124 }
```

BoxFit.cover memberi tahu kerangka kerja bahwa gambar harus sekecil mungkin tetapi menutupi seluruh kotak rendernya.


## Langkah 3: Terakhir, ubah menjadi ListView

Pada langkah terakhir ini, atur semua elemen dalam ListView, bukan Column, karena ListView mendukung scroll yang dinamis saat aplikasi dijalankan pada perangkat yang resolusinya lebih kecil.

```

99   return MaterialApp(
100     title: 'Flutter layout: Syava Aprilia - 2241760129',
101     home: Scaffold(
102       appBar: AppBar(
103         title: const Text('Flutter layout demo'),
104       ), // AppBar
105       body: ListView(
106         children: [
107           Image.asset(
108             'images/bali.jpeg',
109             width: 600,
110             height: 270,
111             fit: BoxFit.cover,
112           ), // Image.asset
113           titleSection, // Bagian judul
114           buttonSection, // Bagian tombol
115           textSection, // Bagian teks
116         ],
117       ), // ListView
118     ), // Scaffold
119   ); // MaterialApp
120 }
121




```



### Pura Uluwatu Bali

Pecatu, Kec. Kuta Sel,Kab. Badung, Bali, Indonesia

★ 41

 CALL
 ROUTE
 SHARE

Pura Luhur Uluwatu terletak di Desa Pecatu, Kuta Selatan, Badung, Bali. Pura ini berdiri di atas tebing setinggi 97 meter yang menjorok ke laut dan merupakan salah satu Pura Sad Kayangan, dipercaya sebagai penyangga 9 mata angin oleh umat Hindu. Awalnya, pura ini digunakan untuk memuja Empu Kuturan, seorang pendeta suci abad ke-11, dan kemudian Dang Hyang Nirartha yang mencapai moksa di sini, yang menginspirasi nama "Luhur Uluwatu." Selain menjadi tempat ibadah, Pura Uluwatu juga terkenal dengan Pantai Pecatu di bawahnya, yang menjadi destinasi selancar internasional.Syava Aprilia 2241760129