

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL V
ANTARMUKA PENGGUNA LANJUTAN**



**Disusun Oleh :
Arzario Irsyad Al Fatih/2211104032
SE 06 2**

**Asisten Praktikum :
Muhammad Faza Zulian Gesit Al Barru
Aisyah Hasna Aulia**

**Dosen Pengampu :
Yudha Islami Sulistya**

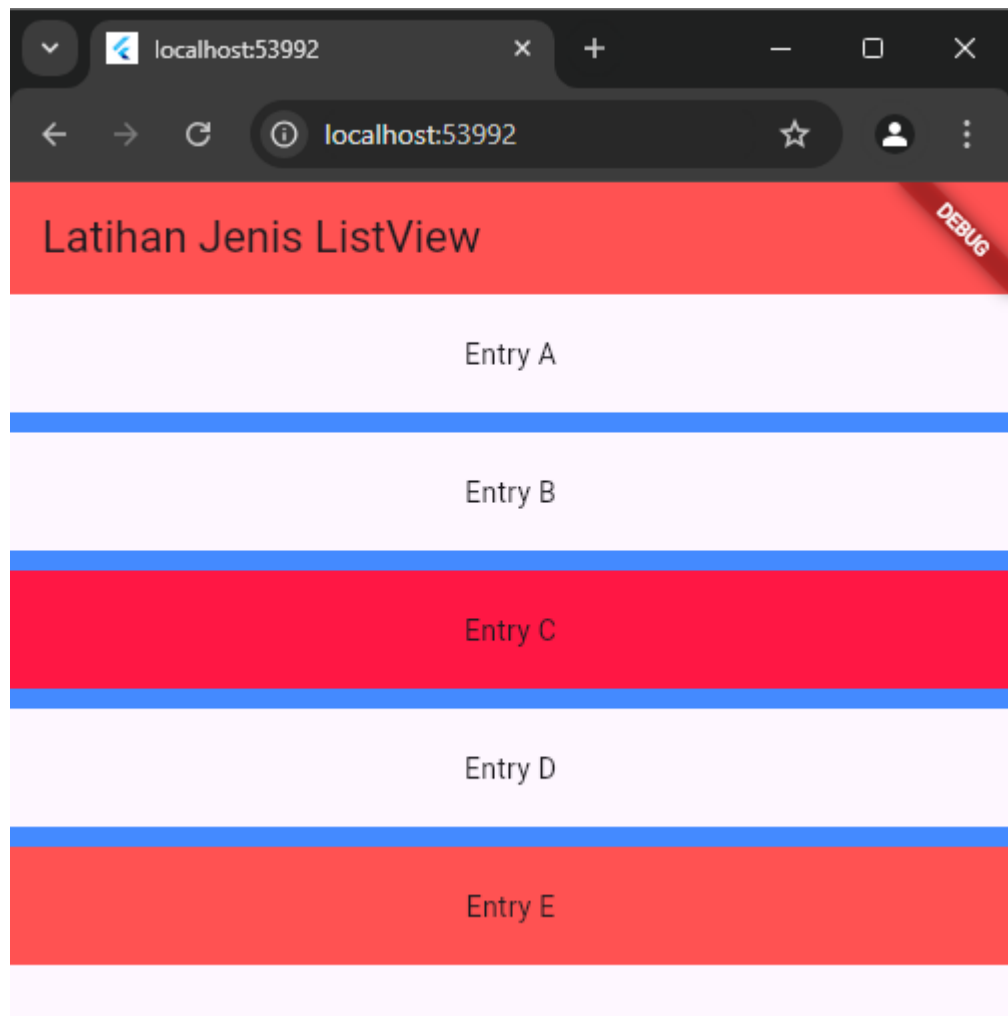
**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

1. GUIDED

a. ListView.builder

ListView.builder digunakan untuk membuat daftar (list) yang panjang dan dinamis. Daftar ini hanya membangun item yang terlihat di layar, sehingga efisien dalam penggunaan memori untuk data yang banyak. ListView.builder menerima dua parameter penting: itemCount (jumlah item yang akan ditampilkan) dan itemBuilder (fungsi yang akan memproduksi widget untuk setiap item).

```
lib > jenis_list_view.dart > JenisListView
3  class JenisListView extends StatelessWidget {
7    Widget build(BuildContext context) {
11      appBar: AppBar(
12        title: Text("Latihan Jenis ListView"),
13        backgroundColor: Colors.redAccent,
14      ), // AppBar
15      body: ListView.separated(
16        itemBuilder: (BuildContext context, int index) {
17          return Container(
18            height: 59,
19            color: Colors.redAccent[colorCodes[index]],
20            child: Center(
21              child: Text("Entry ${entries[index]}"),
22            ), // Center
23          ); // Container
24        },
25        itemCount: entries.length,
26        separatorBuilder: (BuildContext context, int index) {
27          return Container(
28            height: 10,
29            color: Colors.blueAccent,
30          ); // Container
31        }, // ListView.separated
32      ); // Scaffold
33    }
34  }
```



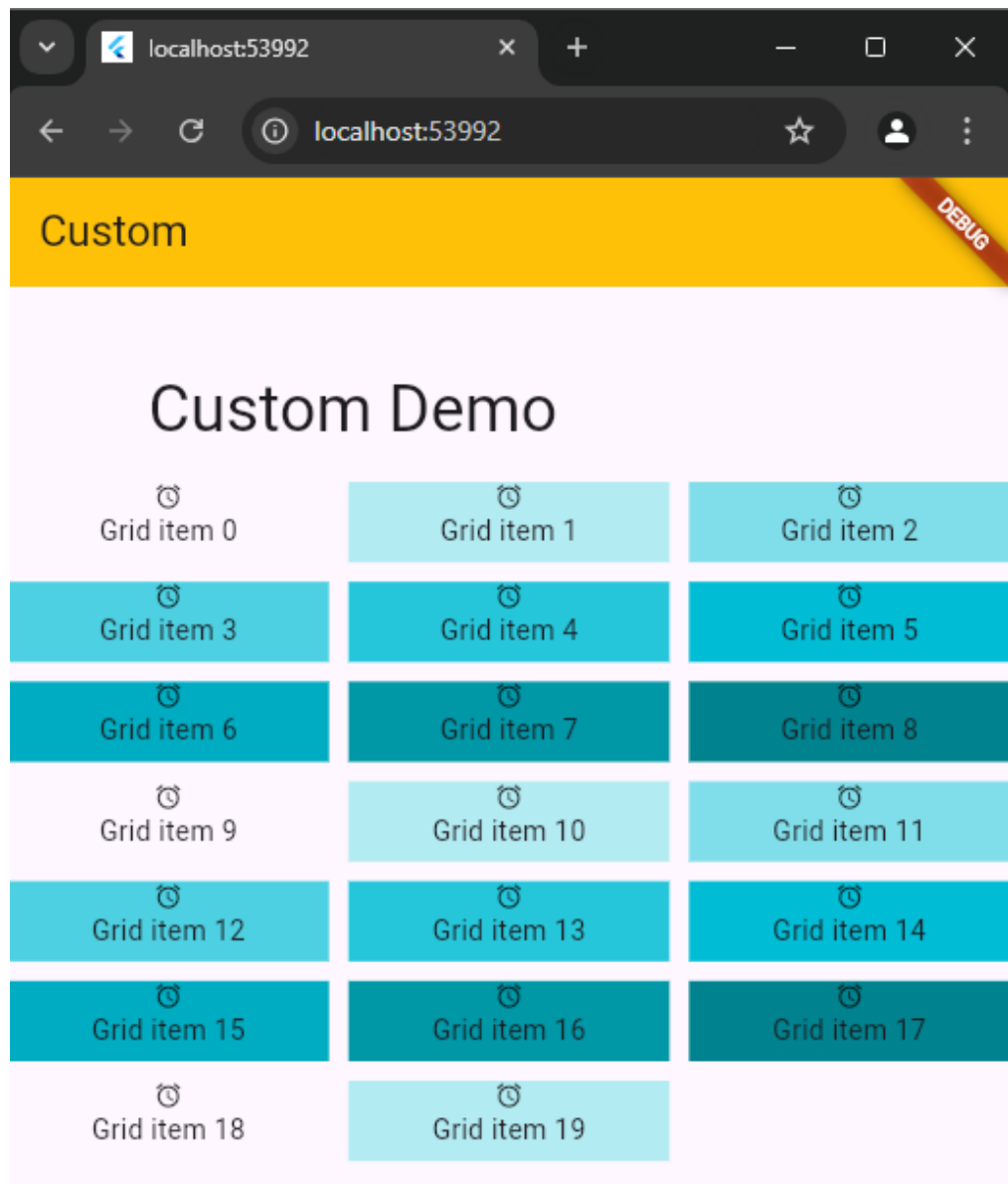
b. ListView.Separated

ListView.separated mirip dengan ListView.builder, tetapi dengan tambahan kemampuan untuk menyisipkan widget pemisah di antara setiap item, seperti garis atau ruang. Hal ini berguna jika kamu ingin memisahkan item-item dalam daftar dengan pembatas visual yang konsisten. ListView.separated menerima parameter separatorBuilder yang mengatur widget pemisah.

```

lib > custom.dart > Custom
1 import 'package:flutter/material.dart';
2
3 class Custom extends StatelessWidget {
4   const Custom({super.key});
5
6   @override
7   Widget build(BuildContext context) {
8     return Scaffold(
9       appBar: AppBar(
10        title: const Text("Custom"),
11        backgroundColor: Colors.amber,
12      ), // AppBar
13      body: CustomScrollView(
14        slivers: <Widget>[
15          const SliverAppBar(
16            pinned: true,
17            expandedHeight: 100,
18            flexibleSpace: FlexibleSpaceBar(
19              title: Text('Custom Demo'),
20            ), // FlexibleSpaceBar
21          ), // SliverAppBar
22          SliverGrid(
23            delegate: SliverChildBuilderDelegate(
24              (BuildContext context, int index) {
25                return Container(
26                  height: 100,
27                  alignment: Alignment.center,
28                  color: Colors.cyan[100 * (index % 9)],
29                  child: Column(
30                    children: [
31                      Icon(Icons.access_alarm, size: 15),
32                      Text('Grid item $index'),
33                    ],
34                  ), // Column
35                ); // Container
36              },
37              childCount: 20,
38            ), // SliverChildBuilderDelegate
39            gridDelegate: const SliverGridDelegateWithMaxCrossAxisExtent(
40              maxCrossAxisExtent: 200,
41              mainAxisSpacing: 10,
42              crossAxisSpacing: 10,
43              childAspectRatio: 4,
44            ), // SliverGridDelegateWithMaxCrossAxisExtent
45          ), // SliverGrid
46        ], // <Widget>[]
47      ), // CustomScrollView
48    ); // Scaffold
49  }
50

```



c. Flexible dan Expanded

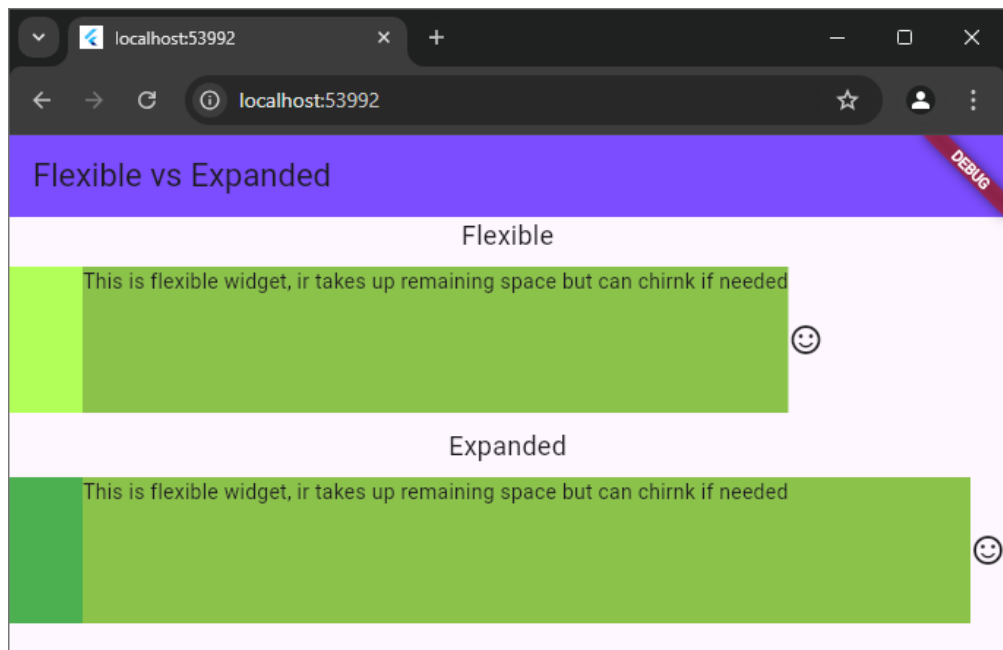
Flexible digunakan dalam widget Flex, seperti Row atau Column, untuk memungkinkan sebuah widget menyesuaikan ukurannya secara dinamis berdasarkan ruang yang tersedia. Jika ada ruang lebih, Flexible mengizinkan widget untuk mengambil ruang yang lebih besar atau lebih kecil sesuai kebutuhan, tetapi masih menjaga prioritas elemen lain yang memiliki ukuran tetap atau yang menggunakan Expanded.

Expanded juga digunakan di dalam Flex widgets, tetapi berbeda dengan Flexible, Expanded memaksa widget untuk mengisi semua ruang yang tersedia di axis utama, biasanya horizontal (Row) atau vertikal (Column). Expanded memastikan widget tersebut akan menggunakan semua ruang sisa setelah widget lain yang ukurannya ditentukan terpasang.

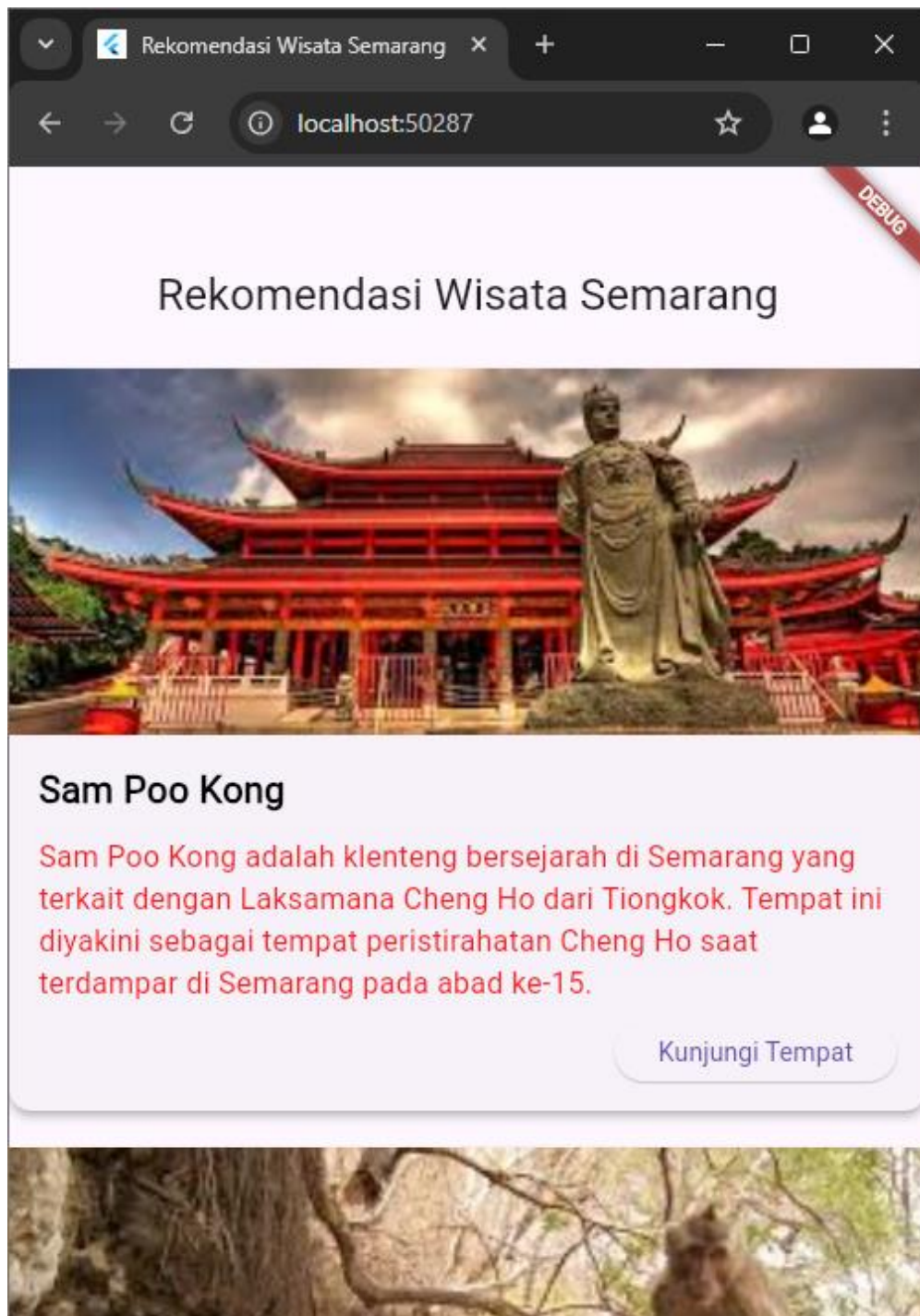
```

1 import 'package:flutter/material.dart';
2
3 class FlexibleExpanded extends StatelessWidget {
4   const FlexibleExpanded({super.key});
5
6   @override
7   Widget build(BuildContext context) {
8     return Scaffold(
9       appBar: AppBar(
10        title: const Text("Flexible vs Expanded"),
11        backgroundColor: Colors.blueGrey,
12      ), // AppBar
13      body: Padding(
14        padding: const EdgeInsets.all(16.0),
15        child: Column(
16          crossAxisAlignment: CrossAxisAlignment.start,
17          children: [
18            const Text(
19              "Expanded",
20              style: TextStyle(fontSize: 18),
21            ), // Text
22            const SizedBox(height: 8),
23            Row(
24              children: [
25                Container(
26                  width: 50,
27                  height: 100,
28                  color: Colors.green,
29                ), // Container
30                Expanded( // Menggunakan Expanded di bagian pertama
31                  child: Container(
32                    height: 100,
33                    color: Colors.lightGreen,
34                    padding: const EdgeInsets.all(8.0),
35                    child: const Text(
36                      "This is an expanded widget, it forces the widget to take up all the remaining space.",
37                      style: TextStyle(fontSize: 10),
38                    ), // Text
39                  ), // Container
40                ), // Expanded
41                const Icon(Icons.sentiment_satisfied_alt_outlined),
42              ], // Row
43            ), // Row
44            const SizedBox(height: 16),
45            const Text(
46              "Flexible",
47              style: TextStyle(fontSize: 18),
48            ), // Text
49            const SizedBox(height: 8),
50            Row(
51              children: [
52                Container(
53                  width: 50,
54                  height: 100,
55                  color: Colors.green,
56                ), // Container
57                Flexible( // Menggunakan Flexible di bagian kedua
58                  child: Container(
59                    height: 100,
60                    color: Colors.lightGreen,
61                    padding: const EdgeInsets.all(8.0),
62                    child: const Text(
63                      "This is a flexible widget, it takes up the remaining space but can shrink if needed.",
64                      style: TextStyle(fontSize: 10),
65                    ), // Text
66                  ), // Container
67                ), // Flexible
68                const Icon(Icons.sentiment_satisfied_alt_outlined),
69              ], // Row
70            ), // Row
71          ], // Column
72        ), // Padding
73      ); // Scaffold
74    }
75  }

```



- 2. UNGUIDED
 - a. Output



b. Code


```

import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Rekomendasi Wisata Semarang',
      home: TouristRecommendationScreen(),
    );
  }
}

class TouristRecommendationScreen extends StatelessWidget {
  final List<TouristPlace> places = [
    TouristPlace(
      name: 'Sam Poo Kong',
      description:
        'Sam Poo Kong adalah klinteng bersejarah di Semarang yang terkait dengan Laksamana Cheng Ho dari Tiongkok. Tempat ini diyakini sebagai tempat peristirahatan Cheng Ho saat terdampar di Semarang pada abad ke-15.',
      imageUrl:
        'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ1SZFf6zXdT-Blz341JIN-IrADSoJ2bQjAeg&s',
    ),
    TouristPlace(
      name: 'Goa Kreo',
      description:
        'Goa Kreo adalah tempat wisata alam dengan gua alami yang dihuni oleh banyak monyet. Terletak di dekat Waduk Jatibarang, Goa Kreo menawarkan suasana sejuk dan pemandangan alam yang indah.',
      imageUrl:
        'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSNF5TKWYujshPCeGLYft3qgs0MgqvFXpQ9wA&s',
    ),
    TouristPlace(
      name: 'Museum Ranggawarsito',
      description:
        'Museum Ranga Warsito adalah museum yang menyimpan koleksi benda-benda sejarah, budaya, dan teknologi. Terletak di Semarang, museum ini menyajikan kisah perjalanan panjang dan perkembangan Kota Semarang.',
      imageUrl:
        'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSg7skGN2M0QsCvSok1fKqYR4PEN04VI9Az4g&s',
    ),
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: CustomScrollView(
        slivers: [
          SliverAppBar(
            expandedHeight: 100,
            floating: false,
            pinned: true,
            flexibleSpace: FlexibleSpaceBar(
              title: Text(
                'Rekomendasi Wisata Semarang',
                style: TextStyle(fontSize: 16.0),
              ),
              centerTitle: true,
              titlePadding: EdgeInsets.only(bottom: 16),
            ),
          ),
          SliverList(
            delegate: SliverChildBuilderDelegate(
              (BuildContext context, int index) {
                return _buildPlaceCard(context, places[index]);
              },
              childCount: places.length,
            ),
          ),
        ],
      ),
    );
  }

  Widget _buildPlaceCard(BuildContext context, TouristPlace place) {
    return Card(
      required this.name,
      required this.description,
      required this.imageUrl,
    );
  }
}

```

```

    ],
  ),
);
}

Widget _buildPlaceCard(BuildContext context, TouristPlace place) {
  return Card(
    margin: const EdgeInsets.symmetric(vertical: 10),
    elevation: 4,
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Image.network(
          place.imageUrl,
          width: double.infinity,
          height: 200,
          fit: BoxFit.cover,
        ),
        Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                place.name,
                style: const TextStyle(
                  fontSize: 20,
                  fontWeight: FontWeight.bold,
                  color: Colors.black,
                ),
              ),
              const SizedBox(height: 10),
              Text(
                place.description,
                style: const TextStyle(
                  fontSize: 16,
                  color: Color.fromARGB(255, 246, 29, 29),
                ),
              ),
              const SizedBox(height: 10),
              Row(
                mainAxisAlignment: MainAxisAlignment.end,
                children: [
                  ElevatedButton(
                    onPressed: () {
                      ScaffoldMessenger.of(context).showSnackBar(
                        SnackBar(
                          content: Text('Mengunjungi ${place.name}'),
                        ),
                      );
                    },
                  ),
                  const Text('Kunjungi Tempat'),
                ],
              ),
            ],
          ),
        ),
      ],
    ),
  );
}

class TouristPlace {
  final String name;
  final String description;
  final String imageUrl;

  TouristPlace

```

