

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL V
ANTARMUKA PENGGUNA LANJUTAN**



**Disusun Oleh :
Dewi Atika Muthi / 2211104042
SE-06-02**

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

**Dosen Pengampu :
Yudha Islami Sulistya, S.Kom., M.Cs**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2024

BAB I

PENDAHULUAN

A. DASAR TEORI

Flutter adalah framework open-source yang dikembangkan oleh Google untuk membangun antarmuka pengguna (UI) dalam aplikasi lintas platform. Keistimewaan Flutter terletak pada pendekatan pengembangan yang berbeda, menggunakan bahasa pemrograman Dart. Dart merupakan bahasa yang dioptimalkan untuk kinerja tinggi, dan dengan adanya hot-reload, pengembang dapat melihat perubahan langsung pada aplikasi saat proses pengembangan. Berikut

Berikut beberapa konsep penting yang dibahas dalam modul ini meliputi:

1. **ListView.builder**: Widget ini digunakan untuk membuat daftar scrollable yang efisien untuk data dalam jumlah besar. ListView.builder hanya merender item yang terlihat di layar, mengoptimalkan penggunaan memori dan kinerja.
2. **ListView.separated**: Varian dari ListView yang memungkinkan penambahan pemisah (separator) antara item-item dalam daftar. Ini berguna untuk meningkatkan keterbacaan dan estetika daftar.
3. **Flexible dan Expanded**:
 - **Flexible**: Widget yang memberikan fleksibilitas kepada child-nya dalam mengambil ruang yang tersedia, dengan mempertimbangkan faktor flex.
 - **Expanded**: Turunan dari Flexible yang secara otomatis mengisi semua ruang yang tersedia tanpa batasan.
4. **CustomScrollView**: Widget scrollable yang memungkinkan pembuatan custom scrolling effects dengan menggunakan sliver widgets seperti SliverAppBar, SliverList, dan SliverGrid.

Penggunaan widget-widget ini memungkinkan pengembang untuk membuat antarmuka pengguna yang dinamis dan responsif, meningkatkan pengalaman pengguna dalam aplikasi Flutter.

B. MAKSUD DAN TUJUAN

Praktikum ini bertujuan untuk:

1. Memahami dan mengimplementasikan konsep layout lanjutan dalam Flutter.
2. Mempraktikkan penggunaan ListView.builder dan ListView.separated untuk menampilkan data dalam bentuk list.
3. Memahami perbedaan dan use case untuk widget Flexible dan Expanded.
4. Mengeksplorasi penggunaan CustomScrollView untuk membuat antarmuka scrollable yang kompleks.
5. Mengaplikasikan pengetahuan dalam membuat aplikasi rekomendasi wisata dengan UI yang menarik dan fungsional.

BAB II IMPLEMENTASI

A. PRAKTIKUM

1. ListView.builder

Soal Studi Case

Membuat tampilan daftar (list) menggunakan widget ListView.builder di Flutter untuk membuat tampilan daftar yang dinamis dan efisien, terutama ketika jumlah elemen dalam daftar besar. Setiap item dalam daftar diwarnai secara berbeda.

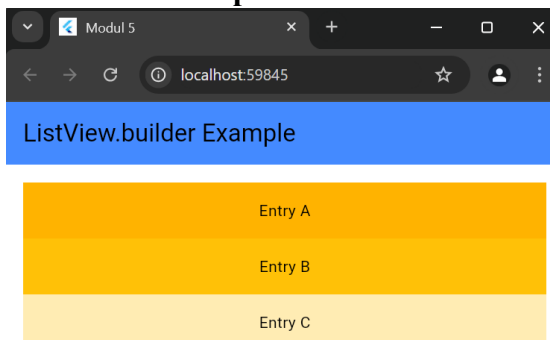
Sourcecode

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

class ViewBuild extends StatelessWidget {
  final List<String> entries = <String>['A', 'B', 'C'];
  final List<int> colorCodes = <int>[600, 500, 100];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("ListView.builder Example"),
        backgroundColor: Colors.blueAccent,
      ),
      body: Padding(
        padding: const EdgeInsets.all(8.0),
        child: ListView.builder(
          padding: const EdgeInsets.all(8),
          itemCount: entries.length,
          itemBuilder: (BuildContext context, int index) {
            return Container(
              height: 50,
              color: Colors.amber[colorCodes[index]],
              child: Center(
                child: Text('Entry ${entries[index]}'),
              ),
            );
          },
        ),
      ),
    );
  }
}
```

Screenshoot Output



Deskripsi Program

Data List:

entries: Sebuah list berisi tiga string, yaitu "A", "B", dan "C". List ini digunakan untuk menampilkan teks pada setiap elemen daftar.

colorCodes: Sebuah list berisi tiga nilai integer yang merepresentasikan kode warna amber dengan intensitas berbeda (600, 500, dan 100).

ListView.builder:

Menggunakan ListView.builder untuk menampilkan daftar elemen dengan jumlah item sesuai dengan panjang list entries.

Setiap elemen ditampilkan dalam Container dengan tinggi 50px dan latar belakang warna yang diambil dari colorCodes.

Di dalam Container, terdapat widget Center yang menempatkan teks di tengah, menampilkan teks "Entry A", "Entry B", dan "Entry C" berdasarkan urutan indeks dari entries.

2. ListView.separated

Soal Studi Case

Membuat tampilan daftar item menggunakan widget ListView.separated. Masing-masing item di daftar ini berupa huruf (A, B, C, D), yang diberi warna latar belakang yang berbeda-beda.

Sourcecode

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

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

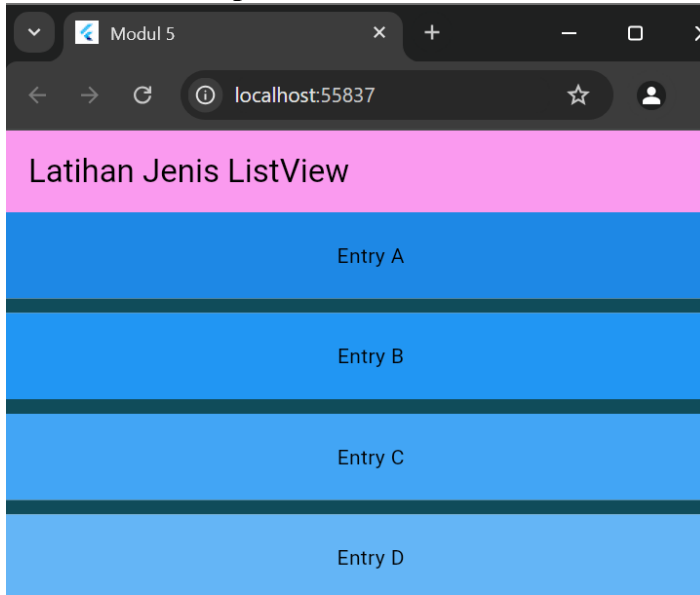
  @override
  Widget build(BuildContext context) {
    final List<String> entries = <String>['A', 'B', 'C', 'D'];
    final List<int> colorCodes = <int>[600, 500, 400, 300];
    return Scaffold(
      appBar: AppBar(
        title: const Text("Latihan Jenis ListView"),
        backgroundColor: const Color.fromARGB(255, 250, 154, 239),
      ),
      body: ListView.separated(
        itemBuilder: (BuildContext context, int index) {
          return Container(
            height: 59,
            color: Colors.blue[colorCodes[index]],
            child: Center(
              child: Text("Entry ${entries[index]}"),
            ),
          );
        },
        itemCount: entries.length,
        separatorBuilder: (BuildContext context, int index) {
          return Container(
            height: 10,
            color: const Color.fromARGB(255, 16, 76, 90),
          );
        },
      ),
    );
  }
}
```

```

    },
  ),
);
}
}

```

Screenshoot Output



Deskripsi Program

Komponen utama dari aplikasi ini adalah widget `ListView.separated`, yang digunakan untuk menampilkan daftar elemen dengan pemisah di antaranya.

- **Item dalam List:** Daftar terdiri dari 4 elemen yang diberi label "Entry A", "Entry B", "Entry C", dan "Entry D". Setiap elemen dalam daftar memiliki latar belakang berwarna biru dengan variasi intensitas warna (kode warna 600, 500, 400, dan 300).
- **Pemrosesan Pemisah:** Antara setiap elemen dalam daftar, terdapat pemisah berupa container dengan tinggi 10 dan warna latar belakang biru tua (kode ARGB: 255, 16, 76, 90). Pemisah ini memberikan ruang visual antar elemen, sehingga daftar terlihat lebih terstruktur dan mudah dibaca.

Pewarnaan Dinamis: Pewarnaan dinamis diterapkan pada setiap item dalam daftar. Warna latar belakang item diambil dari daftar warna `colorCodes`, yang berisi kode warna biru dengan intensitas yang bervariasi, mulai dari biru tua hingga biru muda

3. Flexible dan Expanded

Soal Studi Case

Membuat program untuk menampilkan perbedaan antara widget Flexible dan Expanded dalam sebuah antarmuka menggunakan kolom dan baris.

Sourcecode

```

import 'package:flutter/material.dart';

class FlexExp extends StatelessWidget {

```

```

const FlexExp({super.key});

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text("Flexible and Expanded"),
      backgroundColor: const Color.fromARGB(255, 14, 135, 210),
    ),
    body: Column(
      children: [
        Text(
          "Flexible",
          style: TextStyle(fontSize: 18),
        ),
        SizedBox(
          height: 8,
        ),
        Row(
          children: [
            Container(
              width: 50,
              height: 100,
              color: const Color.fromARGB(255, 251, 175, 255),
            ),
            Flexible(
              child: Container(
                height: 100,
                color: const Color.fromARGB(255, 115, 222, 255),
                child: Text(
                  "This is flexible widget, it takes up
remaining space but can chirk if needed"),
                ),
              ),
            Icon(Icons.sentiment_satisfied_alt_outlined)
          ],
        ),
        SizedBox(
          height: 10,
        ),
        Text(
          "Expanded",
          style: TextStyle(fontSize: 18),
        ),
        SizedBox(
          height: 8,
        ),
        Row(
          children: [
            Container(
              width: 50,
              height: 100,
              color: const Color.fromARGB(255, 251, 175, 255),
            ),
            Expanded(
              child: Container(
                height: 100,
                color: const Color.fromARGB(255, 115, 222, 255),
                child: Text(
                  "This is Expanded Widget, it forces the
widget to take up all the remaining space"),
                ),
              ),
            Icon(Icons.sentiment_satisfied_alt_outlined)
          ],
        ),
      ],
    ),
  );
}

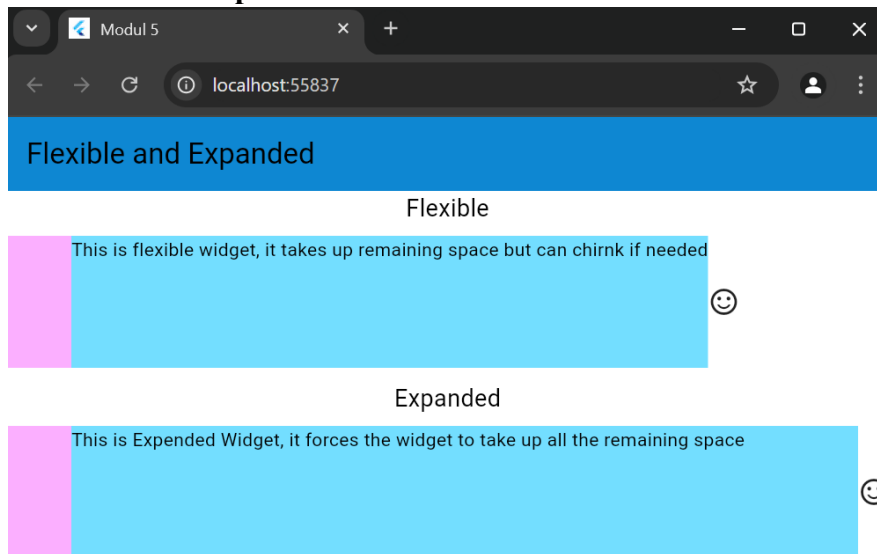
```

```

        ),
    ),
    Icon(Icons.sentiment_satisfied_alt_outlined)
  ],
),
],
),
);
}
}

```

Screenshoot Output



Deskripsi Program

Flexible Widget:

Bagian pertama antarmuka menampilkan teks "Flexible" diikuti oleh sebuah Row yang berisi tiga elemen:

- Container berukuran tetap (50px lebar, 100px tinggi) dengan latar belakang ungu muda.
- Sebuah Flexible widget yang berisi Container dengan teks yang menampilkan deskripsi singkat tentang Flexible. Widget **ini dapat mengisi ruang yang tersisa** dalam baris **tetapi akan menyusut jika diperlukan**, sesuai dengan ruang yang tersedia.
- Sebuah Icon berupa wajah tersenyum di ujung baris.

Expanded Widget:

Bagian kedua antarmuka menampilkan teks "Expanded" diikuti oleh Row serupa dengan tiga elemen:

- Container berukuran tetap sama dengan warna ungu muda.
- Sebuah Expanded widget yang berisi Container dengan teks penjelasan tentang Expanded. Widget ini memaksa kontainer untuk **mengambil seluruh ruang yang tersisa dalam baris**, memastikan **semua ruang diisi oleh widget tersebut**.
- Icon yang sama seperti pada baris sebelumnya.

4. CustomScrollView

Soal Studi Case

Membuat tata letak scrollable yang dapat menampung beberapa jenis elemen, termasuk SliverAppBar dan SliverGrid.

Sourcecode

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

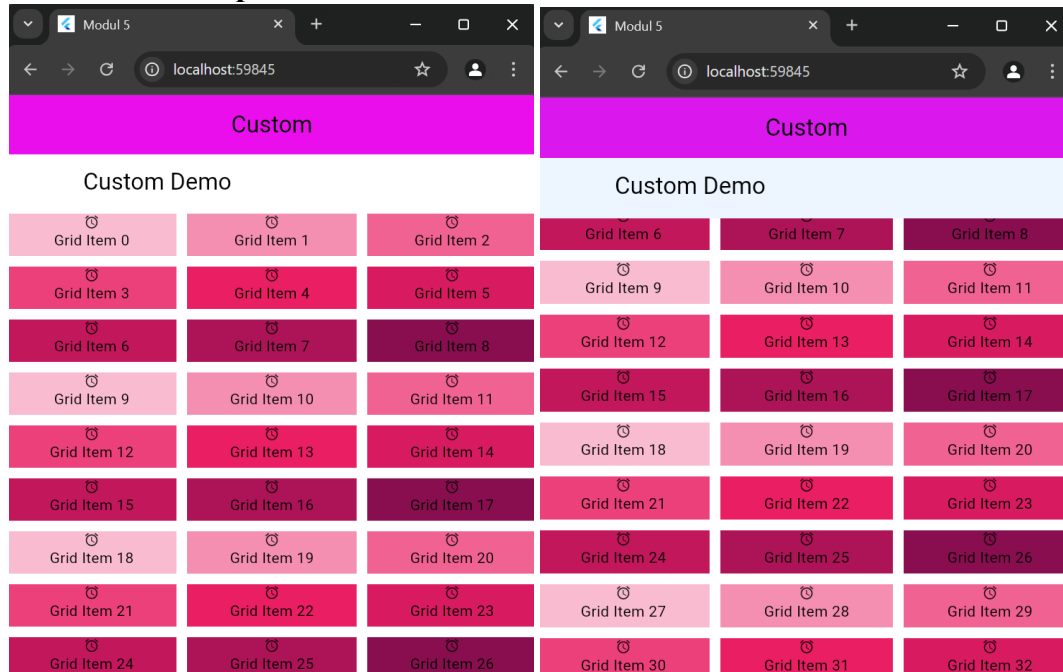
class Customscrollview extends StatelessWidget {
  const Customscrollview({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Custom"),
        centerTitle: true,
        backgroundColor: const Color.fromARGB(255, 234, 14, 237),
      ),
      body: CustomScrollView(
        slivers: <Widget>[
          // Sliver App Bar
          const SliverAppBar(
            pinned: true,
            expandedHeight: 20,
            flexibleSpace: FlexibleSpaceBar(
              title: Text('Custom Demo'),
            ),
          ),
          // Sliver Grid
          SliverGrid(
            delegate: SliverChildBuilderDelegate(
              (BuildContext context, int index) {
                return Container(
                  alignment: Alignment.center,
                  color: Colors.pink[100 * (index % 9 + 1)],
                  child: Column(
                    children: [
                      Icon(Icons.access_alarm, size: 15),
                      Text('Grid Item $index')
                    ],
                  ),
                );
              },
            ),
            childCount: 25,
            gridDelegate: const
SliverGridDelegateWithMaxCrossAxisExtent(
              maxCrossAxisExtent: 200,
              mainAxisSpacing: 10,
              crossAxisSpacing: 10,
              childAspectRatio: 4,
            ),
          ),
          // Sliver Fixed List
        ],
      ),
    );
  }
}
```



```
}
```

Screenshot Output



Deskripsi Program

SliverAppBar:

Pada bagian atas tampilan scroll, terdapat **SliverAppBar** yang memiliki properti **pinned**, artinya bar tersebut **akan tetap berada di bagian atas layar** saat pengguna menggulir ke bawah. Tinggi **SliverAppBar** ini bersifat dinamis, dapat diperluas hingga 20px. Di dalamnya terdapat **FlexibleSpaceBar** yang menampilkan teks "Custom Demo".

SliverGrid:

Di bawah **SliverAppBar**, terdapat **SliverGrid** yang menampilkan **daftar grid** yang terdiri dari 35 item. Setiap item ditampilkan dalam Container dengan tinggi dan warna latar yang berbeda-beda berdasarkan indeksinya. Setiap grid menampilkan ikon alarm kecil (`Icons.access_alarm`) dan teks yang menunjukkan nomor item, seperti "Grid Item 0", "Grid Item 1", dan seterusnya. Warna setiap item bervariasi secara bertahap dengan warna pink yang berbeda tingkat kecerahannya.

GridDelegate:

Untuk pengaturan grid, digunakan **SliverGridDelegateWithMaxCrossAxisExtent** yang membatasi lebar maksimum setiap item hingga 200px, dengan jarak antar-item 10px di setiap sisi.

5. TUGAS MANDIRI (Unguided)

1. Modifikasi project Rekomendasi Wisata pada Tugas Unguided 04 modul Antarmuka Pengguna dengan mengimplementasikan widget CustomScrollView, SliverAppBar, dan SliverList untuk merekomendasikan beberapa tempat wisata disertai foto, nama wisata, dan deskripsi singkat! (buatlah se kreatif mungkin).
(p/s: izin menggunakan wisata Malang karena keterbatasan izin gambar Banyumas)

Sourcecode:

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

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

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Wisata Malang',
      theme: ThemeData(
        primaryColor: Color.fromARGB(255, 143, 221, 243),
        fontFamily: 'Roboto',
      ),
      home: const WisataMalangPage(),
    );
  }
}

class WisataMalangPage extends StatelessWidget {
  const WisataMalangPage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: CustomScrollView(
        slivers: <Widget>[
          SliverAppBar(
            expandedHeight: 200.0,
            floating: false,
            pinned: true,
            flexibleSpace: FlexibleSpaceBar(
              title: const Text(
                'Wisata Malang',
                style: TextStyle(
                  color: Colors.white,
                  fontWeight: FontWeight.bold,
                ),
              ),
            ),
            background: Image.network(
              'https://asset.kompas.com/crops/ikjJCP121V1kdUT6o12wRqPpEAk=/0x363:765x873/750x500/data/photo/2023/12/26/658a7bf12e3f1.jpeg',
              fit: BoxFit.cover,
            ),
            backgroundColor: const Color.fromARGB(255, 143, 221, 243),
          ),
        ],
      ),
    );
  }
}
```

```

SliverPadding(
  padding: const EdgeInsets.all(10.0),
  sliver: SliverList(
    delegate: SliverChildListDelegate([
      const WisataItem(
        image:
'https://upload.wikimedia.org/wikipedia/commons/e/ec/Mount_Bromo_panorama.jpg',
        name: 'Gunung Bromo',
        description:
'Gunung berapi aktif dengan pemandangan matahari terbit yang menakjubkan.',
        color: Colors.orange,
        icon: Icons.landscape,
      ),
      const WisataItem(
        image:
'https://img.inews.co.id/media/1200/files/inews_new/2021/08/09/jatim_park.jpg',
        name: 'Jatim Park',
        description:
'Taman hiburan keluarga dengan berbagai wahana dan atraksi menarik.',
        color: Colors.blue,
        icon: Icons.attractions,
      ),
      const WisataItem(
        image:
'https://static.promediateknologi.id/crop/0x0:0x0/0x0/webp/photo/p2/231/2024/09/11/Segarnya-suasana-kawasan-Air-Terjun-Coban-Rondo-Gambar-dari-Dhiky-Aditya-Shutterstock-1200347082.png',
        name: 'Air Terjun Coban Rondo',
        description:
'Air terjun indah dengan ketinggian 84 meter, dikelilingi hutan pinus.',
        color: Colors.teal,
        icon: Icons.water_drop,
      ),
      const WisataItem(
        image:
'https://upload.wikimedia.org/wikipedia/commons/thumb/f/fb/Batu_Secret_Zoo.jpg/300px-Batu_Secret_Zoo.jpg',
        name: 'Batu Secret Zoo',
        description:
'Kebun binatang modern dengan koleksi hewan dari berbagai belahan dunia.',
        color: Colors.green,
        icon: Icons.pets,
      ),
      const WisataItem(
        image:
'https://static.promediateknologi.id/crop/0x0:0x0/0x0/webp/photo/p2/74/2024/03/27/Pantai-Balekambang-1601454064.png',
        name: 'Pantai Balekambang',
        description:

```

```

        'Pantai eksotis dengan Pura Amerta Jati yang
        menyerupai Tanah Lot di Bali.',
        color: Colors.blue,
        icon: Icons.beach_access,
    ),
  ],
),
],
),
);
}
}

class WisataItem extends StatelessWidget {
  final String image;
  final String name;
  final String description;
  final Color color;
  final IconData icon;

  const WisataItem({
    Key? key,
    required this.image,
    required this.name,
    required this.description,
    required this.color,
    required this.icon,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Card(
      margin: const EdgeInsets.only(bottom: 16.0),
      elevation: 5.0,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(15.0),
      ),
      child: Column(
        children: [
          ClipRRect(
            borderRadius:
              const BorderRadius.vertical(top: Radius.circular(15.0)),
            child: Image.network(
              image,
              height: 200,
              width: double.infinity,
              fit: BoxFit.cover,
            ),
          ),
          Container(
            color: color.withOpacity(0.1),
            padding: const EdgeInsets.all(16.0),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Row(
                  children: [
                    Icon(icon, color: color),
                    const SizedBox(width: 8),
                    Text(

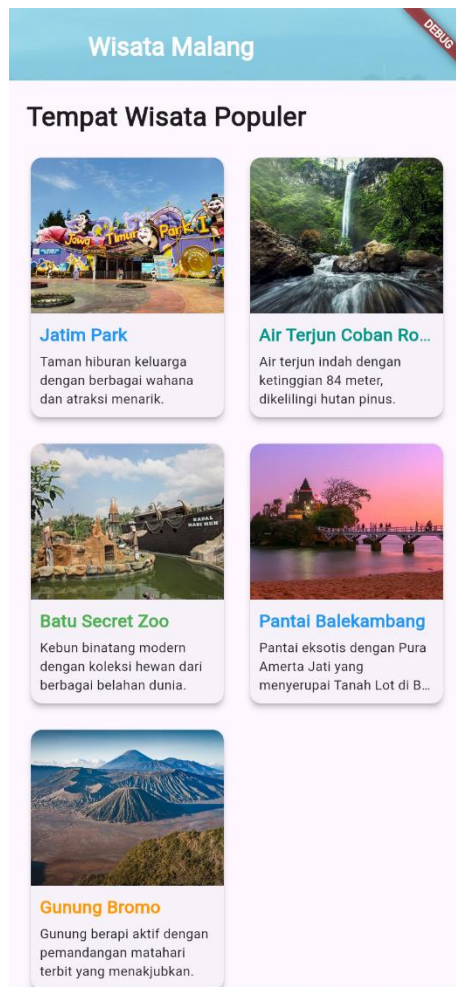
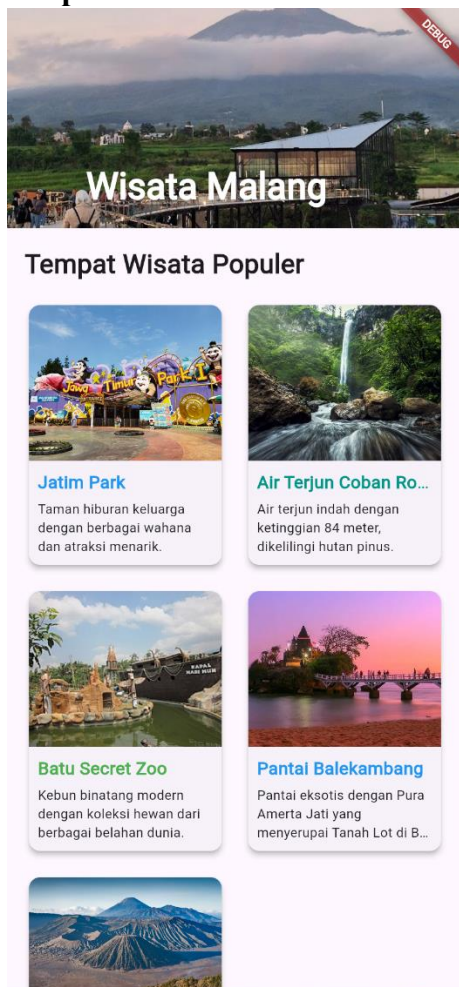
```

```

        name,
        style: TextStyle(
          fontSize: 20,
          fontWeight: FontWeight.bold,
          color: color,
        ),
      ),
    ],
  ),
  const SizedBox(height: 8),
  Text(
    description,
    style: TextStyle(
      fontSize: 16,
      color: Colors.grey[700],
    ),
  ),
),
],
),
),
),
),
);
}
}

```

Output:



Deskripsi Program:

CustomScrollView:

- Mengatur scrolling keseluruhan halaman
- Memungkinkan penggunaan berbagai jenis sliver widget

SliverAppBar:

- App bar yang dapat di-scroll
- Memiliki gambar latar dan judul yang muncul/hilang saat di-scroll

FlexibleSpaceBar:

- Bagian dari SliverAppBar yang dapat diperluas
- Menampilkan gambar latar dan judul "Wisata Malang"

SliverPadding

- Memberikan padding pada sliver child-nya

SliverToBoxAdapter

- Mengubah widget biasa menjadi sliver
- Digunakan untuk menampilkan judul "Tempat Wisata Populer"

SliverGrid

- Menampilkan grid yang dapat di-scroll
- Berisi card-card tempat wisata

SliverGridDelegateWithMaxCrossAxisExtent

- Mengatur layout grid
- Memastikan responsivitas dengan mengatur ukuran maksimum setiap item

Card

- Menampilkan informasi tempat wisata dalam bentuk card

ClipRRect

- Memotong gambar agar sesuai dengan bentuk card

Column

- Mengatur layout vertikal dalam card

Expanded

- Membagi ruang secara fleksibel antara gambar dan teks dalam card

BAB III

KESIMPULAN DAN SARAN

A. KESIMPULAN

Praktikum ini telah memberikan pemahaman mendalam tentang penggunaan widget-widget lanjutan dalam Flutter untuk membuat antarmuka pengguna yang lebih dinamis dan responsif. Melalui implementasi `ListView.builder` dan `ListView.separated`, kita belajar cara efisien untuk menampilkan daftar data yang panjang. Penggunaan `Flexible` dan `Expanded` memungkinkan kita untuk membuat layout yang adaptif terhadap berbagai ukuran layar. `CustomScrollView` dengan berbagai sliver widgets membuka kemungkinan untuk membuat antarmuka scrolling yang kompleks dan menarik.

B. REFERENSI

Flutter Documentation. (2023). `ListView` class.

<https://api.flutter.dev/flutter/widgets/ListView-class.html>

Flutter Documentation. (2023). `Flexible` class. <https://api.flutter.dev/flutter/widgets/Flexible-class.html>

Flutter Documentation. (2023). `Expanded` class.

<https://api.flutter.dev/flutter/widgets/Expanded-class.html>

Flutter Documentation. (2023). `CustomScrollView` class.

<https://api.flutter.dev/flutter/widgets/CustomScrollView-class.html>