

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL V
ANTARMUKA PENGGUNA LANJUTAN**



Disusun Oleh :

Zivana Afra Yulianto / 2211104039

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

1. Pengantar Antarmuka Pengguna (UI)

Antarmuka pengguna (UI) adalah elemen aplikasi yang berinteraksi langsung dengan pengguna. Desain UI bertujuan untuk memfasilitasi interaksi yang intuitif dan efisien, terutama dalam aplikasi wisata yang harus menarik dan mudah dinavigasi.

2. Komponen Dasar Antarmuka Pengguna

- **Layout:** Penataan elemen UI menggunakan **CustomScrollView**, **SliverAppBar**, dan **SliverList** untuk menyajikan konten secara responsif.
- **Warna dan Tipografi:** Pemilihan warna dan font yang tepat untuk menciptakan pengalaman visual yang menarik dan profesional.
- **Interaksi:** Mendesain tombol dan elemen interaktif agar interaksi pengguna terasa natural, seperti tombol "Lihat Detail".

3. Tujuan Desain Antarmuka Pengguna

- **Meningkatkan Pengalaman Pengguna:** Memberikan pengalaman positif yang mendorong eksplorasi aplikasi.
- **Mengoptimalkan Navigasi:** Memudahkan pengguna menemukan informasi tanpa kebingungan.
- **Menyampaikan Informasi Secara Efektif:** Menyajikan konten dengan jelas agar mudah dipahami.

B. MAKSUD DAN TUJUAN

1. Maksud

Maksud dari modul ini adalah untuk memberikan panduan dalam implementasi elemen desain UI yang kompleks dan dinamis di Flutter, membantu pengembang memahami penggunaan widget lanjutan.

2. Tujuan

- **Peningkatan Keterampilan Pengembang:** Memperdalam pemahaman tentang widget Flutter seperti **CustomScrollView** dan **SliverAppBar**.
- **Desain Responsif:** Mengajarkan teknik agar aplikasi beradaptasi dengan berbagai perangkat.
- **Solusi Praktis:** Menyediakan contoh aplikasi sebagai referensi.
- **Mendorong Kreativitas:** Menginspirasi pengembang untuk bereksperimen dengan desain UI.

BAB II IMPLEMENTASI

A. GUIDED

1. JENIS LIST VIEW

Code:

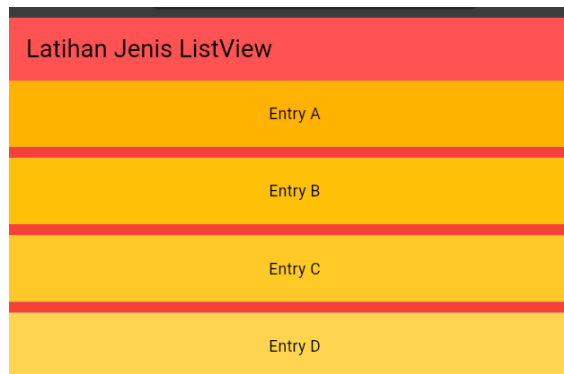
```
import 'package:flutter/material.dart';

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: Colors.redAccent,
      ),
      body: ListView.separated(
        itemBuilder: (BuildContext context, int index) {
          return Container(
            height: 59,
            color: Colors.amber[colorCodes[index]],
            child: Center(
              child: Text("Entry ${entries[index]}"),
            ),
          );
        },
        itemCount: entries.length,
        separatorBuilder: (BuildContext context, int index) {
          return Container(
            height: 10,
            color: Colors.red,
          );
        },
      ),
    );
  }
}
```

Screenshoot :



Deskripsi :

- **List Data:**

- `entries`: Daftar sederhana berisi nilai-nilai 'A', 'B', 'C', dan 'D' yang akan ditampilkan dalam daftar.

- **ListView.separated:**

- `ListView.separated` digunakan untuk menampilkan daftar dengan pemisah berupa garis horizontal (`Divider()`).
- Setiap elemen dalam daftar akan ditampilkan sebagai `ListTile` dengan teks "Entry A", "Entry B", "Entry C", atau "Entry D" sesuai indeksinya.

- **AppBar:**

- Menampilkan judul sederhana "Latihan ListView".

2. FLEXIBLE EXPANDED

Code :

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

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

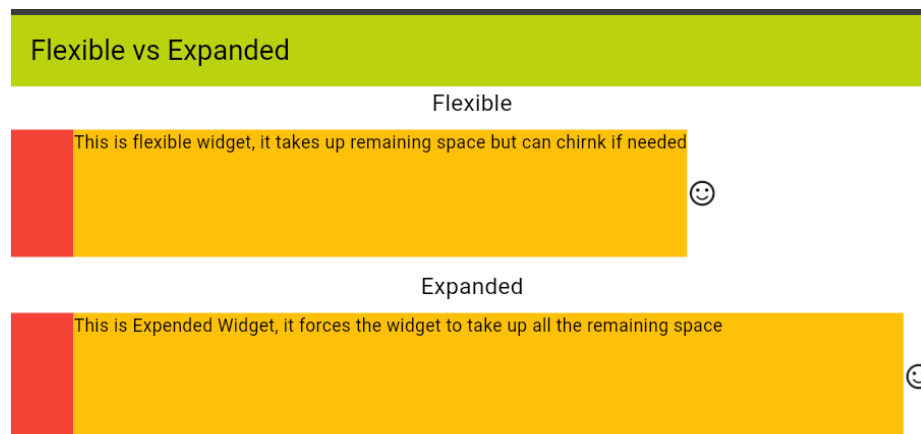
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Flexible vs Expanded"),
        backgroundColor: const Color.fromRGBO(187, 210, 14, 1),
      ),
      body: Column(
        children: [
          Text(
            "Flexible",
            style: TextStyle(fontSize: 18),
          ),
          SizedBox(
            height: 8,
          ),
          Row(
            children: [
              Container(
                width: 50,
                height: 100,
                color: Colors.red,
              ),
              Flexible(
                child: Container(
                  height: 100,
                  color: Colors.amber,
                  child: Text(
                    "This is flexible widget, it takes up remaining space but can shrink 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: Colors.red,
            ),
            Expanded(
              child: Container(
                height: 100,
                color: Colors.amber,
                child: Text(
                  "This is Expanded Widget, it forces the widget to take up all the
remaining space"),
                ),
            Icon(Icons.sentiment_satisfied_alt_outlined)
          ],
        ),
      ],
    );
  }
}

```

Screenshoot :



Deskripsi :

Kode ini adalah contoh sederhana untuk menunjukkan perbedaan antara widget **Flexible** dan **Expanded** dalam Flutter. Aplikasi menampilkan dua contoh dalam bentuk **Row** yang mengandung beberapa widget:

1. **Flexible:**

- Widget **Flexible** mengambil sisa ruang yang tersedia, tetapi masih dapat mengecil jika ada ruang terbatas. Dalam contoh ini, teks di dalam `Flexible` disisipkan di antara kotak merah dan ikon, mengambil sisa ruang yang ada di baris tersebut.

2. **Expanded:**

- Widget **Expanded** memaksa widget di dalamnya untuk mengambil seluruh sisa ruang yang tersedia dalam baris, tanpa bisa mengecil. Dalam contoh ini, teks di dalam `Expanded` mengambil semua ruang yang tersisa antara kotak merah dan ikon.

Setiap contoh ditampilkan dalam baris terpisah, dengan judul yang menunjukkan perbedaan antara **Flexible** dan **Expanded**.

3. CUSTOM SCROLL VEW

Code :

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

class Customscrollveiw extends StatelessWidget {

  const Customscrollveiw({super.key});

  @override
  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: const Text("Custom"),

        centerTitle: true,

        backgroundColor: const Color.fromRGBO(187, 210, 14, 1),

      ),

      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.teal[100 * (index % 9 + 1)],

                  child: Column(

                    children: [

                      Icon(Icons.access_alarm, size: 15),

                      Text('Grid Item $index')

                    ],

                  ),

                );

              },

              childCount: 20,

            ),

          ),

        ],

      ),

    );

  }

}
```

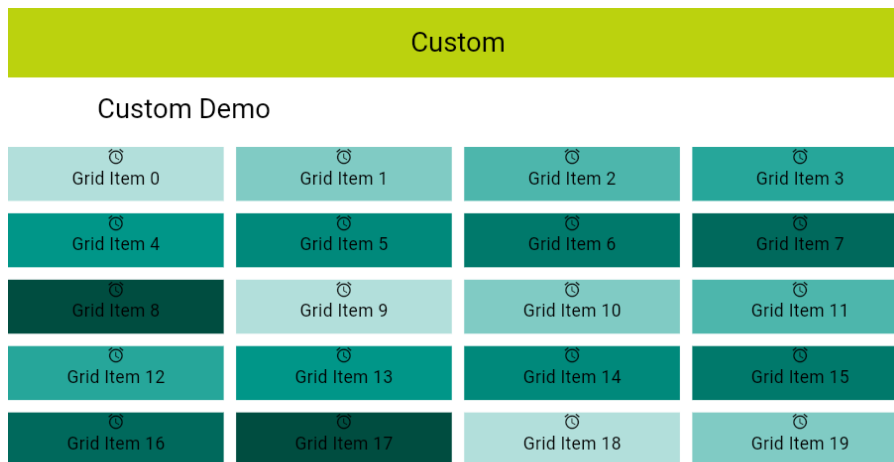


```

        gridDelegate: const SliverGridDelegateWithMaxCrossAxisExtent(
          maxCrossAxisExtent: 200,
          mainAxisSpacing: 10,
          crossAxisSpacing: 10,
          childAspectRatio: 4,
        ),
      ),
    // Sliver Fixed List
  ],
),
);
}
}

```

Screenshoot :



Deskripsi :

- **SliverAppBar:**

- Sebuah **SliverAppBar** yang dipasang secara permanen di bagian atas layar. Judul "Custom Demo" ditampilkan di dalamnya, memberikan tampilan yang menarik saat pengguna menggulir konten.

- **SliverGrid:**

- Sebuah **SliverGrid** yang menampilkan 20 item grid. Setiap item terdiri dari ikon alarm dan teks yang menunjukkan nomor item. Warna latar belakang item berubah berdasarkan indeksinya, memberikan efek visual yang dinamis. Grid ini dirancang untuk menyesuaikan diri dengan ukuran layar, dengan pengaturan untuk spasi antar item.

B. UNGUIDED

Modifikasi project Rekomendasi Wisata pada Tugas Unguided 04 modul Antarmuka Pengguna dengan mengimplementasikan widget CustomScrollView, SliverAppBar, dan SliverList untuk merekomendasikan beberapa tempat wisata yang ada di Banyumas disertai foto, nama wisata, dan deskripsi singkat! (buatlah se kreatif mungkin

Code :

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

void main() {
  SystemChrome.setSystemUIOverlayStyle(const SystemUiOverlayStyle(
    statusBarColor: Colors.transparent,
    statusBarIconBrightness: Brightness.light,
  ));
  runApp(const WisataBanyumasApp());
}

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Wisata Banyumas',
      theme: ThemeData(
        fontFamily: 'Poppins',
        primaryColor: const Color(0xFF07375F),
        colorScheme: ColorScheme.fromSwatch()
          .copyWith(secondary: const Color(0xFF4E9F3D)),
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      home: const WisataListScreen(),
    );
  }
}

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

```

class WisataListScreen extends StatelessWidget {

  const WisataListScreen({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 Banyumas',

                style: TextStyle(

                  fontSize: 24,

                  fontWeight: FontWeight.bold,

                  color: Colors.white,

                  shadows: [

                    Shadow(

                      blurRadius: 10.0,

                      color: Colors.black,

                      offset: Offset(2, 2),

                    ),

                  ],

                ),

              ),

              background: Stack(

                fit: StackFit.expand,

                children: [

                  Image.network(

                    'https://www.shutterstock.com/image-illustration/abstract-white-pattern-waves-texture-260nw-2463089043.jpg',

                    fit: BoxFit.cover,

                  ),

                  Container(

                    decoration: BoxDecoration(

                      gradient: LinearGradient(

                        colors: [

                          Colors.black.withOpacity(0.6),

                          Colors.transparent,

                        ],

                        begin: Alignment.bottomCenter,

                        end: Alignment.topCenter,

```

```

        ),
    ),
    ),
    ],
    ),
    ),
    ),
    SliverList(
        delegate: SliverChildBuilderDelegate(
            (BuildContext context, int index) {
                return WisataCard(wisata: wisataList[index]);
            },
            childCount: wisataList.length,
        ),
    ),
    ],
    ),
    );
}

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

    Wisata({
        required this.name,
        required this.description,
        required this.imageUrl,
    });
}

final List<Wisata> wisataList = [
    Wisata(
        name: 'Baturraden',
        description:
            'Kawasan wisata alam pegunungan dengan pemandian air panas, air terjun, dan hutan pinus yang sejuk.',
        imageUrl:
            'https://upload.wikimedia.org/wikipedia/commons/thumb/7/7f/Baturraden_overview_from_ridge%2C_Purwokerto%2C_2015-03-23.jpg/600px-Baturraden_overview_from_ridge%2C_Purwokerto%2C_2015-03-23.jpg',
    ),

```

```

        Wisata(
            name: 'Telaga Sunyi',
            description:
                'Danau tersembunyi di tengah hutan dengan air yang jernih, ideal untuk camping dan trekking.',
            imageUrl:
                'https://static.promediateknologi.id/crop/0x538:960x1191/0x0/webp/photo/p2/74/2024/06/30/IMG-20240630-WA0005-1398012367.jpg',
        ),
        Wisata(
            name: 'Curug Cipendok',
            description:
                'Air terjun spektakuler dengan ketinggian 92 meter, dikelilingi hutan tropis yang asri.',
            imageUrl:
                'https://static.promediateknologi.id/crop/0x0:0x0/750x500/webp/photo/p1/593/2024/07/09/curug_cipendok-1298016609.jpg',
        ),
        Wisata(
            name: 'Small World Purwokerto',
            description:
                'Taman miniatur dengan replika landmark dunia, cocok untuk edukasi dan foto-foto.',
            imageUrl:
                'https://www.lamudi.co.id/journal/wp-content/uploads/2023/04/1-3.jpg',
        ),
        Wisata(
            name: 'Owabong Waterpark',
            description:
                'Taman rekreasi air terbesar di Purbalingga dengan berbagai wahana seru untuk segala usia.',
            imageUrl:
                'https://static.promediateknologi.id/crop/0x0:0x0/750x500/webp/photo/p1/09/2024/04/09/at-raksi-flyboard-di-owabong-purbalingga-2408471420.jpg',
        ),
    ];

class WisataCard extends StatelessWidget {
    final Wisata wisata;

    const WisataCard({Key? key, required this.wisata}) : super(key: key);

    @override
    Widget build(BuildContext context) {
        return Card(
            margin: const EdgeInsets.all(8.0),
            elevation: 4.0,
            shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12.0)),
            child: Column(
                crossAxisAlignment: CrossAxisAlignment.stretch,
                children: <Widget>[

```

```

Hero(
  tag: wisata.name,
  child: ClipRRect(
    borderRadius:
      const BorderRadius.vertical(top: Radius.circular(12.0)),
    child: Image.network(
      wisata.imageUrl,
      height: 200,
      fit: BoxFit.cover,
    ),
  ),
),
Padding(
  padding: const EdgeInsets.all(16.0),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: <Widget>[
      Text(
        wisata.name,
        style: const TextStyle(
          fontSize: 22,
          fontWeight: FontWeight.bold,
          color: Color.fromARGB(255, 30, 33, 81),
        ),
      ),
      const SizedBox(height: 8.0),
      Text(
        wisata.description,
        style: TextStyle(
          fontSize: 16,
          color: Colors.grey[600],
        ),
      ),
      Row(
        mainAxisAlignment: MainAxisAlignment.end,
        children: [
          ElevatedButton(
            onPressed: () {
              Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (context) =>
                    WisataDetailScreen(wisata: wisata),
                ),
              );
            },
          ),
        ],
      ),
    ],
  ),
),

```

```

        style: ElevatedButton.styleFrom(
          backgroundColor: const Color(0xFF07375F),
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(20),
          ),
        ),
        child: const Text(
          'Lihat Detail',
          style: TextStyle(color: Colors.white),
        ),
      ),
    ],
  ),
],
);
}
}

class WisataDetailScreen extends StatelessWidget {
  final Wisata wisata;

  const WisataDetailScreen({Key? key, required this.wisata}) : super(key: key);

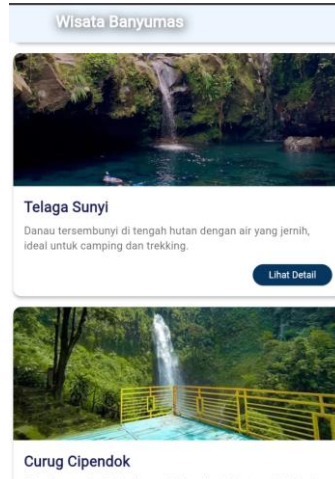
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(wisata.name),
      ),
      body: Column(
        children: [
          Hero(
            tag: wisata.name,
            child: Image.network(
              wisata.imageUrl,
              fit: BoxFit.cover,
            ),
          ),
          Padding(
            padding: const EdgeInsets.all(16.0),
            child: Text(
              wisata.description,
              style: const TextStyle(fontSize: 18),
            ),
          ),
        ],
      ),
    );
  }
}

```

Screenshoot Output



(tampilan awal)



(saat di scroll)



(saat klik lihat detail)

Deskripsi Program

Aplikasi ini adalah platform rekomendasi tempat wisata di Banyumas, yang dibangun menggunakan Flutter. Fitur utama dari aplikasi ini meliputi:

1. Halaman Utama:

- Halaman utama menampilkan daftar tempat wisata dengan menggunakan widget **CustomScrollView**. Di bagian atas, terdapat **SliverAppBar** yang menampilkan judul "Wisata Banyumas" dan gambar latar belakang dengan efek gradien.

2. Daftar Wisata:

- Daftar tempat wisata ditampilkan dalam bentuk **SliverList**, di mana setiap item adalah kartu wisata yang berisi gambar, nama, dan deskripsi singkat. Setiap kartu dirancang dengan efek bayangan untuk tampilan yang lebih menarik.

3. Detail Wisata:

- Setiap kartu memiliki tombol "Lihat Detail" yang membawa pengguna ke layar detail. Layar detail menampilkan gambar dan deskripsi lengkap tentang tempat wisata yang dipilih. Animasi transisi menggunakan **Hero** widget untuk memberikan pengalaman visual yang halus saat berpindah antar layar.

4. Desain yang Responsif:

- Aplikasi menggunakan tema yang konsisten dengan warna yang menarik dan font yang mudah dibaca, menjadikan pengalaman pengguna lebih nyaman.

Secara keseluruhan, aplikasi ini menawarkan antarmuka yang ramah pengguna dan menyoroti keindahan tempat wisata di Banyumas dengan cara yang interaktif dan menarik.

BAB III

KESIMPULAN DAN SARAN

A. KESIMPULAN

Modul Antarmuka Pengguna Lanjutan ini memberikan wawasan mendalam mengenai teknik dan komponen desain yang esensial dalam pengembangan aplikasi menggunakan Flutter. Melalui penggunaan widget lanjutan seperti **CustomScrollView**, **SliverAppBar**, dan **SliverList**, pengembang dapat menciptakan antarmuka yang responsif dan menarik, yang tidak hanya meningkatkan pengalaman pengguna tetapi juga mempermudah navigasi dalam aplikasi.

Penggunaan warna, tipografi, dan elemen interaktif secara efektif berkontribusi pada penciptaan desain yang estetis dan fungsional. Selain itu, modul ini mendorong pengembang untuk memahami pentingnya interaksi yang intuitif, sehingga pengguna dapat menjelajahi aplikasi dengan mudah dan nyaman.

Dengan tujuan untuk meningkatkan keterampilan pengembang, modul ini menyediakan pendekatan praktis yang dapat langsung diterapkan dalam proyek nyata. Diharapkan, pengembang dapat menerapkan pengetahuan ini untuk menciptakan aplikasi yang tidak hanya menarik secara visual tetapi juga memberikan pengalaman pengguna yang optimal.

Secara keseluruhan, pemahaman dan penerapan teknik antarmuka pengguna yang baik sangat penting dalam menciptakan aplikasi yang berhasil, dan modul ini menjadi langkah awal yang signifikan bagi pengembang dalam perjalanan mereka.