

LAPORAN PRAKTIKUM
MOBILE & WEB SERVICE PRAKTIK

Pertemuan ke-2

Semester Ganjil TA. 2024/2025

Flutter Widget

Dosen Pengampu: Suyud Widiono, S.Pd., M.Kom.



Disusun oleh :

Regita Cahya Arrahma (5220411359)

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2024**

DAFTAR ISI

DAFTAR ISI	i
BAB 1 FLUTTER WIDGET	1
1.1. Dasar Teori	1
1.1.1. Flutter Widget	1
1.2. Praktikum	3
1.2.1. MaterialApp	3
1.2.2. Scaffold	4
1.2.3. Center.....	6
1.2.4. SizedBox	8
1.2.5. Text	11
1.2.6. Expanded.....	11
1.2.7. Container.....	16
1.2.8. Row dan Column	18
DAFTAR PUSTAKA	27

BAB 1

FLUTTER WIDGET

1.1. Dasar Teori

1.1.1. Flutter Widget

a. Pengertian

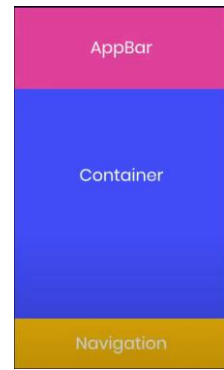
Widget dalam Flutter adalah komponen dasar yang sangat penting untuk mengembangkan aplikasi. Fungsi utama widget adalah mengatur dan mengendalikan tampilan *user interface* (UI). Setiap widget memiliki fungsi tersendiri dan dapat digabungkan dengan widget lain untuk menciptakan tampilan UI yang lebih kompleks. Contohnya, padding dan margin juga merupakan jenis widget yang dapat digunakan untuk mengatur jarak antar elemen dalam UI, sedangkan tema adalah widget yang membantu dalam menerapkan gaya dan konsistensi di seluruh aplikasi. Dengan menggunakan widget, pengembang dapat dengan mudah menentukan posisi, ukuran, dan tampilan elemen UI sesuai dengan interaksi pengguna, sehingga pengalaman pengguna menjadi lebih dinamis dan responsif.

Selain itu, kombinasi berbagai widget memberikan fleksibilitas yang lebih dalam desain dan fungsi aplikasi. Pengembang dapat memilih dari berbagai jenis widget, baik yang statis maupun yang dinamis, untuk memenuhi kebutuhan spesifik aplikasi yang mereka buat. Kemampuan untuk menyesuaikan tampilan UI dengan kombinasi widget yang berbeda memberikan kebebasan kreatif bagi pengembang. Dengan demikian, penggunaan widget dalam Flutter tidak hanya mempercepat proses pengembangan, tetapi juga meningkatkan kualitas dan daya tarik visual aplikasi yang dihasilkan.

b. Istilah Widget pada Flutter.

Flutter menawarkan banyak sekali widget bawaan, dan berikut adalah beberapa di antaranya.

- **Container:** berfungsi sebagai wadah untuk menyimpan atau mengelola widget-widget lainnya. Didalam Container terdapat elemen AppBar dan Navigation. AppBar adalah bagian atas dari aplikasi untuk menampilkan judul dan kontrol navigasi dasar, sedangkan Navigation digunakan untuk berpindah antar halaman atau bagian aplikasi lainnya.
- **MaterialApp:** widget tingkat atas yang mengonfigurasi aplikasi untuk menggunakan tema material design.
- **Scaffold:** widget layout dasar yang sangat berguna untuk membuat tampilan dasar aplikasi dengan cepat dan konsisten.
- **Stateless:** digunakan untuk menampilkan elemen-elemen yang sifatnya statis, artinya setelah data ditampilkan, data tersebut tidak akan berubah lagi selama aplikasi berjalan.
- **Statefull:** digunakan untuk menampilkan data-data yang dinamis atau data yang kita telah tampilkan sewaktu-waktu dapat mengalami perubahan.
- **SafeArea:** digunakan untuk memastikan bahwa konten aplikasi tidak terpotong atau tertimpa oleh area tampilan yang biasanya ditempati oleh elemen-elemen sistem seperti status bar, *notch* (potongan layar), tombol navigasi, atau bilah lainnya pada perangkat.
- **Layout:** digunakan untuk mengatur tata letak atau posisi dari widget-widget lain di dalam antarmuka aplikasi. Widget ini menentukan bagaimana widget anak (*child*) akan disusun, apakah secara horizontal, vertikal, *grid*, atau lainnya.
- **Interactive:** digunakan untuk mengelola interaksi pengguna dengan aplikasi, seperti tombol, slider, atau checkbox. Widget ini biasanya merespons input pengguna (*tap*, *click*, *drag*, dll.) dan memicu tindakan tertentu.
- **Display:** digunakan untuk menampilkan teks atau informasi lainnya yang bersifat statis atau dinamis. Widget ini termasuk yang digunakan untuk mendeskripsikan informasi kepada pengguna, seperti teks, ikon, atau widget visual lain.



- **Animation:** digunakan untuk menambahkan animasi atau transisi visual ke dalam antarmuka aplikasi, baik untuk efek yang halus atau transisi antara status berbeda.
- **Input:** digunakan untuk menerima input dari pengguna, seperti memasukkan teks, memilih tanggal, atau input numerik. Widget ini berinteraksi dengan pengguna untuk mengumpulkan data.
- **Center:** digunakan untuk menempatkan widget *child* di posisi tengah relatif terhadap dirinya. Widget ini sering digunakan untuk memusatkan elemen di tengah layar atau dalam widget lain.
- **SizedBox:** digunakan untuk membuat kotak (*box*) dengan ukuran tetap atau memberikan jarak antara widget-widget *child*-nya
- **Text:** digunakan untuk menampilkan teks di layar.
- **Expanded:** widget yang memperluas widget *child*-nya untuk mengisi ruang yang tersedia di dalam parent yang fleksibel, biasanya digunakan dalam *row*, *column*, atau *flex*.
- **Row:** digunakan untuk menyusun widget *child* dalam baris atau kolom secara horizontal.
- **Column:** digunakan untuk Menyusun widget *child* dalam baris atau kolom secara vertical.
- **Nested:** sebuah widget ditempatkan di dalam widget lain. Dalam hal ini, satu widget menjadi parent atau induk dari widget lain yang disebut *child*.
- **Image:** digunakan untuk menampilkan gambar di aplikasi. Flutter menyediakan widget khusus untuk menampilkan gambar baik dari jaringan (*online*), file lokal, atau asset bawaan.

1.2. Praktikum

1.2.1. MaterialApp

Kode program penggunaan MaterialApp:

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Quick Note',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const MyHomePage(),
    );
  }
}
```

Pada kode program di atas, MaterialApp mengonfigurasi aplikasi dengan sebuah judul, tema, dan widget awal (MyHomePage). home menentukan widget awal yang akan ditampilkan saat aplikasi diluncurkan, dalam hal ini adalah MyHomePage.

1.2.2. Scaffold

Kode program penggunaan Scaffold:

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Quick Note',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const MyHomePage(),
    );
  }
}

class MyHomePage extends StatefulWidget {
```

```

const MyHomePage({super.key});

@override
State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'Home Page',
          style: TextStyle(fontSize: 16.0, fontWeight: FontWeight.w600),
        ),
      ),
      body: const SafeArea(
        child: Center(
          child: Text('Hello World!'),
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: () => log('Hello World'),
        child: const Icon(Icons.add_rounded),
      ),
    );
  }
}

```

Pada kode program di atas, Scaffold menyediakan struktur dasar halaman dengan AppBar di bagian atas FloatingActionButton disudut kanan bawah. body adalah bagian utama dari halaman yang berisi konten utama, dalam hal ini sebuah teks ditengah layar.

Berikut hasil *running* dari kode program:



1.2.3. Center

Kode program penggunaan Center:

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Quick Note',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const CenterPage(),
    );
  }
}
```



```

class CenterPage extends StatefulWidget {
  const CenterPage({super.key});

  @override
  State<CenterPage> createState() => _CenterPageState();
}

class _CenterPageState extends State<CenterPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'Center',
          style: TextStyle(fontSize: 16.0, fontWeight: FontWeight.w600),
        ),
      ),
      body: SafeArea(
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.min,
            children: [
              const Text('This text is in the center of the screen'),
              const SizedBox(
                height: 16.0,
              ),
              Container(
                width: 200.0,
                height: 100.0,
                color: Colors.black,
                child: const Text(
                  'Text without center',
                  style: TextStyle(color: Colors.white),
                ),
              ),
              const SizedBox(
                height: 16.0,
              ),
              Container(
                width: 200.0,
                height: 100.0,
                color: Colors.pink,
                child: const Center(
                  child: Text(
                    'Text with center',
                    style: TextStyle(color: Colors.white),
                  ),
                ),
              ),
            ],
          ),
        ),
      ),
    );
  }
}

```

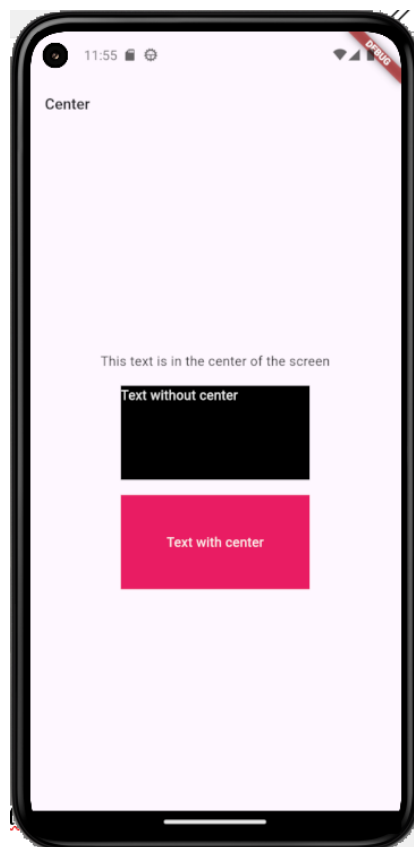
```

    ),
  ),
],
),
)),
);
}
}

```

Pada kode program diatas, fungsi `child` pada widget `Center` adalah untuk memusatkan widget di dalamnya, baik secara horizontal maupun vertikal, dalam ruang yang tersedia.

Berikut hasil *running* dari kode program:



1.2.4. SizedBox

Kode program penggunaan `SizedBox`:

```

import 'package:flutter/material.dart';

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

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

```

```

@override
Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Quick Note',
    theme: ThemeData(
      colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
      useMaterial3: true,
    ),
    home: const SizedboxPage(),
  );
}

class SizedboxPage extends StatefulWidget {
  const SizedboxPage({super.key});

  @override
  State<SizedboxPage> createState() => _SizedboxPageState();
}

class _SizedboxPageState extends State<SizedboxPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'SizedBox',
          style: TextStyle(fontSize: 16.0, fontWeight: FontWeight.w600),
        ),
      ),
      body: SafeArea(
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              const Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  Text('This text and'),
                  Text('the next text have no distance'),
                ],
              ),
              const SizedBox(
                height: 16.0,
              ),
              const Row(
                mainAxisAlignment: MainAxisAlignment.center,

```

```

        children: [
          Text('This text end'),
          SizedBox(
            width: 24.0,
          ),
          Text('The next text have distance'),
        ],
      ),
      const SizedBox(
        height: 16.0,
      ),
      Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Container(
            width: 50.0,
            height: 50.0,
            color: Colors.black,
          ),
          Container(
            width: 50.0,
            height: 50.0,
            color: Colors.pink,
          )
        ],
      ),
      const SizedBox(
        height: 16.0,
      ),
      Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Container(
            width: 50.0,
            height: 50.0,
            color: Colors.black,
          ),
          const SizedBox(
            width: 24.0,
          ),
          Container(
            width: 50.0,
            height: 50.0,
            color: Colors.pink,
          )
        ],
      ),
    ],
  ],

```

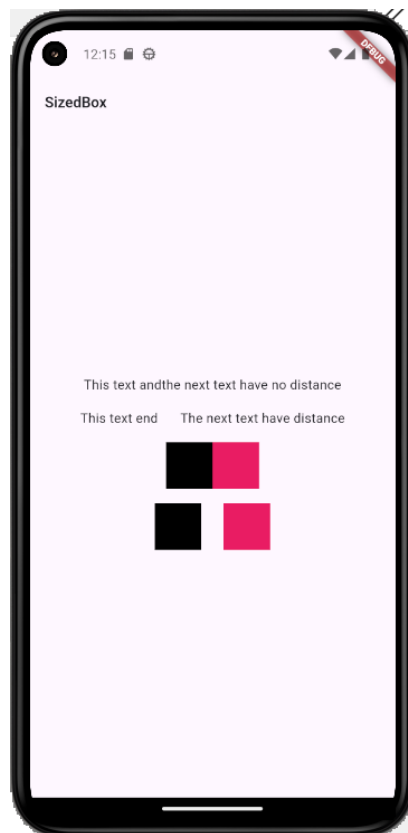
```

    ),
  )),
);
}
}

```

Pada kode program diatas, `SizeBox` digunakan untuk memberikan jarak (*spacing*) antara elemen-elemen dalam *user interface*. Properti utamanya adalah `width` dan `height`. `width` digunakan untuk menentukan lebar dari `SizeBox`, sedangkan `height` digunakan untuk menentukan tinggi dari `SizeBox`.

Berikut adalah hasil *running* dari kode program:



1.2.5. Text

Kode program penggunaan `Text`:

```

import 'package:flutter/material.dart';

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

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

```

```

@override
Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Quick Note',
    theme: ThemeData(
      colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
      useMaterial3: true,
    ),
    home: const TextPage(),
  );
}

class TextPage extends StatefulWidget {
  const TextPage({super.key});

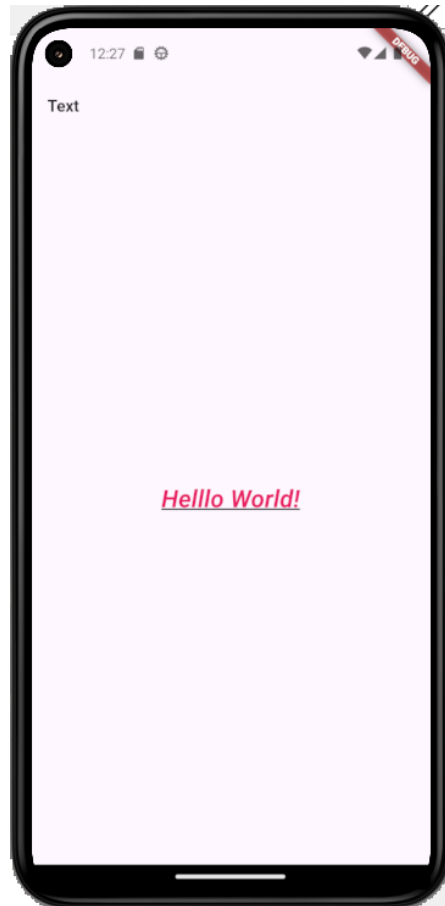
  @override
  State<TextPage> createState() => _TextPageState();
}

class _TextPageState extends State<TextPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'Text',
          style: TextStyle(fontSize: 16.0, fontWeight: FontWeight.w600),
        ),
      ),
      body: const SafeArea(
        child: Center(
          child: Text(
            'Hello World!',
            style: TextStyle(
              fontSize: 24.0,
              fontWeight: FontWeight.w600,
              color: Colors.pink,
              decoration: TextDecoration.underline,
              fontStyle: FontStyle.italic,
            ),
          ),
        ),
      ),
    );
  }
}

```

Pada kode program diatas, widget Text digunakan untuk menampilkan tulisan 'Hello World!' di tengah layer yang diatur tampilannya dengan menggunakan TextStyle.

Berikut adalah hasil *running* dari kode program:



1.2.6. Expanded

Contoh penggunaan Expanded:

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Quick Note',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
```

```

        useMaterial3: true,
      ),
      home: const ExpandedPage(),
    );
  }
}

class ExpandedPage extends StatefulWidget {
  const ExpandedPage({super.key});

  @override
  State<ExpandedPage> createState() => _ExpandedPageState();
}

class _ExpandedPageState extends State<ExpandedPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'Expanded',
          style: TextStyle(fontSize: 16.0, fontWeight: FontWeight.w600),
        ),
      ),
      body: SafeArea(
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              const Text("The container below don't use expansion"),
              const SizedBox(
                height: 16.0,
              ),
              Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  Container(
                    width: 50.0,
                    height: 50.0,
                    color: Colors.black,
                  ),
                  const SizedBox(
                    width: 16.0,
                  ),
                  Container(
                    width: 50.0,
                    height: 50.0,
                    color: Colors.pink,

```

1


```

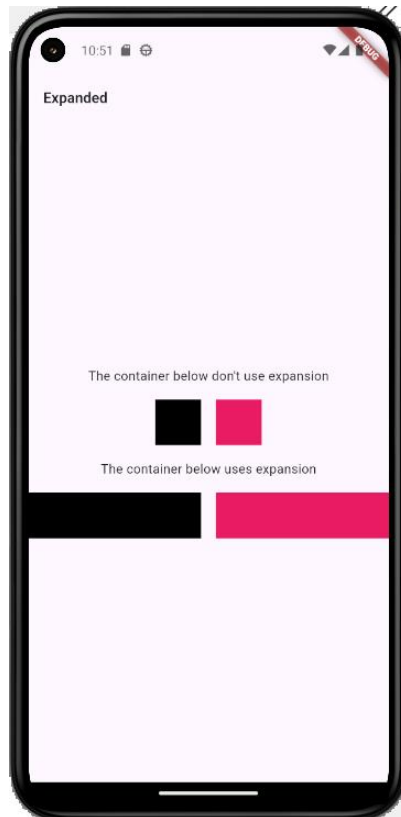
    ),
  ],
),
const SizedBox(
  height: 16.0,
),
const Text('The container below uses expansion'),
const SizedBox(
  height: 16.0,
),
Row(
  children: [
    Expanded(
      child: Container(
        width: 50.0,
        height: 50.0,
        color: Colors.black,
      ),
    ),
    const SizedBox(
      width: 16.0,
    ),
    Expanded(
      child: Container(
        width: 50.0,
        height: 50.0,
        color: Colors.pink,
      ),
    ),
  ],
),
],
),
)),
);
}
}

```

Pada kode program di atas, bagian pertama (di luar penggunaan Expanded), dua Container memiliki lebar tetap (50.0), dan karena tidak menggunakan Expanded, mereka akan menempati ruang sesuai dengan ukuran yang ditetapkan secara eksplisit. Lebar keduanya adalah tetap meskipun terdapat ruang kosong yang tersisa di Row. Pada bagian kedua (dengan Expanded), dua Container dimasukkan ke dalam widget Expanded. Fungsi Expanded adalah membuat widget *child* (dalam hal ini Container)

mengambil semua ruang yang tersedia di dalam Row, setelah memperhitungkan widget lain seperti `SizedBox` yang digunakan untuk memberi jarak.

Berikut adalah hasil *running* dari kode program:



1.2.7. Container

Kode program penggunaan Container:

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Quick Note',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const ContainerPage(),
    );
  }
}
```

```

    );
  }
}

class ContainerPage extends StatefulWidget {
  const ContainerPage({super.key});

  @override
  State<ContainerPage> createState() => _ContainerPageState();
}

class _ContainerPageState extends State<ContainerPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'Container',
          style: TextStyle(fontSize: 16.0, fontWeight: FontWeight.w600),
        ),
      ),
      body: SafeArea(
        child: Center(
          child: Container(
            width: 200.0,
            height: 100.0,
            decoration: BoxDecoration(
              color: Colors.pink.shade50,
              borderRadius: BorderRadius.circular(8.0),
              border: Border.all(color: Colors.pink.shade300, width: 1.0),
              boxShadow: const [
                BoxShadow(
                  color: Colors.black12,
                  blurRadius: 16.0,
                  spreadRadius: 4.0,
                  offset: Offset(0.0, 4.0))
              ],
            ),
            child: const Center(
              child: Text(
                'Hello World!',
                style: TextStyle(
                  fontSize: 24.0,
                  fontWeight: FontWeight.w600,
                  color: Colors.pink,
                ),
              ),
            ),
          ),
        ),
      ),
    );
  }
}

```

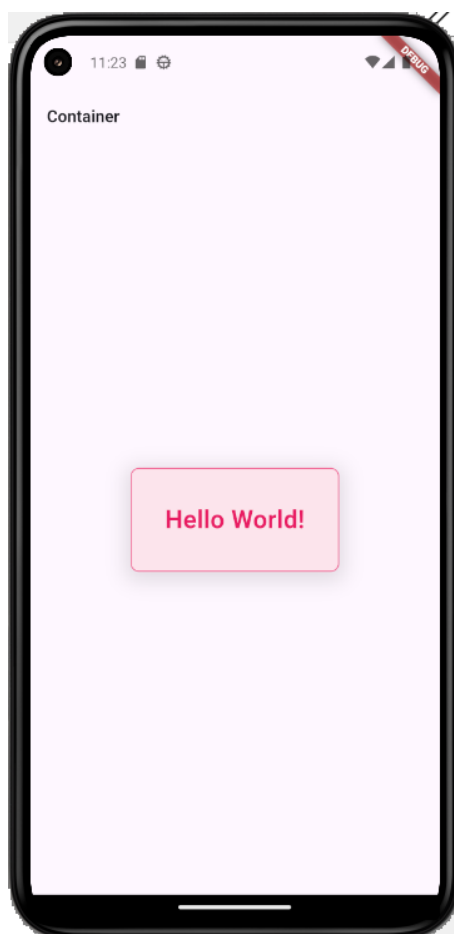
```

    ),
  )),
);
}
}

```

Pada kode program di atas, Container berfungsi mengatur tampilan dan tata letak widget teks. Dengan lebar 200 piksel dan tinggi 100 piksel, Container memiliki latar belakang merah muda, sudut melengkung, garis batas, serta bayangan untuk memberikan efek visual. Teks 'Hello World!' ditempatkan di tengah Container dengan menggunakan Center, menciptakan tata letak yang rapi dan estetis.

Berikut adalah hasil *running* dari kode program:



1.2.8. Row dan Column

a. Row

Kode program penggunaan Row:

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

```

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Quick Note',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor:
Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const RowPage(),
    );
  }
}

class RowPage extends StatefulWidget {
  const RowPage({super.key});

  @override
  State<RowPage> createState() => _RowPageState();
}

class _RowPageState extends State<RowPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'Row',
          style: TextStyle(fontSize: 16.0, fontWeight:
FontWeight.w600),
        ),
      ),
      body: SafeArea(
        child: Center(
          child: Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Container(
                width: 50.0,
                height: 50.0,
                color: Colors.black,

```

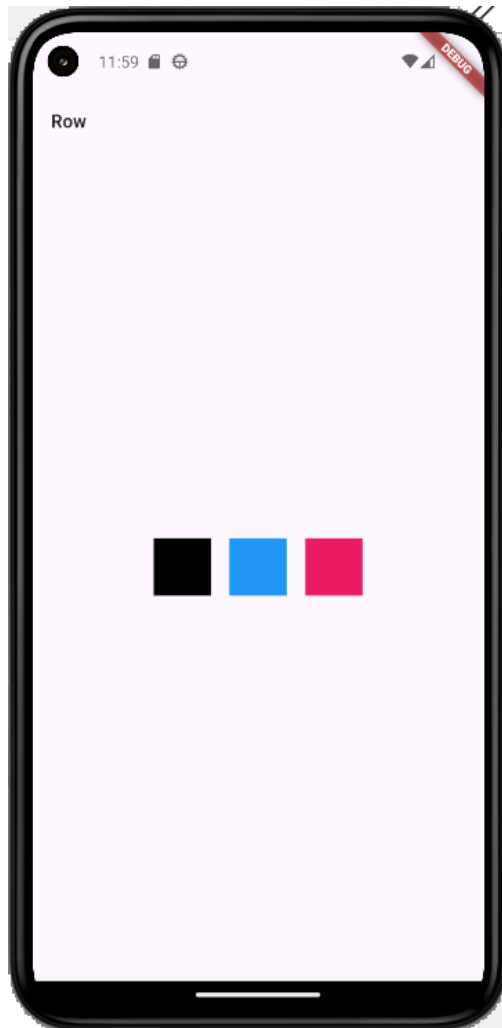
```

    ),
    const SizedBox(
      width: 16.0,
    ),
    Container(
      width: 50.0,
      height: 50.0,
      color: Colors.blue,
    ),
    const SizedBox(
      width: 16.0,
    ),
    Container(
      width: 50.0,
      height: 50.0,
      color: Colors.pink,
    ),
  ],
),
),
),
);
}
}

```

Pada kode program di atas, Pada kode tersebut, Row berfungsi untuk menyusun beberapa widget Container secara horizontal dalam satu baris. Di dalam Row, ada tiga Container dengan ukuran 50x50 piksel yang masing-masing diberi warna hitam, biru, dan merah muda. `mainAxisAlignment: MainAxisAlignment.center` digunakan untuk memusatkan baris di bagian tengah layar. `SizedBox` dengan lebar 16 piksel ditambahkan di antara Container untuk memberikan jarak antar elemen. Dengan demikian, Row mengatur layout elemen secara horizontal dan menjaga jarak antar elemen dengan rapi.

Berikut adalah hasil *running* dari kode program:



b. Column

Kode program penggunaan Column:

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Quick Note',
      theme: ThemeData(
```

```

        colorScheme: ColorScheme.fromSeed(seedColor:
Colors.deepPurple),
        useMaterial3: true,
    ),
    home: const ColumnPage(),
);
}
}

class ColumnPage extends StatefulWidget {
  const ColumnPage({super.key});

  @override
  State<ColumnPage> createState() => _ColumnPageState();
}

class _ColumnPageState extends State<ColumnPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'Column',
          style: TextStyle(fontSize: 16.0, fontWeight:
FontWeight.w600),
        ),
      ),
      body: SafeArea(
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Container(
                width: 100.0,
                height: 100.0,
                color: Colors.black,
              ),
              const SizedBox(height: 16.0),
              Container(
                width: 100.0,
                height: 100.0,
                color: Colors.blue,
              ),
              const SizedBox(
                height: 16.0,
              ),
              Container(
                width: 100.0,

```



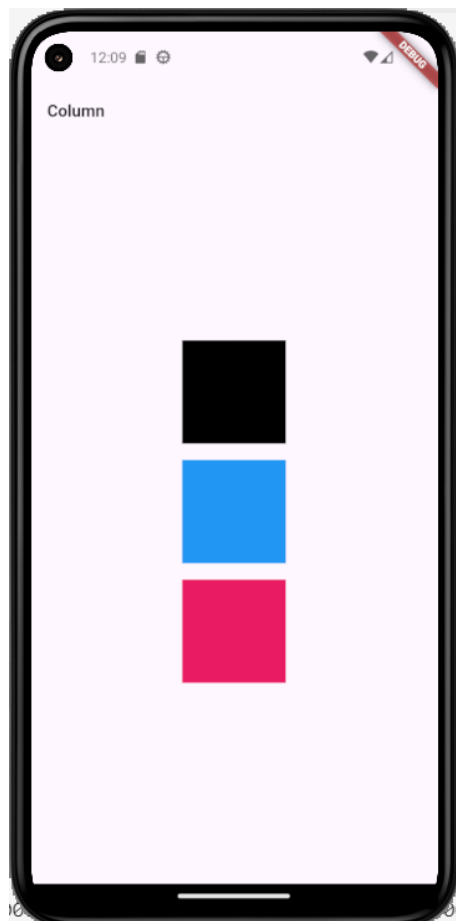
```

        height: 100.0,
        color: Colors.pink,
      ),
    ],
  )),
),
);
}
}

```

Pada kode program diatas, widget Column digunakan untuk menampilkan beberapa widget *child* dalam susunan vertikal. Di dalam widget Column, tiga Container dengan ukuran dan warna berbeda ditampilkan satu per satu dari atas ke bawah. `mainAxisAlignment: MainAxisAlignment.center` pada Column mengatur agar semua widget *child* ditempatkan di tengah vertikal layar. `SizedBox` digunakan untuk memberikan jarak antar widget Container. Dengan menggunakan Column, widget ini memungkinkan elemen-elemen untuk disusun secara terorganisir dalam satu kolom tanpa tumpang tindih.

Berikut hasil *running* dari kode program:



c. Penggabungan Row dan Column

Kode program penggunaan penggabungan Row dan Column:

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Quick Note',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor:
Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const ColumnRowPage(),
    );
  }
}

class ColumnRowPage extends StatefulWidget {
  const ColumnRowPage({super.key});

  @override
  State<ColumnRowPage> createState() => _ColumnRowPageState();
}

class _ColumnRowPageState extends State<ColumnRowPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'Column & Row',
          style: TextStyle(fontSize: 16.0, fontWeight:
FontWeight.w600),
        ),
      ),
      body: SafeArea(
        child: Center(
```

```

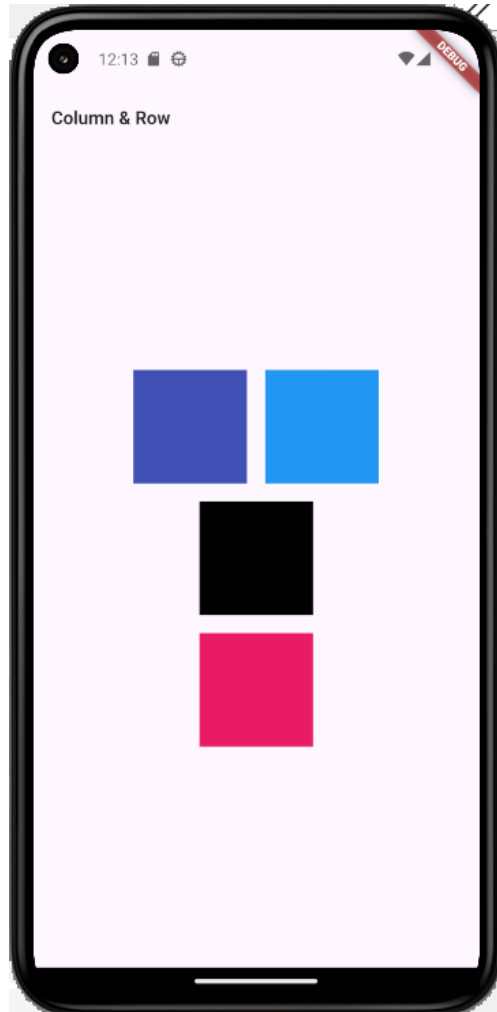
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Row(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Container(
                  width: 100.0,
                  height: 100.0,
                  color: Colors.indigo,
                ),
                const SizedBox(
                  width: 16.0,
                ),
                Container(
                  width: 100.0,
                  height: 100.0,
                  color: Colors.blue,
                ),
              ],
            ),
            const SizedBox(
              height: 16.0,
            ),
            Container(
              width: 100.0,
              height: 100.0,
              color: Colors.black,
            ),
            const SizedBox(
              height: 16.0,
            ),
            Container(
              width: 100.0,
              height: 100.0,
              color: Colors.pink,
            ),
          ],
        )),
      ),
    );
  }
}

```

Pada kode program di atas, terdapat kombinasi penggunaan widget Column dan Row untuk mengatur tata letak widget secara vertikal dan horizontal. di dalam widget Column, ada widget Row yang menampilkan dua

Container berwarna indigo dan blue di samping satu sama lain secara horizontal. Di bawah Row tersebut, terdapat dua Container lagi (berwarna black dan pink) yang ditampilkan secara vertikal dengan jarak yang diatur oleh `SizeBox`.

Berikut hasil *running* dari kode program:



DAFTAR PUSTAKA

(N.d.). Diakses dari

<https://dspace.uui.ac.id/bitstream/handle/123456789/20790/05.3%20Bab%203.pdf?sequence=7&isAllowed=y>

(N.d.). Diakses dari

https://www.scribd.com/embeds/721139002/content?start_page=1&view_mode=scroll&access_key=key-fFexxf7r1bzEfWu3HKwf

BuildWithAngga. (n.d.). Belajar Perkenalan Flutter. Diakses dari

<https://class.buildwithangga.com/course-playing/vAqOvE1pmJ/1>

Akbar, S. (2020). Belajar Widget-widget Pada Flutter : Flutter Starter Pack Part 1. Diakses

dari <https://medium.com/flutter-developer-indonesia/belajar-widget-widget-pada-flutter-flutter-starter-pack-part-1-7f386a02fbb6>

Anggara, A., & Pratama, D. (2024). *Modul Praktikum Mobile & Web Service*. Yogyakarta: Universitas Teknologi Yogyakarta.