

Pertemuan II – Widget *Row* dan *Column*

2.1. Ringkasan

Komponen	Uraian
Tujuan Praktikum	<ol style="list-style-type: none">1. Memahami konsep widget <i>Row</i> dan <i>Column</i> dalam Flutter.2. Membuat dan menggunakan widget kustom (<i>custom widget</i>).3. Mengatur tata letak widget secara horizontal dan vertikal.4. Menampilkan widget menggunakan properti visual seperti warna, border, dan icon.
Dasar Teori	<ol style="list-style-type: none">1. <i>Row</i> dan <i>Column</i> adalah widget layout yang digunakan untuk mengelompokkan dan mengatur beberapa widget anak (<i>children</i>).2. <i>Row</i> mengatur widget secara horisontal (kiri ke kanan).3. <i>Column</i> mengatur widget secara vertikal (atas ke bawah).4. Widget di dalam <i>Row</i> atau <i>Column</i> disebut <i>children</i> (anak), sedangkan <i>Row</i> dan <i>Column</i> disebut <i>parent</i> (induk).5. <i>Row</i> dan <i>Column</i> termasuk widget <i>invisible</i> artinya tidak terlihat oleh pengguna secara visual tetapi berfungsi sebagai pengaturan layout.
Alat & Bahan	<ol style="list-style-type: none">1. Flutter SDK telah terpasang dan IDE siap digunakan (Visual Studio Code atau Android Studio).2. <i>Emulator</i> atau <i>smartphone</i> Android/iOS tersedia dan siap digunakan untuk menjalankan aplikasi.3. Proyek baru dengan template awal sudah dibuat dan siap dimodifikasi.

2.2. Langkah Kerja

Langkah 1 - Membuat Proyek Baru

- a. Buat proyek baru dengan nama *row_and_column*.
- b. Ubah kode program awal agar properti *title* pada *MaterialApp* menjadi “*Row and Column*”.
- c. Ubah *AppBar* title menjadi “*Row and Column*”.

Kode program:

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10    return MaterialApp(
11      title: 'Row and Column',
12      home: Scaffold(
13        appBar: AppBar(
14          title: Text('Row and Column'),
15        ),
16        body: Center(
17          child: Text('Hello, Flutter!'),
18        ),
19      ),
20    );
21  }
22 }
```

Langkah 2 - Membuat Widget Kustom

- Buat widget baru berupa kotak biru dengan icon jempol.
- Buat *class* baru turunan dari *StatelessWidget* di bawah *class MyApp*.
- Nama class: *KotakBiruJempol*.
- Gunakan widget *Container* di *method build()*.

Kode program:

```
1 class KotakBiruJempol extends StatelessWidget {
2   @override
3   Widget build(BuildContext context) {
4     return Container(
5       width: 100,
6       height: 100,
7       decoration: BoxDecoration(
8         color: Colors.blue,
9         border: Border.all(color: Colors.black, width: 2),
10        borderRadius: BorderRadius.circular(12),
11      ),
12      child: Icon(
13        Icons.thumb_up,
14        color: Colors.white,
15        size: 40,
16      ),
17    );
18  }
19 }
```

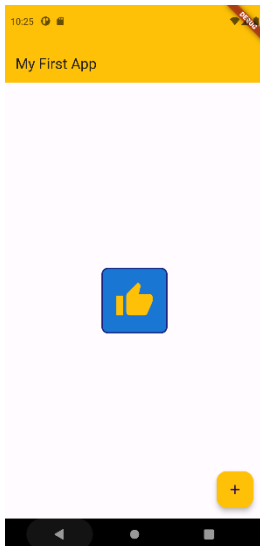
Langkah 3 - Menampilkan widget kustom

Ganti widget *Text* di dalam *Center* menjadi *KotakBiruJempol*.

Kode program:

```
1 body: Center(
2   child: KotakBiruJempol(),
3 ),
```

Hasil:



Langkah 4 - Menambahkan beberapa widget dengan *Row*

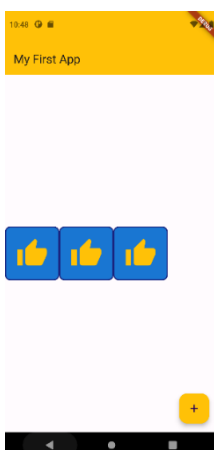
- Bungkus KotakBiruJempol dengan widget *Row* di dalam *Center*.
- Tambahkan dua widget *KotakBiruJempol* lainnya di properti *children*.

Kode program:

```

1  body: Center(
2    child: Row(
3      mainAxisAlignment: MainAxisAlignment.center,
4      children: [
5        KotakBiruJempol(),
6        SizedBox(width: 20), // jarak antar kotak
7        KotakBiruJempol(),
8        SizedBox(width: 20),
9        KotakBiruJempol(),
10     ],
11   ),
12 ),
```

Hasil:



Langkah 5 - Mengatur tata letak vertikal dengan *Column*

- Ganti *Row* menjadi *Column* untuk menampilkan kotak secara vertikal.
- Gunakan *SizedBox* untuk memberi jarak antar kotak.

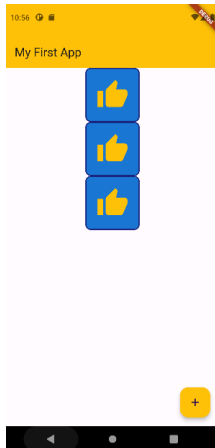
Kode program:

```

1 body: Center(
2   child: Column(
3     mainAxisAlignment: MainAxisAlignment.center,
4     children: [
5       KotakBiruJempol(),
6       SizedBox(height: 20),
7       KotakBiruJempol(),
8       SizedBox(height: 20),
9       KotakBiruJempol(),
10    ],
11  ),
12 ),

```

Hasil:



2.3. Kesimpulan

- Widget *Row* dan *Column* memudahkan pengaturan layout horisontal dan vertikal.
- Widget custom dapat dibuat menggunakan *StatelessWidget* dan *Container*.
- Properti seperti *width*, *height*, *color*, *border*, *borderRadius*, dan *Icon* membantu mempercantik tampilan.
- Row* dan *Column* adalah widget *invisible* yang berfungsi sebagai *parent* untuk widget *children*.

2.4. Tugas Praktikum

- Tambahkan 4 kotak biru dengan warna berbeda.
- Gabungkan *Row* dan *Column* untuk membuat grid 2x2.
- Ubah icon menjadi *Icons.favorite* dan atur warnanya *merah*.
- Tambahkan *Text* di bawah setiap *icon* menggunakan widget *Column*.

2.5. Format Laporan Praktikum

Dari hasil praktikum ini, buatlah laporan dengan format sebagai berikut:

- Cover

Memuat identitas laporan meliputi judul praktikum, nama, NIM, kelas, dan tanggal pelaksanaan.

- Tujuan

Berisi uraian singkat mengenai tujuan praktikum, yaitu memahami penggunaan widget *Row* dan *Column* dalam mengatur tata letak antarmuka aplikasi Flutter.

c. Langkah Kerja Singkat

Menyajikan ringkasan tahapan praktikum yang dilakukan, misalnya membuat project baru, menambahkan kode *Row* dan *Column*, serta menjalankan aplikasi pada *emulator* atau *smartphone*.

d. Screenshot

Menampilkan bukti hasil praktikum berupa *screenshot* output program menggunakan widget *Row* dan *Column*, lengkap dengan keterangan singkat pada setiap gambar.

e. Kesimpulan

Menguraikan ringkasan hasil praktikum dan pemahaman yang diperoleh penggunaan widget *Row* dan *Column*.