

# **LAPORAN PRAKTIKUM MOBILE PROGRAMMING**

## **MODUL II – Widget *Row* dan *Column***



Nama : Salis Qodri Mufti Muhammad

NIM : 230605110069

Kelas : E

Tanggal : 4 September 2025

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
GANJIL 2025/2026**

## I. Tujuan

Tujuan dari praktikum ini adalah mengetahui tentang UI pada Flutter, yakni Row dan Column sebagai dasar tata letak (Layout) antarmuka aplikasi. Dengan mempelajari hal tersebut, maka mahasiswa diharapkan bisa:

1. Mengatur posisi dan susunan elemen UI secara horizontal menggunakan widget Row.
2. Mengatur posisi dan susunan elemen UI secara vertikal menggunakan widget Column.
3. Mengombinasikan Row dan Column untuk membentuk antarmuka yang lebih kompleks dan responsif.
4. Menerapkan *alignment* dan *spacing* pada widget Row dan Column untuk membentuk antarmuka yang lebih terstruktur.

## II. Langkah Kerja

1. Membuat widget custom

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()*.

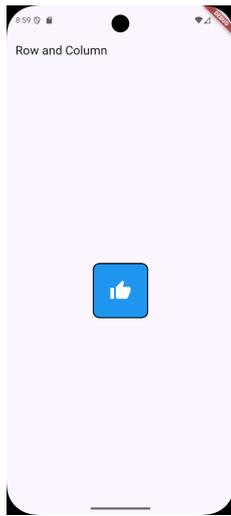
```
class KotakBiruJempol extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Container(
      width: 100,
      height: 100,
      decoration: BoxDecoration(
        color: Colors.blue,
        border: Border.all(color: Colors.black, width: 2),
        borderRadius: BorderRadius.circular(12),
      ), // BoxDecoration
      child: Icon(Icons.thumb_up, color: Colors.white, size: 40),
    ); // Container
  }
}
```

## 2. Menampilkan widget custom

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

```
body: Center(  
  child: KotakBiruJempol()  
) , // Center
```

Hasil:



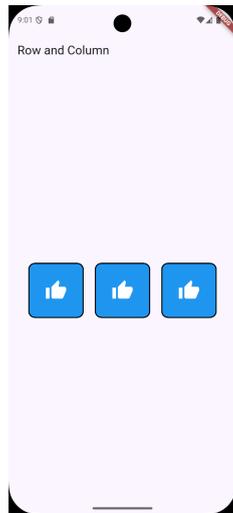
## 3. Menambahkan beberapa widget

Bungkus *KotakBiruJempol* dengan widget *Row* di dalam *Center*.

Tambahkan dua widget *KotakBiruJempol* lainnya di properti *children*.

```
body: Center(  
  child: Row(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: [  
      KotakBiruJempol(),  
      SizedBox(width: 20),  
      KotakBiruJempol(),  
      SizedBox(width: 20),  
      KotakBiruJempol(),  
    ],  
  ), // Row  
) , // Center
```

Hasil:



#### 4. Mengatur tata letak column

Ganti *Row* menjadi *Column* untuk menampilkan kotak secara vertikal.

Gunakan *SizeBox* untuk memberi jarak antar kotak.

```
body: Center(  
  child: column(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: [  
      KotakBiruJempol(),  
      SizeBox(height: 20),  
      KotakBiruJempol(),  
      SizeBox(height: 20),  
      KotakBiruJempol(),  
    ],  
  ), // Column
```

Hasil:



## 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*.

Kode Program:

Class KotakBiruJempol

```
class KotakBiruJempol extends StatelessWidget {
  final Color color;
  final IconData icon;
  final double size;

  const KotakBiruJempol({
    super.key,
    required this.color,
    required this.icon,
    this.size = 100,
  });

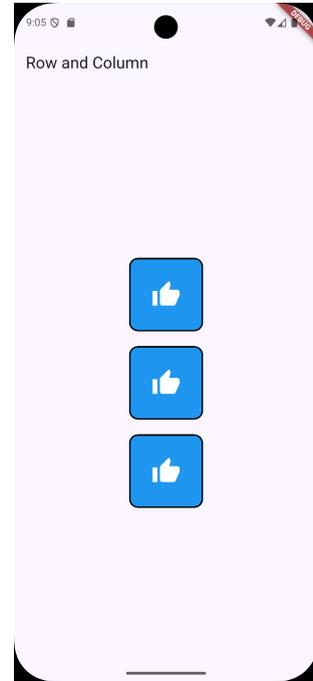
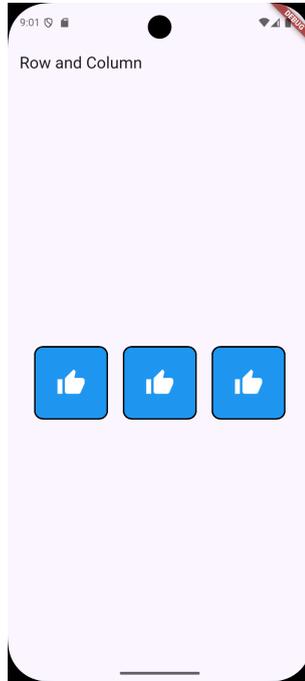
  @override
  Widget build(BuildContext context) {
    return Container(
      width: 100,
      height: 100,
      decoration: BoxDecoration(
        color: color,
        border: Border.all(color: Colors.black, width: 2),
        borderRadius: BorderRadius.circular(12),
      ), // BoxDecoration
      child: Icon(icon, color: Colors.white, size: size * 0.4),
    ); // Container
  }
}
```

## Body

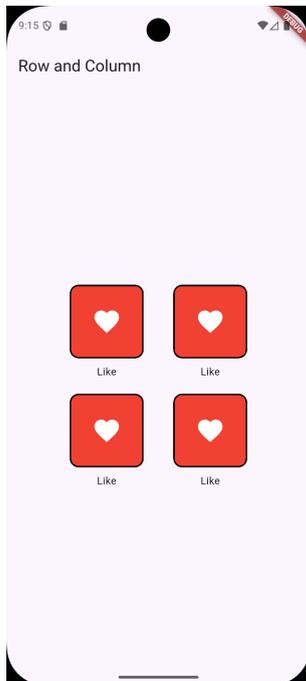
```
16 body: Center(  
17   child: Column(  
18     mainAxisAlignment: MainAxisAlignment.center,  
19     children: [  
20       Row(  
21         mainAxisAlignment: MainAxisAlignment.center,  
22         children: [  
23           Column(  
24             children: const [  
25               KotakBiruJempol(  
26                 color: Colors.red,  
27                 icon: Icons.favorite,  
28                 size: 100,  
29               ), // KotakBiruJempol  
30               SizedBox(height: 8),  
31               Text('Like'),  
32             ],  
33           ), // Column  
34           const SizedBox(width: 40),  
35           Column(  
36             children: const [  
37               KotakBiruJempol(  
38                 color: Colors.red,  
39                 icon: Icons.favorite,  
40                 size: 100,  
41               ), // KotakBiruJempol  
42               SizedBox(height: 8),  
43               Text('Like'),  
44             ],  
45           ), // Column  
46         ],  
47       ), // Row  
48       const SizedBox(height: 20),  
49       Row(  
50         mainAxisAlignment: MainAxisAlignment.center,  
51         children: [  
52           Column(  
53             children: const [  
54               KotakBiruJempol(  
55                 color: Colors.red,  
56                 icon: Icons.favorite,  
57                 size: 100,  
58               ), // KotakBiruJempol  
59               SizedBox(height: 8),  
60               Text('Like'),  
61             ],  
62           ), // Column  
63           const SizedBox(width: 40),  
64           Column(  
65             children: const [  
66               KotakBiruJempol(  
67                 color: Colors.red,  
68                 icon: Icons.favorite,  
69                 size: 100,  
70               ), // KotakBiruJempol  
71               SizedBox(height: 8),  
72               Text('Like'),  
73             ],  
74           ), // Column  
75         ],  
76       ), // Row  
77     ],  
78   ), // Column  
79 ), // Center
```

### III. Screenshot Hasil

Praktikum:



Tugas Praktikum:



#### **IV. Kesimpulan**

Pada praktikum ini, dapat disimpulkan bahwa widget Row dan Column berperan penting dalam penataan antarmuka aplikasi. Row digunakan untuk menyusun elemen secara horizontal. Sedangkan, Column digunakan untuk menyusun elemen secara vertikal. Kombinasi antara keduanya yang kemudian dipadukan dengan alignment dan spacing akan membangun tampilan aplikasi yang rapi, terstruktur dan responsif sesuai dengan kebutuhan pengembangan aplikasi.