

LAPORAN TEORI MOBILE PROGRAMMING
MODUL 3



Nama : Firman Fadilah Noor
NIM : 240605110083
Kelas : B
Tanggal : 17 September 2025

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

I. Tujuan

1. Memahami konsep main axis cross axis pada widget Row dan Column dalam Flutter.
2. Mengetahui penggunaan `MainAxisSize.max` dan `MainAxisSize.min`.
3. Mempelajari nilai property `MainAxisAlignment` untuk mengatur distribusi ruang.
4. Mempelajari nilai property `CrossAxisAlignment` untuk mengatur posisi pada sumbu silang.
5. Mengimplementasikan properti-properti tersebut dalam kodeFlutter.

II. Langkah Kerja

1. Membuat proyek Flutter baru.
2. Menambahkan kode *row* dan *column*
3. Menjalankan aplikasi pada *emulator* dan *smartphone*
4. Pengujian

III. Screenshot Hasil

a. Kode Program

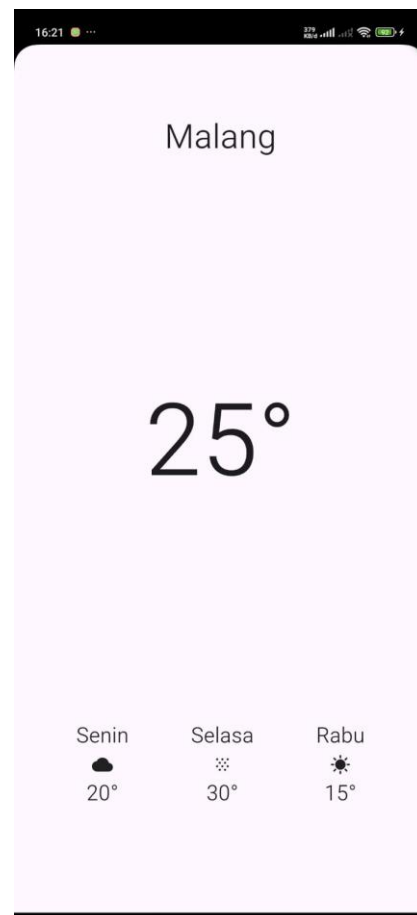
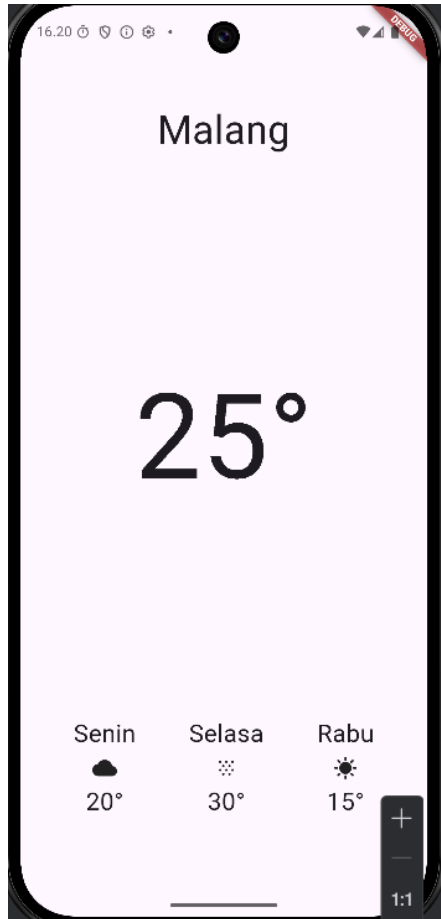
```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9
10  @override
11  Widget build(BuildContext context) {
12    return MaterialApp(
13      title: 'Alignment',
14      theme: ThemeData.light(),
15      home: Scaffold(
16        body: Center(
17          child: Column(
18            mainAxisAlignment: MainAxisAlignment.spaceAround,
19            children: [
20              Text(
21                'Malang',
22                style: TextStyle(fontSize: 40),
23              ), // Text Text
24              Text(
25                '25°',
26                style: TextStyle(fontSize: 120),
27              ), // Text Text
28              Row(
29                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
30                children: [
31                  harian('Senin', Icons.cloud, '20°'),
32                  harian('Selasa', Icons.snowing, '30°'),
33                  harian('Rabu', Icons.sunny, '15°'),
```

```
34                ],
35              ), // Row Row
36            ],
37          ), // Column Column
38        ), // Center Center
39      ), // Scaffold Scaffold
40    ); // MaterialApp MaterialApp
41  }
42 }
43
44 Container harian (String hari, IconData icon, String suhu) {
45   return Container(
46     height: 105,
47     child: Column(
48       mainAxisAlignment: MainAxisAlignment.spaceBetween,
49       children: [
50         Text(
51           hari,
52           style: TextStyle(fontSize: 25),
53         ), // Text Text
54         Icon(icon, size: 25),
55         Text(
56           suhu,
57           style: TextStyle(fontSize: 25),
58         ), // Text Text
59       ],
60     ), // Column Column
61   ); // Container Container
62 }
63 }
```

b. Penjelasan Kode Program

Kode program di atas merupakan sebuah aplikasi sederhana menggunakan framework **Flutter** yang menampilkan informasi cuaca. Program dimulai dari fungsi `main()` yang menjalankan widget utama `MyApp`. Kelas `MyApp` merupakan `StatelessWidget` yang membangun tampilan antarmuka melalui `MaterialApp`. Dalam `MaterialApp`, digunakan `Scaffold` sebagai struktur dasar halaman, dan di dalam body terdapat `Center` yang memusatkan seluruh konten di layar. Konten utama disusun menggunakan `Column` yang mengatur widget secara vertikal dengan `mainAxisAlignment: MainAxisAlignment.spaceAround` agar setiap elemen memiliki jarak yang merata. Pada bagian atas ditampilkan teks "Malang" sebagai nama kota, kemudian di bawahnya ditampilkan suhu utama "25°" dengan ukuran font yang besar. Setelah itu terdapat `Row` yang menampilkan tiga informasi cuaca harian (Senin, Selasa, dan Rabu) secara horizontal dengan `mainAxisAlignment: MainAxisAlignment.spaceEvenly` agar setiap elemen tersebar merata. Masing-masing informasi harian dibentuk melalui fungsi `harian`, yang mengembalikan sebuah `Container` berisi `Column` yang menampilkan teks hari, ikon cuaca, dan suhu harian. Dengan demikian, keseluruhan tampilan aplikasi membentuk tata letak cuaca sederhana yang rapi dan terpusat.

Output:



IV. Kesimpulan

Penggunaan widget Row dan Column pada Flutter sangat penting untuk mengatur tata letak elemen antarmuka secara horizontal dan vertikal. Dengan memanfaatkan properti seperti `mainAxisAlignment` dan `crossAxisAlignment`, pengembang dapat mengontrol posisi dan distribusi ruang antar elemen agar tampilan menjadi lebih rapi, terstruktur, dan responsif. Implementasi konsep ini pada aplikasi sederhana cuaca menunjukkan bahwa pengaturan tata letak yang tepat dapat menciptakan antarmuka yang menarik dan mudah dipahami pengguna.