

**LAPORAN TEORI MOBILE PROGRAMMING**  
**MODUL 4**



Nama : Erai Bagusalm  
NIM : 240605110088  
Kelas : Mobile Programming 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 dan cross axis pada widget Row dan Column.
2. Mengetahui penggunaan `MainAxisSize.max` dan `MainAxisSize.min`.
3. Mempelajari nilai properti `MainAxisAlignment` untuk mengatur distribusi ruang.
4. Mempelajari nilai properti `CrossAxisAlignment` untuk mengatur posisi pada sumbu silang.
5. Mengimplementasikan properti-properti tersebut dalam kodeFlutter.

ii. **Langkah Kerja**

1. Membuat Project Baru
2. Menambahkan widget *row*, *expanded*, dan *flexible*.
3. Menjalankan aplikasi pada *emulator* atau *smartphone*.
4. Pengujian

### iii. Screenshot Hasil

#### a. Kode Program

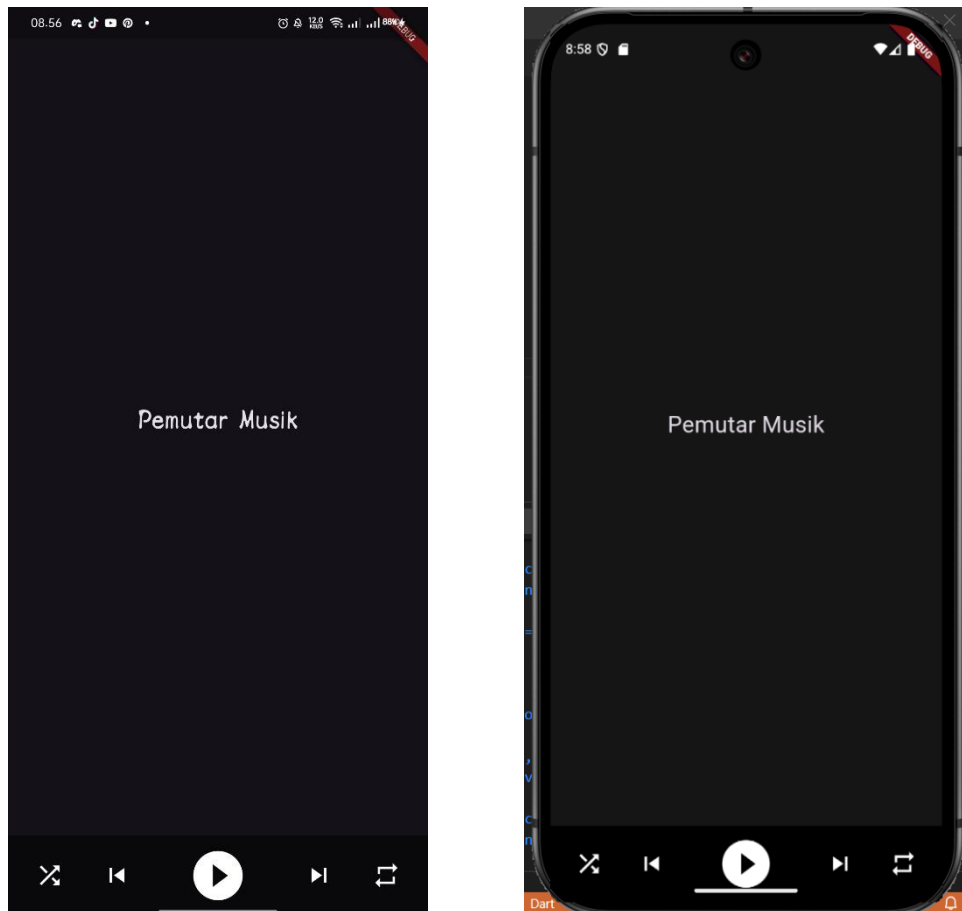
```
1 import 'package:flutter/material.dart';
2
3 Run | Debug | Profile
4 void main() {
5   runApp(const MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   const MyApp({super.key});
10
11   // This widget is the root of your application.
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       title: 'Flutter Demo',
16       theme: ThemeData.dark(),
17       home: const MyHomePage(title: 'Flutter Demo Home Page'),
18     ); // MaterialApp
19   }
20
21   class MyHomePage extends StatefulWidget {
22     const MyHomePage({super.key, required this.title});
23
24     final String title;
25
26     @override
27     State<MyHomePage> createState() => _MyHomePageState();
28   }
```

```
29 class _MyHomePageState extends State<MyHomePage> {
30   @override
31   Widget build(BuildContext context) {
32     return Scaffold(
33       body: Center(
34         child: Text(
35           'Pemutar Musik',
36           style: TextStyle(fontSize: 24),
37         ), // Text
38       ), // Center
39       bottomNavigationBar: Container(
40         padding: EdgeInsets.symmetric(vertical: 10, horizontal: 8),
41         color: Colors.black54,
42         child: Row(
43           children: [
44             Expanded(
45               child: Icon(Icons.shuffle, color: Colors.white, size: 30),
46             ), // Expanded
47             Expanded(
48               child: Icon(Icons.skip_previous, color: Colors.white, size: 30),
49             ), // Expanded
50             Flexible(
51               fit: FlexFit.tight,
52               flex: 2,
53               child: Icon(Icons.play_circle_fill, color: Colors.white, size: 60),
54             ), // Flexible
55             Expanded(
56               child: Icon(Icons.skip_next, color: Colors.white, size: 30),
57             ), // Expanded
58             Expanded(
59               child: Icon(Icons.repeat, color: Colors.white, size: 30),
60             ), // Expanded
61           ], // Row
62         ), // Container
63       ), // Scaffold
64     );
65   }
66 }
```

#### b. Penjelasan Kode Program

Kode program di atas aplikasi sederhana Flutter dengan tema gelap yang menampilkan teks “Pemutar Musik” di bagian tengah layar, serta sebuah control bar pemutar musik di bagian bawah menggunakan properti `bottomNavigationBar`. Pada control bar tersebut, digunakan `Container` dengan padding dan warna latar hitam transparan agar tampak kontras. Di dalamnya terdapat sebuah `Row` yang berisi lima ikon yaitu Shuffle, Previous, Play, Next, dan Repeat. Penempatan posisi pada ikon diatur dengan kombinasi `Expanded` dan `Flexible`. Empat ikon (Shuffle, Previous, Next, dan Repeat) menggunakan `Expanded` agar ukurannya rata dan merata memenuhi ruang, sementara ikon Play menggunakan `Flexible` dengan `flex: 2` dan `fit: FlexFit.tight`, sehingga tampil lebih besar dan menonjol dibanding tombol lainnya. Dengan demikian, kode ini mendemonstrasikan penggunaan widget `Expanded` dan `Flexible` untuk mengatur tata letak tombol kontrol musik secara proporsional dalam sebuah bar di bagian bawah aplikasi.

Output :



#### iv. Kesimpulan

Berdasarkan laporan yang telah dibuat, dapat disimpulkan bahwa penggunaan widget Row, Expanded, dan Flexible pada Flutter memungkinkan pengaturan tata letak komponen aplikasi secara lebih proporsional dan fleksibel. Expanded digunakan untuk membagi ruang secara merata antar widget, sedangkan Flexible memberikan keleluasaan untuk menentukan ukuran widget dengan pengaturan flex dan fit. Pada contoh control bar pemutar musik, keduanya membuat tombol Play dapat ditampilkan lebih besar dibanding tombol lainnya tanpa mengganggu tampilan ruang tombol lain. Hal ini menunjukkan bahwa pemahaman dan penerapan widget tata letak sangat penting untuk menghasilkan antarmuka yang rapi, proporsional, dan fungsional dalam pengembangan aplikasi mobile.