

LAPORAN PRAKTIKUM MOBILE PROGRAMMING
MODUL 4 – FLEXIBLE DAN EXPANDED



Nama : KHARISMA ROMADHON EKA SAFITRI

NIM : 230605110107

Kelas : E

Tanggal : Kamis, 11 September 2025

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

I. Tujuan

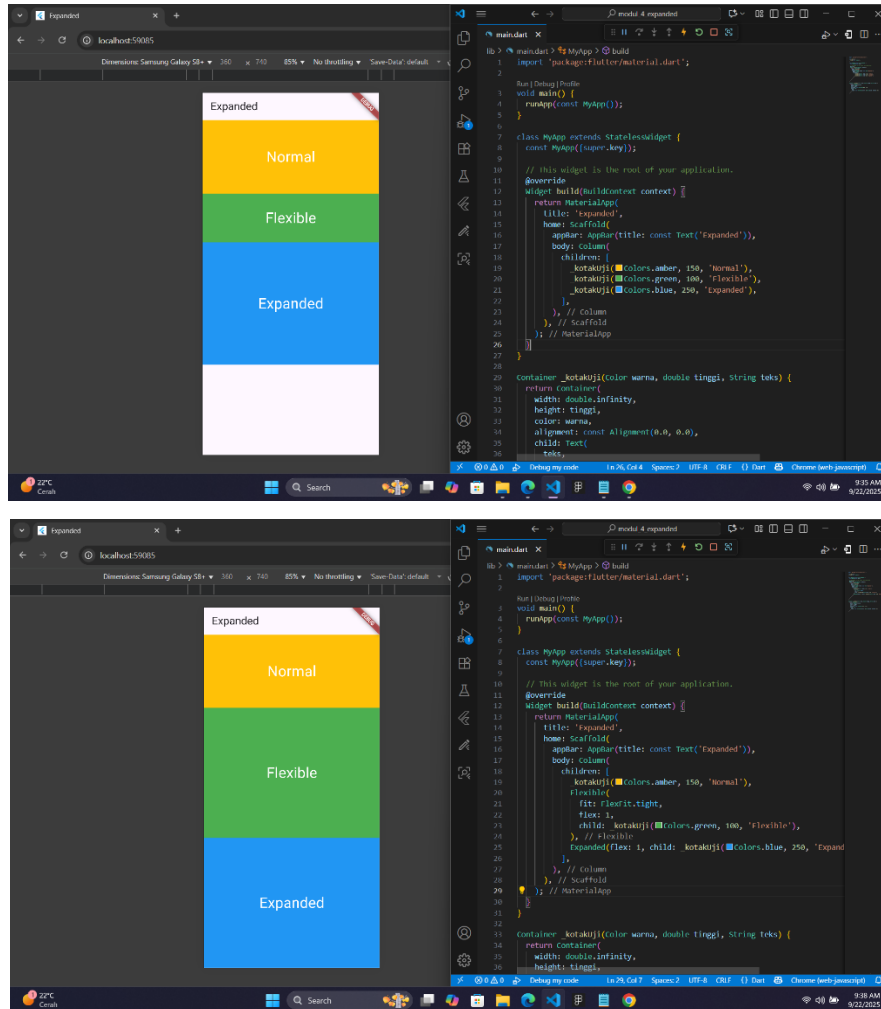
- a. Memahami penggunaan widget Row, Expanded, dan Flexible.
- b. Mempelajari cara mendistribusikan ruang secara proporsional antar tombol kontrol musik.
- c. Mengetahui perbedaan efek penggunaan Flexible (flex, fit) dan Expanded pada tata letak UI.

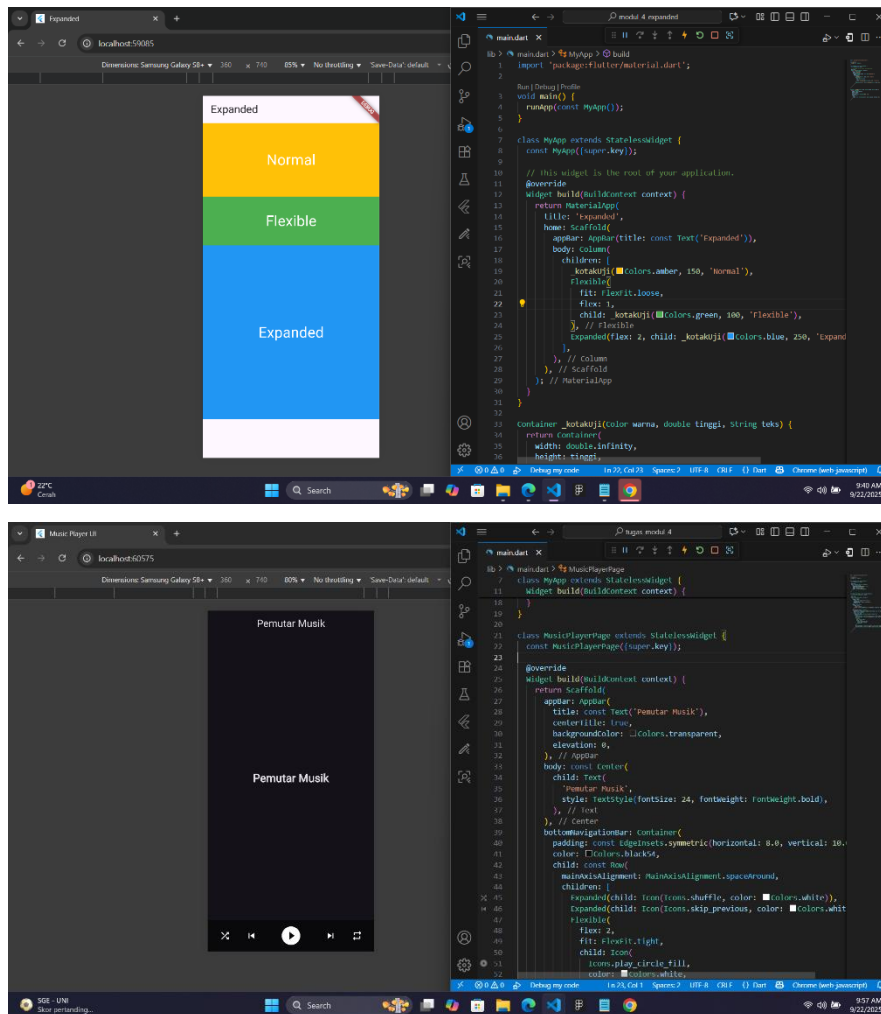
II. Langkah Kerja

1. Membuat proyek Flutter baru dan mengatur tema aplikasi menjadi `ThemeData.dark()`.
2. Menambahkan
3. Menambahkan Scaffold dengan properti `bottomNavigationBar` untuk menempatkan Control Bar di bagian bawah layar.
4. Membuat row yang berisi lima tombol icon : shuffle, previous, next dan repeat, menggunakan widget expanded agar ruang terbagi secara merata.
5. Menggunakan widget flexible dengan property `flex:2` dan `fit: FlexFit.tight` khusus untuk tombol play agar tampil lebih besar dari tombol lainnya.
6. Memberi warna putih pada semua icon.

III. Screenshot Hasil

Sertakan hasil praktikum berupa tangkapan layar, diberi keterangan singkat.





Pada main() dan MyApp ditambahkan elemen theme ThemeData.dark().

Kemudian membuat class dengan nama MusicPlayerPage yang merupakan halaman utama aplikasi yang menggunakan Scaffold untuk menyediakan struktur dasar.

Properti bottomNavigationBar ini digunakan untuk menempatkan Control Bar di bagian bawah layar.

Widget Row digunakan untuk menata ikon-ikon tombol secara horizontal.

Expanded digunakan untuk empat tombol (Shuffle, Previous, Next, Repeat) untuk memastikan mereka memiliki besar dan jarak yang sama secara merata di sepanjang Row.

Flexible digunakan pada tombol Play. Dengan flex: 2 dan fit: FlexFit.tight, tombol play akan memenuhi dua kali lipat ruang yang dialokasikan untuk tombol lain, membuatnya terlihat lebih besar dan menonjol.

IV. Source Code

```
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(
      debugShowCheckedModeBanner: false,
      title: 'Music Player UI',
      theme: ThemeData.dark(),
      home: const MusicPlayerPage(),
    );
  }
}

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

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Pemutar Musik'),
      centerTitle: true,
      backgroundColor: Colors.transparent,
      elevation: 0,
    ),
    body: const Center(
      child: Text(
        'Pemutar Musik',
        style: TextStyle(fontSize: 24, fontWeight:
FontWeight.bold),
      ),
    ),
    bottomNavigationBar: Container(
      padding: const EdgeInsets.symmetric(horizontal:
8.0, vertical: 10.0),
      color: Colors.black54,
      child: const Row(
        mainAxisAlignment:
MainAxisAlignment.spaceAround,
        children: [
          Expanded(child: Icon(Icons.shuffle, color:
Colors.white)),
          Expanded(child: Icon(Icons.skip_previous,
color: Colors.white)),
          Flexible(

```

```

        flex: 2,
        fit: FlexFit.tight,
        child: Icon(
          Icons.play_circle_fill,
          color: Colors.white,
          size: 48,
        ),
      ),
      Expanded(child: Icon(Icons.skip_next,
color: Colors.white)),
      Expanded(child: Icon(Icons.repeat, color:
Colors.white)),
    ],
  ),
),
);
}
}

```

V. Kesimpulan

Expanded dan **Flexible** sangat efektif untuk mengelola distribusi ruang dalam tata letak UI, terutama saat digabungkan dalam sebuah Row atau Column. Setiap tombol dapat ditampilkan dengan ukuran yang berbeda tanpa mengganggu tata letak tombol lainnya karena sisa ruang akan disesuaikan secara otomatis menyusun tata letak aplikasi yang lebih rapi, fleksibel, dan responsif terhadap berbagai ukuran layar.