

# Petunjuk Praktikum

Mata Kuliah	Pemrograman Mobile
Pertemuan	Minggu-07
Topik	Dasar Widget.
Pengampu	Yoppy Yunhasnawa, S.ST., M.Sc.

## Topik-4: Tampilan Awal Aplikasi MyDJ

Pada praktikum ini, kita akan membuat halaman awal dari aplikasi MyDJ - jurnal harian untuk guru. Aplikasi ini digunakan oleh para guru untuk mendokumentasikan kegiatan pengajaran yang dilakukan oleh mereka setiap hari. Dokumentasinya bisa terdiri dari teks, foto, maupun video. Berikut ini adalah tampilan aplikasi yang diharapkan:

### 1. Persiapan

1. Buatlah sebuah proyek Flutter baru di komputer Anda dengan nama mydj . 2.

Pada deskripsi proyek tersebut isikan: Aplikasi jurnal harian guru .

3. Pada isian Organization , jangan lupa ganti com.example , menjadi com.namaanda , atau id.ac.polinema .

4. Hapus semua source code template pada file `main.dart` di project baru Anda. 5.

Pada main.dart , tambahkan kode berikut:

```
import 'package:flutter/material.dart';
import 'package:mydj/pages/simple_home_page.dart';

void main() {
  runApp(const MyApp());
}
class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
```

```

return MaterialApp(
  title: 'MyDJ',
  theme: ThemeData(
    colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
    home: const SimpleHomePage(title: 'Jurnal Harian Guru')
  );
}
}

```

6. Buatlah folder bernama pages di folder lib . Pada folder tersebut, tambahkan 1 file dart baru, beri nama dengan simple\_home\_page.dart .

7. Pada file tersebut, tambahkan kode class `SimpleHomePage` berikut:

```

class SimpleHomePage extends StatefulWidget
{
  const SimpleHomePage({super.key, required this.title});

  final String title;

  @override
  State<StatefulWidget> createState()
  {
    return _SimpleHomePageState(); // Tidak masalah bila saat ini error karena memang class
    _SimpleHomePageState belum kita buat.
  }
}

```

8. Berikutnya, pada file yang sama, di bawah class `SimpleHomePage` tersebut, tambahkan class `_SimpleHomePageState` seperti yang namanya tertulis di return statement pada fungsi createState() sebelumnya.

```

class _SimpleHomePageState extends State<SimpleHomePage>
{
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Theme.of(context).colorScheme.inversePrimary, title: Text(widget.title),
      ),
      body: Center(

    )
  );
}

```

```
}  
}
```

9. Jalankan aplikasi Anda (boleh di emulator atau di web), perhatikan seharusnya sudah dibuat tampilan aplikasi kosong dengan App Bar berwarna ungu yang memiliki judul.

## 2. Menggunakan Layout

1. Di class `SimpleHomePage` sebelumnya, pada bagian `body: Center()`, modifikasilah kodenya agar menjadi seperti berikut ini.

```
body: Center(  
  child: Row(  
    mainAxisAlignment: MainAxisAlignment.center,  
    crossAxisAlignment: CrossAxisAlignment.center,  
    children: [  
      ElevatedButton(onPressed: () => {}, child: Text('Lihat Jurnal')),  
    ],  
  ),  
)
```

2. Hot reload aplikasi Anda, maka sekarang akan tampil sebuah tombol di bagian tengah aplikasi. Mengapa tombolnya bisa ada di tengah?
3. Tambahkan satu `ElevatedButton` lagi di bawah tombol `Lihat Jurnal`, hot reload aplikasi Anda dan amati apa yang terjadi!

```
ElevatedButton(onPressed: () => {}, child: Text('Buat Jurnal'))
```

Akan ditampilkan 2 button yang saling berjajar kanan-kiri di tengah halaman. Mengapa demikian?

4. Saat ini, dua button tersebut terlihat menempel. Bagaimana caranya agar 'agak terpisah'? Jawabannya adalah dengan menggunakan widget `SizedBox`. Widget ini sangat berguna dan fleksibel untuk mengatur-atur jarak di-antara 2 widget. Widget ini berupa kotak sederhana yang bisa diatur-atur panjang (width) dan lebarnya (height). Di antara kedua `ElevatedButton` yang baru saja kita buat, tambahkan `SizedBox` sehingga isi dari `children[]` menjadi seperti berikut:

```
ElevatedButton(onPressed: () => {}, child: Text('Lihat Jurnal')), SizedBox(width: 20),
```

```
ElevatedButton(onPressed: () => {}, child: Text('Buat Jurnal'))
```

5. [Latihan] Cobalah sendiri untuk menambahkan 2 ElevatedButton lagi di yang berjajar juga namun kali ini di bawah button sebelumnya. Berilah teks pada Button tersebut: Akun dan Tentang Aplikasi .

### 3. Menggunakan Assets

1. Selanjutnya kita akan menambahkan logo ke halaman depan aplikasi kita. Untuk melakukannya, pertama-tama buatlah folder `assets/images` di project Anda, di posisi yang sejajar dengan folder `lib` .
2. Carilah logo Kemdikbud di internet yang berjenis PNG. Unduh dan letakkan file tersebut di folder yang baru saja Anda buat. Ganti nama file tersebut menjadi `logo_kemdikbud.png` .
3. Bukalah file `pubspec.yaml` . Editlah file tersebut dengan menambahkan kode berikut di bawah section flutter: .

```
flutter:  
assets:  
  - assets/images/logo_kemdikbud.png
```

Perhatian: Gunakan 2 spasi pada file YAML dan BUKAN tab. File YAML sensitif terhadap spasi.

4. Klik tombol `Sync` yang ada di sebelah kanan atas editor untuk mensinkronkan gambar yang baru kita tambahkan tadi.
5. Selanjutnya, di class `_SimpleHomePage` , pada bagian `Row(children:[])` , tambahkan widget berikut ini sebelum button yang paling awal.

```
Image.asset(  
  'assets/images/logo_kemdikbud.png',  
  width: 150,  
  height: 150  
)
```

6. Hot Reload aplikasi Anda, akan dimunculkan gambar logo Kemdikbud di aplikasi Anda. Apabila muncul error yang mengatakan bahwa image-nya tidak ada, lakukan Hot Restart

pada proyek Anda.

7. [Latihan] Bagaimana posisi Image Anda? Cobalah sendiri untuk mengubah letak Image sedemikian sehingga menjadi di atas Button-Button yang ada.

## 4. Menggunakan Navigasi

Satu aplikasi biasanya terdiri dari beberapa halaman. Jarang ada aplikasi yang terdiri dari satu halaman saja. Pada bagian ini kita akan coba menambahkan satu halaman dan membuat agar aplikasi kita bisa berpindah dari halaman Home ke halaman baru tersebut ketika pengguna mengklik tombol Lihat Jurnal .

1. Pada folder lib/pages , tambahkan satu file baru bernama lihat\_jurnal\_page.dart . Pada file tersebut tambahkan 1 class widget dan 1 class state standar seperti pada file sebelumnya. Pertama-tama tambahkan class widget-nya dulu.

```
import 'package:flutter/material.dart';

class LihatJurnalPage extends StatefulWidget {
  const LihatJurnalPage({super.key, required this.title});
  final String title;

  @override
  State<LihatJurnalPage> createState() {
    {
      return _LihatJurnalPageState(); // Seperti biasa, baris ini masih error. }
    }
  }
```

2. Berikutnya kita tambahkan class state dari page tersebut dengan menambahkan kode berikut ini di bawah class sebelumnya.

```
class _LihatJurnalPageState extends State<LihatJurnalPage> {
  @override
  Widget build(BuildContext context) {
    {
      return Scaffold(
        appBar: AppBar(
          backgroundColor: Theme.of(context).colorScheme.inversePrimary, title: Text(widget.title),
        ),
        body: Center(
          child: Column(
```

```

mainAxisAlignment: MainAxisAlignment.center,
crossAxisAlignment: CrossAxisAlignment.center,
children: []
)
);
}
}

```

3. Sampai di sini, kita sudah membuat suatu halaman kosong dengan AppBar berwarna ungu seperti sebelumnya. Berikutnya, pada class `SimpleHomePageSial` tambahkan method berikut yang memanggil class Navigator , yang akan memindahkan halaman ke page tertentu ketika ia dieksekusi.

```

void _openLihatJurnalPage(BuildContext context)
{
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (context) => LihatJurnalPage(title: 'Lihat Jurnal') // Halaman kedua (Widget)
    ),
  );
}

```

4. Berikutnya, modifikasilah ElevatedButton `Lihat Jurnal` yang ada di halaman home agar ketika diklik menjalankan action `_openLihatJurnalPage()` yang sudah kita buat sebelumnya.

```
ElevatedButton(onPressed: () => {_openLihatJurnalPage(context)}, child: Text('Lihat Jurnal')),
```

Perhatikan pada Button tersebut, kita menambahkan action pada property onPressed pada ElevatedButton dengan *fat arrow notation*.

5. Reload aplikasi Anda, seharusnya ketika Button diklik, aplikasi akan berpindah ke halaman baru.

## Tugas Latihan

1. Jawablah semua pertanyaan yang ada di petunjuk praktikum di atas.
2. Kerjakan semua yang diminta pada bagian [Latihan].
3. Modifikasilah tampilan aplikasi MyDJ agar serupa dengan halaman aplikasi yang ada di

bagian awal petunjuk praktikum ini.

4. Buatlah 3 lagi halaman kosong, dan hubungkan dengan 3 button lain yang ada pada tampilan awal ini.

## Penutup

Setelah menyelesaikan praktikum ini, harapannya Anda sudah memahami tentang:

1. Tumpukan Widget dalam struktur dasar aplikasi Flutter.
2. Layout, khususnya Center, Column, dan Row.
3. Cara menggunakan asset dalam aplikasi, dan;
4. Cara berpindah ke halaman lain dari aplikasi.