

## **MINGGU 6: SIDEBAR, DRAWER, DAN UI**

<b>Pokok Bahasan</b>	<b>: SideBar, Drawer, GridView, dan UI</b>
<b>Minggu</b>	<b>: 6</b>
<b>Tempat</b>	<b>: Daring / Luring Politeknik Negeri Jember</b>
<b>Alokasi Waktu</b>	<b>: 100 menit x 4</b>

### **a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu memahami konsep SideBar, Drawer, GridView, dan UI dalam Flutter
2. Mahasiswa mampu menerapkan SideBar, Drawer, GridView, dan UI dalam Flutter

### **b. Indikator**

Kemampuan mahasiswa dalam memahami dan menerapkan SideBar, Drawer, GridView, dan UI dalam Flutter

### **c. Dasar Teori**

Drawer, yang biasa disebut sebagai sidebar pada desain web, adalah panel dari desain material yang muncul dengan slide secara horizontal. Drawer ini biasanya digunakan untuk menampilkan link tautan navigasi pada aplikasi. Drawer biasanya digunakan dengan properti Scaffold.drawer. Child dari drawer biasanya adalah ListView yang diawali dengan DrawerHeader. DrawerHeader ini menampilkan informasi status tentang pengguna yang aktif saat ini. Children lain yang lainnya sering dibuat dengan ListTile, lalu diakhiri dengan AboutLostTile.

AppBar dengan otomatis akan menampilkan IconButton yang sesuai untuk menampilkan Drawer ketika Drawer sudah tersedia di dalam Scaffold. Kemudian Scaffold juga dengan otomatis menangani edge-swipe gesture untuk menampilkan drawer. Drawer yang sudah terbuka dapat ditutup dengan memanggil Navigator.pop.

### **d. Alat dan Bahan**

1. Java Development Kit (JDK)
2. Android Studio;
3. Android SDK;
4. Flutter SDK;
5. Teks Editor (atau bisa juga pakai Android Studio atau Visual Studio Code).

#### e. Prosedur Kerja

1. Sama seperti minggu-minggu sebelumnya, buatlah project baru dengan nama “sidebar\_gridview\_ui” untuk mencoba Sidebar dan GridView.
2. Dalam materi ini, source code default dari project flutter tidak perlu dihapus.
3. Pertama, tambahkan drawer: dengan widget Drawer didalamnya pada Scaffold yang ada dalam class \_MyHomePageState yang meng-extend State<MyHomePage>.

```
drawer: Drawer();
```

4. Lalu tambahkan child: dengan widget ListView dan widget itu memiliki children: ListTile seperti pada source code di bawah ini.

```
drawer: Drawer(  
  child: ListView(  
    children: <Widget>[  
      const ListTile(  
        title: Text("Setting"),  
        trailing: Icon(Icons.settings),  
      ),  
      const ListTile(  
        title: Text("Close"),  
        trailing: Icon(Icons.close),  
      ),  
    ],  
  ),  
)
```

5. Coba jalankan dan perhatikan hasilnya. Selanjutnya akan dibuat header dari Drawer dengan menggunakan widget UserAccountsDrawerHeader seperti source code di bawah ini.

```
UserAccountsDrawerHeader(  
  accountName: Text("I Gede Wiryawan"),  
  accountEmail: Text("igdw@windowslive.com"),  
  currentAccountPicture: CircleAvatar(backgroundImage: NetworkImage("https://scholar.googleusercontent.com/citations?view_op=view_photo&user=a_KipnEAAAAJ&ctpid=1")),  
  decoration: BoxDecoration(  
    image: DecorationImage(  
      image: NetworkImage(  
"https://ae01.alicdn.com/kf/HTB1p4MIKVXXXa.XVXXq6xXFXXR/1PC-Zakka-16-Lattice-Desktop-Drawer-Storage-Box-Wooden-Retro-Creative-Storage-Cabinet-Living-Room-Decoration.jpg"),  
      fit: BoxFit.cover)),  
    otherAccountsPictures: <Widget>[  
      CircleAvatar(  

```

```

        backgroundImage: NetworkImage("https://scholar.googleusercontent.com
/citations?view_op=view_photo&user=OFD_sjQAAAAJ&citpid=2"),
    ),
  ],
),

```

6. Jalankan dan perhatikan tampilan sidebar yang dihasilkan.
7. Selanjutnya ditambahkan fitur untuk berganti account apabila NetworkImage yang ada pada <Widget> otherAccountsPictures: disentuh atau onTap. Modifikasi source code yang ada pada otherAccountPicture: seperti menjadi potongan source code berikut ini.

```

otherAccountsPictures: <Widget>[
  GestureDetector(
    onTap: () => gantiUser(),
    child: CircleAvatar(
      backgroundImage: NetworkImage(gambar2),
    ))
],

```

8. Kemudian deklarasikan variabel gambar1, gambar2, dan backup dengan tipe data string, lalu untuk teks nama, tambahkan variabel nama1, nama2, dan backupNama dengan tipe data yang sama. Berikut adalah potongan source code untuk mendeklarasikan variabel-variabel tersebut.

```

String gambar1 =
  "https://scholar.googleusercontent.com/citations?view_op=view_photo&user
=a_KipnEAAAAJ&citpid=1";
String gambar2 =
  "https://scholar.googleusercontent.com/citations?view_op=view_photo&user
=OFD_sjQAAAAJ&citpid=2";
late String backup;

String nama1 = "I Gede Wiryawan";
String nama2 = "Syamsiar Kautsar";
late String backupNama;

```

9. Setelah itu tambahkan void gantiUser() di dalam class \_MyHomePageState untuk mengganti nilai variabel yang sebelumnya telah dideklarasikan seperti potongan source code di bawah ini.

```

void gantiUser() {
  setState(() {
    backup = gambar1;
    gambar1 = gambar2;
    gambar2 = backup;
  });
}

```

```

        backupNama = nama1;
        nama1 = nama2;
        nama2 = backupNama;
    });
}

```

10. Jalankan lalu Tap avatar yang ada di pojok kanan atas sidebar atau drawer, perhatikan hasilnya tersebut.
11. Selanjutnya akan ditambahkan satu file .dart lagi dengan nama detail.dart, sehingga jika avatar di sebelah kiri disentuh atau tap, maka akan diarahkan ke file .dart baru tersebut. Berikut adalah source code dari file .dart yang baru tersebut.

```

import 'package:flutter/material.dart';

class Detail extends StatelessWidget {
  Detail({required this.nama, required this.gambar});
  final String nama;
  final String gambar;

  @override
  Widget build(BuildContext context) {
    var decoration;
    return Scaffold(
      appBar: AppBar(
        title: Text("Detail-$nama"),
        backgroundColor: Colors.red[700],
      ),

      body: Center(
        child: Image(
          image: NetworkImage(gambar),
        )),
    );
  }
}

```

12. Kemudian tambahkan widget GestureDetector pada currentAccountPicture: seperti potongan source code di bawah ini

```

currentAccountPicture: GestureDetector(
  onTap: () {
    Navigator.of(context).pop();
    Navigator.of(context).push(MaterialPageRoute(
      builder: (BuildContext context) =>
        Detail(nama: nama1, gambar: gambar1)));
  },
  child: CircleAvatar(backgroundImage: NetworkImage(gambar1)),
),

```

**f. Hasil dan Pembahasan**

1. Dokumentasi tugas berupa laporan (penjelasan mengenai potongan source-code SideBar, Drawer, dan UI).
2. Dokumentasi tugas berupa file pdf.

**g. Kesimpulan**

Mahasiswa memahami mengenai SideBar, Drawer, GridView, dan UI dalam flutter.

**h. Rubrik Penilaian**

No	Indikator	Skor*			
1	Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas ditunjang dengan bukti referensi	1	2	3	④
2	Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas	1	2	③	4
3	Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas	1	②	3	4
4	Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas	①	2	3	4
Jumlah skor					

