

## Modul 7

# Komponen dalam mendesain Aplikasi

AppBar, FAB dan ButtonNavigation

### Module Overview

---

Mengenal komponen AppBar, FAB dan ButtonNavigation pada flutter. Widget ini digunakan pada property Scaffold (optional) untuk mendesain tampilan UI aplikasi flutter.

### Module Objectives

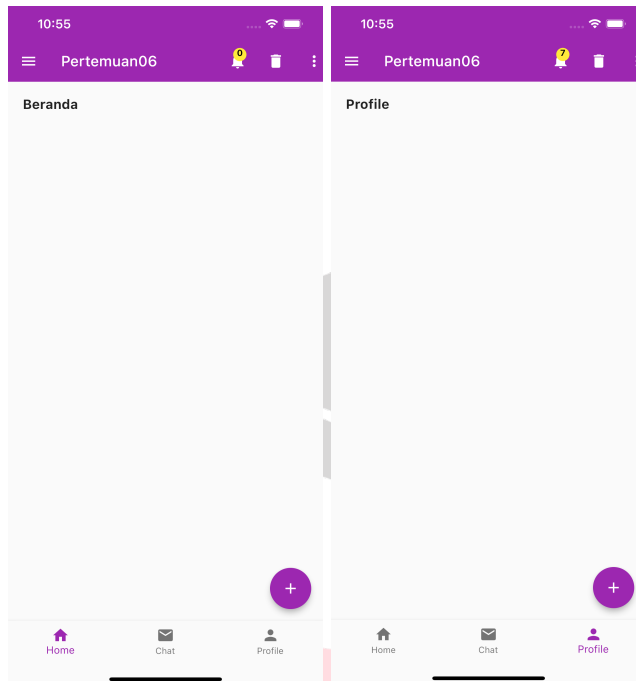
---

Setelah mempelajari dan mempraktikkan modul ini, mahasiswa diharapkan dapat:

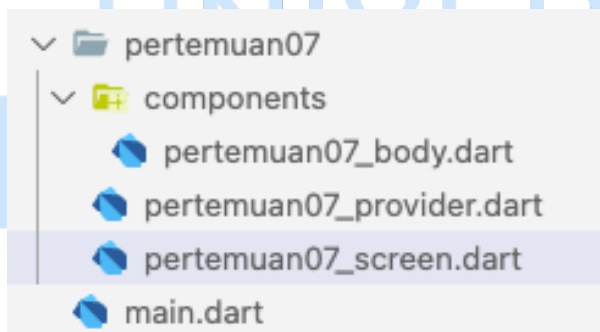
- Mengetahui kegunaan dari AppBar, FAB dan ButtonNavigation.
- Mampu menerapkan AppBar, FAB dan ButtonNavigation pada flutter.

# AppBar, FAB dan ButtonNavigation

## Menerapkan AppBar, FAB dan ButtonNavigation



1. Berikut adalah struktur folder untuk pertemuan 07.



Keterangan:

- pertemuan07\_screen.dart digunakan untuk screen UI flutter (root home)
- pertemuan07\_proivder.dart digunakan untuk provide state tombol FAB
- components > pertemuan07\_body.dart digunakan sebagai **body** UI.

2. Buatlah screen baru dengan nama file **pertemuan07\_screen.dart** pada folder pertemuan07. Import **material**, Gunakan **StatefulWidget** dan gunakan **Scaffold** untuk memulai mendesain komponen. Namai class **statefulWidget** dengan **Pertemuan07Screen**.

```

1  import 'package:flutter/material.dart';
2
3  class Pertemuan07Screen extends StatefulWidget {
4    const Pertemuan07Screen({Key? key}) : super(key: key);
5
6    @override
7    State<Pertemuan07Screen> createState() => _Pertemuan07ScreenState();
8  }
9
10 class _Pertemuan07ScreenState extends State<Pertemuan07Screen> {
11
12   @override
13   Widget build(BuildContext context) {
14 >     return Scaffold( // Scaffold --
15   }
16 }
17
18
19

```

3. Pada scaffold, tambahkan AppBar, dan pada body tambahkan **Padding** untuk mengatur jarak ke dalam komponen.

```

return Scaffold(
  appBar: AppBar( // AppBar --
  body: Padding( // Padding --
); // Scaffold

```

4. Menggunakan **AppBar**. Tambahkan widget **AppBar()** pada property appBar Scaffold.

```

1  appBar: AppBar(
2    leading: const Icon(Icons.menu),
3    title: const Text('Pertemuan06'),
4    actions: [
5      Stack(
6        alignment: Alignment.center,
7        children: [
8          const Icon(Icons.notifications),
9          Positioned(
10           top: 10,
11           right: 0,
12           child: Container(
13             width: 18,
14             height: 18,
15             decoration: BoxDecoration(
16               borderRadius: BorderRadius.circular(20),
17               color: Colors.yellow,
18             ),
19             child: const Text(
20               '1',
21               textAlign: TextAlign.center,
22               style: TextStyle(
23                 fontSize: 12,
24                 color: Colors.black,
25                 fontWeight: FontWeight.bold,
26               ),
27             ),
28           ),
29         ],
30       ),
31       Padding(
32         padding: const EdgeInsets.symmetric(horizontal: 16),
33         child: IconButton(
34           onPressed: () {
35             prov.resetNotif();
36           },
37           icon: const Icon(Icons.delete)),
38     ),
39     const Icon(Icons.more_vert)
40 ],
41 backgroundColor: Colors.purple,
42 elevation: 0,
43 ),

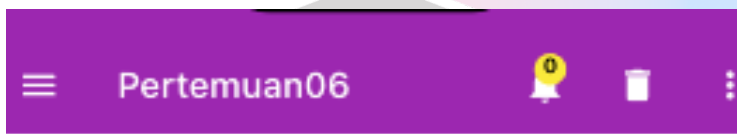
```

Umumnya kita menggunakan property title untuk memberi judul pada AppBar, namun ternyata AppBar memiliki beragam property untuk mendukung desain tampilan AppBar Flutter yaitu: **leading**, **actions**, **backgroundColor**, **elevation**.

Keterangan:

- **leading**, Property yang dapat menempatkan widget pada sebelah kanan AppBar
- **actions**: [], Dapat menerima children berupa list widget.
- **backgroundColor**, untuk memberi warna pada appBar
- **elevation**, mengatur efek shadow pada appBar.

Hasil:



5. Menggunakan **FloatingActionButton()**. Merupakan sebuah jenis button pada flutter yang mengambang pada UI flutter, yang diterapkan pada property **floatingActionButton** Scaffold.



```
1 floatingActionButton: FloatingActionButton(
2     child: const Icon(Icons.add),
3     onPressed: () {},
4 ),
```

### Berinteraksi dengan FAB.

Menggunakan provider, FAB akan menambah ttlNotif pada AppBar setiap pengguna mentrigger tombol FAB.

6. Membuat file provider. Buat sebuah file pertemuan07\_provider.dart dan tambahkan class Pertemuan07Provider extends ChangeNotifier {}



```
1  import 'package:flutter/material.dart';
2
3  class Pertemuan07Provider extends ChangeNotifier {
4    int _ttlNotif = 0;
5
6    int get ttlNotif => _ttlNotif;
7    set setTTLNotif(int val) {
8      _ttlNotif += val;
9      notifyListeners();
10   }
11
12   resetNotif() {
13     _ttlNotif = 0;
14     notifyListeners();
15   }
16 }
```

Keterangan:

- Baris ke-6, adalah sebuah getter untuk mengembalikan nilai dari \_ttlNotif yang akan selalu berubah.
- baris ke-7 s.d 10 adalah sebuah setter untuk menerima value ketika FAB di klik. Ini akan menambah nilai \_ttlNotif dengan value integer yang dikirim, kemudian akan notifyListeners untuk update UI.
- Baris ke -12 s.d 15 adalah sebuah method untuk reset \_ttlNotif ke 0. Method ini akan digunakan nanti pada iconButton delete di actions AppBar.

- Mendaftarkan Provider dan Screen ke **main.dart**. Buka file main.dart, lalu tambahkan **Pertemuan07Provider()** pada MultiProvider dan ganti home Material App ke **Pertemuan07Screen()**.

```

1 Widget build(BuildContext context) {
2   // uncomment ini untuk menggunakan Provider Pertemuan06Provider()
3   // final prov = Provider.of<Pertemuan06Provider>(context);
4   return MaterialApp(
5     title: 'Flutter Demo',
6     debugShowCheckedModeBanner: false,
7     // Gunakan "theme" ini untuk menerapkan enable dark mode sesuai topik materi minggu06 tentang switch.
8
9     // theme: prov.isActive == true ? prov.dark : prov.light,
10    theme: ThemeData(
11      brightness: Brightness.light,
12      primarySwatch: Colors.purple,
13    ),
14    home: const Pertemuan07Screen(),
15  );
16 }

```

```

1 void main() {
2   runApp(MultiProvider(providers: [
3     /// Provider kita pindahkan pada void main agar MyApp dapat menggunakan provider.
4     /// Tambahkan provider disini!
5     /// Provider minggu lalu, ini dihapus jika terjadi error.
6     // ChangeNotifierProvider(create: (_) => Pertemuan05Provider()),
7     // ChangeNotifierProvider(create: (_) => Pertemuan05SCProvider()),
8     // ChangeNotifierProvider(create: (_) => IFBProvider()),
9     // ChangeNotifierProvider(create: (_) => LogicProvider()),
10    // ChangeNotifierProvider(create: (_) => Pertemuan06Provider()),
11
12    //Provider minggu07
13    ChangeNotifierProvider(create: (_) => Pertemuan07Provider()),
14  ], child: const MyApp()));
15 }

```

Keterangan:

- Untuk minggu 07, cukup tambahkan Pertemuan07Provider() seperti pada baris ke-13. Provider lain **jika tdk digunakan**, bisa diberi komentar.

- Menggunakan Provider pada UI Pertemuan07Screen. Inisialisasi provider yang digunakan pada screen dibagian buildWidget.

```

1 @override
2 Widget build(BuildContext context) {
3   final prov = Provider.of<Pertemuan07Provider>(context);

```

Terapkan fungsi **setTTLNotif** provider pada widget FloatingActionButton dibagian onPressed(). Ketika button ditekan maka setTTLNotif akan mengirim value 1 untuk kemudian di tambahkan.



```
1  FloatingActionButton: FloatingActionButton(
2      child: const Icon(Icons.add),
3      onPressed: () {
4          prov.setTTLNotif = 1;
5      }),
```

**Ganti value '1' pada Text** baris ke-12 pada gambar, yang sebelumnya telah diterapkan pada **actions AppBar** dengan prov.ttlNotif.toString(), ini akan menerima value yang diupdate ketika button FAB di klik.



```
1  Positioned(
2      top: 10,
3      right: 0,
4      child: Container(
5          width: 18,
6          height: 18,
7          decoration: BoxDecoration(
8              borderRadius: BorderRadius.circular(20),
9              color: Colors.yellow,
10         ),
11         child: Text(
12             prov.ttlNotif.toString(),
13             textAlign: TextAlign.center,
14             style: const TextStyle(
15                 fontSize: 12,
16                 color: Colors.black,
17                 fontWeight: FontWeight.bold,
18             ),
19         ),
20     ))
```



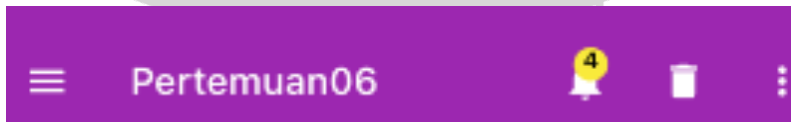
Terapkan **resetNotif()** pada IconButton **Icons.delete** yang berfungsi jika diklik akan reset kembali value ttlNotif ke 0.

```

1  Padding(
2    padding: const EdgeInsets.symmetric(horizontal: 16),
3    child: IconButton(
4      onPressed: () {
5        prov.resetNotif();
6      },
7      icon: const Icon(Icons.delete)),
8  ),

```

Hasil:



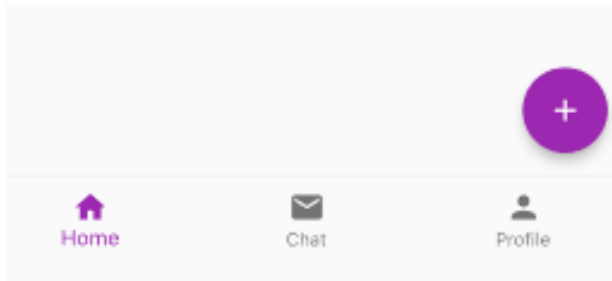
9. Menerapkan **BottomNavigationBar()** pada property bottomNavigationBar Scaffold. Property ini merupakan AppBar Bottom pada flutter yang letaknya di bawah dari UI.

```

1  bottomNavigationBar: BottomNavigationBar(
2    onTap: (val) {},
3    currentIndex: 0,
4    items: const [
5      BottomNavigationBarItem(
6        icon: Icon(Icons.home),
7        label: 'Home',
8      ),
9      BottomNavigationBarItem(
10       icon: Icon(Icons.mail),
11       label: 'Chat',
12     ),
13     BottomNavigationBarItem(
14       icon: Icon(Icons.person),
15       label: 'Profile',
16     )
17   ],
18 ),

```

Tampilan:



Berinteraksi pada **BottomNavigationBar()**. Pada latihan ini, ketika **button Chat** diklik maka akan berganti aktif ke **Chat** dan mengganti body dari aplikasi. Menggunakan state, buatlah default **currentIndex**, buat fungsi **onTap** dan buat **dinamis Body**.



```
1 class _Pertemuan07ScreenState extends State<Pertemuan07Screen> {
2   int _currentIndex = 0;
3
4   void onTabTapped(int index) {
5     setState(() {
6       _currentIndex = index;
7     });
8   }
}
```

Kemudian, terapkan state **\_currentIndex** dan fungsi **onTabTapped** pada **BottomNavigationBar()** yang sebelumnya telah diterapkan.



```
1 bottomNavigationBar: BottomNavigationBar(
2   onTap: onTabTapped,
3   currentIndex: _currentIndex,
4   items: const [
```

10. Buat sebuah komponent untuk letak body. Pada folder components buat file **pertemuan07\_body.dart** dan terapkan sebuah `statelessWidget()` untuk menerima dan menampilkan String Title pada Container > Text.




```

1  import 'package:flutter/material.dart';
2
3  class Pertemuan07Body extends StatelessWidget {
4    final String title;
5    const Pertemuan07Body({Key? key, required this.title}) : super(key: key);
6
7    @override
8    Widget build(BuildContext context) {
9      return Container(
10         child: Text(
11           title,
12           style: const TextStyle(fontWeight: FontWeight.bold, fontSize: 18),
13         ),
14       );
15     }
16   }

```

Kemudian, kembali pada `pertemuan07_screen.dart` dan buat sebuah `list _body` berisi `Pertemuan07Body(title)` pada bagian state.




```

1  final List _body = [
2    const Pertemuan07Body(title: "Beranda"),
3    const Pertemuan07Body(title: "Chat"),
4    const Pertemuan07Body(title: "Profile"),
5  ];

```

Kemudian, terapkan letakkan `list _body` ini pada child `Padding` pada body.

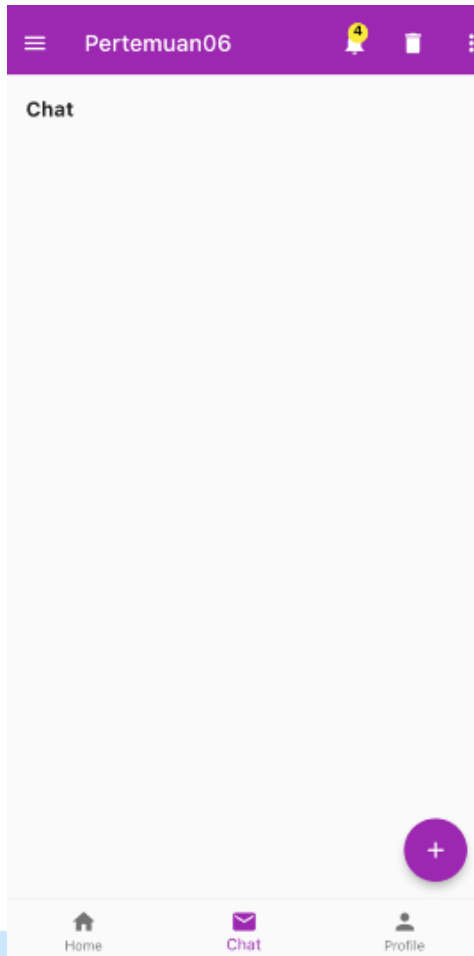


```

1  body: Padding(
2    padding: const EdgeInsets.all(20),
3    child: _body[_currentIndex],
4  ),

```

Hasil: Ketika pengguna klik "Chat" maka body akan tampil untuk menampilkan title "chat". Ini dapat digunakan untuk membuat body yang dinamis pada aplikasi mobile.



## Latihan Praktek

---

Menggunakan AppBar(), BottomNavigationBar(), dan FloatingActionButton(), buatlah sebuah aplikasi sesuai topik Tugas Kelompok.

### Syarat:

- Menerapkan fungsi pada FAB (floatingactionbutton) untuk menampilkan atau mengubah UI menggunakan setState atau Provider().
- Minimal 2 item pada BottomNavigationBar, jika diklik maka body akan berubah untuk menampilkan informasi / widget.
- Title pada AppBar berganti sesuai item BottomNavigationBar yang diklik.

UNIVERSITAS  
MIKROSKIL