

## Modul 6

# Komponen dalam mendesain Aplikasi

### Switches dan Dropdown Buttons

#### Module Overview

---

Mengenal komponen switch dan DropdownButton pada flutter sebagai widget untuk menerima aksi / *trigger* dari pengguna. Switch merupakan sakelar untuk mengaktifkan atau menonaktifkan status suatu item atau kondisi, sedangkan DropdownButton memungkinkan pengguna untuk memilih dari sejumlah item. Tombol menunjukkan item yang telah terpilih saat ini serta tanda panah akan membuka menu untuk memilih item lain.

#### Module Objectives

---

Setelah mempelajari dan mempraktikkan modul ini, mahasiswa diharapkan dapat:

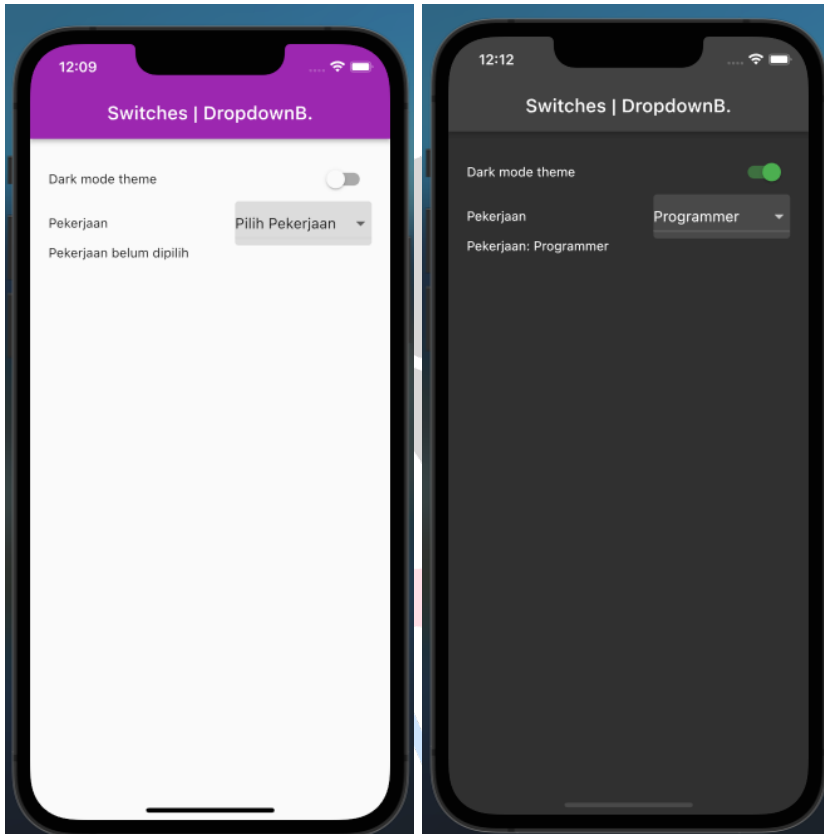
- Mengetahui kapan menggunakan switches dan dropdown button.
- Mampu menerapkan switch dan dropdownbutton pada flutter.

## Switch dan DropDownButton

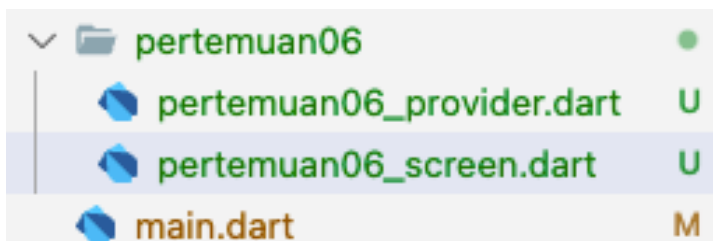
---

**Menerapkan Switch untuk ubah theme aplikasi;**

**Menerapkan DropDownButton untuk memilih pekerjaan;**



1. Pada folder project flutter, buatlah sebuah screen dan provider baru. Berikut adalah struktur folder untuk pertemuan 06.



2. Buatlah screen baru dengan nama file **pertemuan06\_screen.dart** pada folder pertemuan06. Import **material**, Gunakan **StatefulWidget** dan gunakan **Scaffold** untuk memulai mendesain komponen. Namai class statefulWidget dengan **Pertemuan06Screen**.

```

1  import 'package:flutter/material.dart';
2
3  class Pertemuan06Screen extends StatefulWidget {
4    const Pertemuan06Screen({Key? key}) : super(key: key);
5
6    @override
7    State<Pertemuan06Screen> createState() => _Pertemuan06ScreenState()
8  }
9
10 class _Pertemuan06ScreenState extends State<Pertemuan06Screen> {
11   @override
12   Widget build(BuildContext context) {
13 >    return Scaffold( // Scaffold --
14
15   }
16 }

```

3. Pada scaffold, tambahkan AppBar, dan pada body tambahkan **Padding** > **Column**.

```

return Scaffold(
  appBar: AppBar(title: Text('Switches | DropdownB.')),
  body: Padding(
    padding: EdgeInsets.all(20),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        ], // Column
      ), // Padding
    ); // Scaffold

```

4. Menggunakan **Switch**. Tambahkan widget `Switch()` sebagai children dari `column`.

```
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    Text('Dark mode theme'),
    Switch(
      value: false,
      activeColor: Colors.green,
      onChanged: (val) {
        print(val);
      } // Switch
    ),
  ], // Row
```

value pada Switch untuk menyatakan status true / false atau kondisi switch terpilih atau tidak, sedangkan property `onChanged` dapat digunakan untuk mengubah status `Switch()`.



5. Berinteraksi dengan **Switch()** menggunakan **provider** untuk mengubah theme flutter. Buatlah sebuah provider baru untuk pertemuan06 dengan nama file `pertemuan06_provider.dart`. Nama dengan class **Pertemuan06Provider**.

```
1 import 'package:flutter/material.dart';
2
3 class Pertemuan06Provider extends ChangeNotifier {
4   var light = ThemeData(
5     brightness: Brightness.light,
6     primarySwatch: Colors.purple,
7   );
8
9   var dark = ThemeData(
10    brightness: Brightness.dark,
11    primarySwatch: Colors.purple,
12  );
13
14  bool _enableDarkMode = false;
15
16  bool get enableDarkMode => _enableDarkMode;
17
18  set setBrightness(val) {
19    _enableDarkMode = val;
20    notifyListeners();
21  }
22 }
```

mula-mula, inialisasi **ThemeData()** untuk kondisi light dan dark untuk nanti digunakan pada bagian **MaterialApp** di main.dart.

Pada bari ke 14 inialisasi kondisi mula-mula enableDarkMode sebagai value pada **Switch()** yang tadi telah dibuat.

Pada baris ke 18, gunakan set untuk mengubah status switch dan notifylisteners() untuk memperbaharui state.

- Gunakan provider yang telah ditambahkan pada Switch().  
Tambahkan inialisasi berikut agar penulisan provider jauh lebih ringkas.

```
final prov = Provider.of<Pertemuan06Provider>(context);
```

Tambahkan pada widget build(), seperti gambar berikut:

```
@override
Widget build(BuildContext context) {
  final prov = Provider.of<Pertemuan06Provider>(context);
  return Scaffold(
```

Ganti value dan kondisi onChanged class Pertemuan06Provider, seperti gambar berikut:

```
//Switch
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    Text('Dark mode theme'),
    Switch(
      value: prov.enableDarkMode,
      activeColor: Colors.green,
      onChanged: (val) {
        prov.setBrightness = val;
      }) // Switch
  ],
), // Row
```

- Gunakan Provider pada **main.dart** untuk mengatur theme pada **MaterialApp**. Pertama, ubah letak dari **MultiProvider** pada method main() agar class MyApp dapat menggunakan provider.

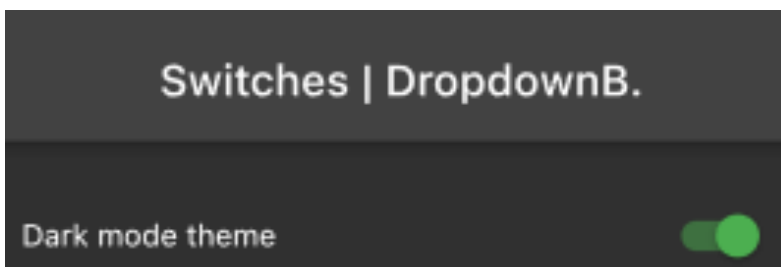
```
void main() {
  runApp(MultiProvider(providers: [
    /// Provider kita pindahkan pada void main agar MyApp dapat
    /// menggunakan provider.
    /// Tambahkan provider disini!
    ChangeNotifierProvider(create: (_) => Pertemuan06Provider()),
  ], child: const MyApp())); // MultiProvider
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    final prov = Provider.of<Pertemuan06Provider>(context);
    return MaterialApp(
      title: 'Flutter Demo',
      debugShowCheckedModeBanner: false,
      theme: prov.enableDarkMode == true ? prov.dark : prov.light,
      home: Pertemuan06Screen(),
    ); // MaterialApp
  }
}
```

theme pada MaterialApp akan disesuaikan dengan status enableDarkMode yang dipicu melalui Swithc().

Restart flutter untuk melihat hasilnya.



8. Menggunakan **DropDownButton** menggunakan **setState**.

Mula-mula buat, state yang nantinya akan digunakan pada dropdownbutton.

```
class _Pertemuan06ScreenState extends State<Pertemuan06Screen> {  
  //List Item  
  String itemSelected = 'Pilih Pekerjaan';  
  List items = [  
    'Pilih Pekerjaan',  
    'Mahasiswa',  
    'Dosen',  
    'Programmer',  
    'UI UX Desainer',  
    'IT Consultan',  
    'Project Manager',  
    'PNS',  
    'Wiraswasta'  
  ];  
}
```

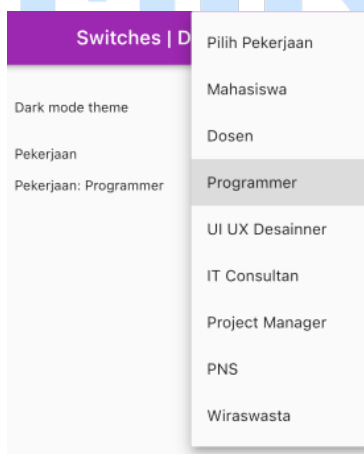
**String itemSelected** digunakan untuk menyatakan value atau state dari dropdownbutton nantinya.

**List items** dengan array berisi sekumpulan item yang digunakan untuk dropdownbutton.

Tambahkan Widget DropdownButton pada children column.

```
//DropdownButton
-Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    const Text('Pekerjaan'),
    DropdownButton(
      items: items.map((item) {
        return DropdownMenuItem(
          child: Text(item),
          value: item,
        ); // DropdownMenuItem
      }).toList(),
      value: itemSelected,
      onChanged: (val) {
        setState(() {
          itemSelected = val as String;
        });
      },
    ) // DropdownButton
  ],
), // Row
```

pada property **items**, mapping isi dari array sebagai item dari **DropdownMenuItem()**. value untuk menyatakan state dari DropDownButton saat ini. onChanged untuk mengubah kondisi sesuai pilihan item dari pengguna.





## Latihan Praktek

---

1. Menggunakan widget **Switch()** untuk menampilkan DropdownButton() dibawahnya, mula-mula kondisi off, jika turn on maka DropDownButton() akan tampil.
2. Tambahkan **DropDownButton()** untuk list 5 cita-cita anda. Ketika salah satu item terpilih maka akan tampil rincian cita-cita anda, untuk style bebas, gunakan layouting yang baik, gambar, text atau widget-widget lain yang telah dipelajari sebelumnya..

UNIVERSITAS  
MIKROSKIL