

## Modul 5

# Komponen dalam Mendesain Aplikasi

CheckBox, RadioButtons, Chips

## Module Overview

---

CheckBox, RadioButton dan Chips adalah widget pada flutter yang memungkinkan user untuk memberi aksi (trigger) untuk kondisi UI yang membutuhkan komponen opsi cek, opsi radio atau chips

## Module Objectives

---

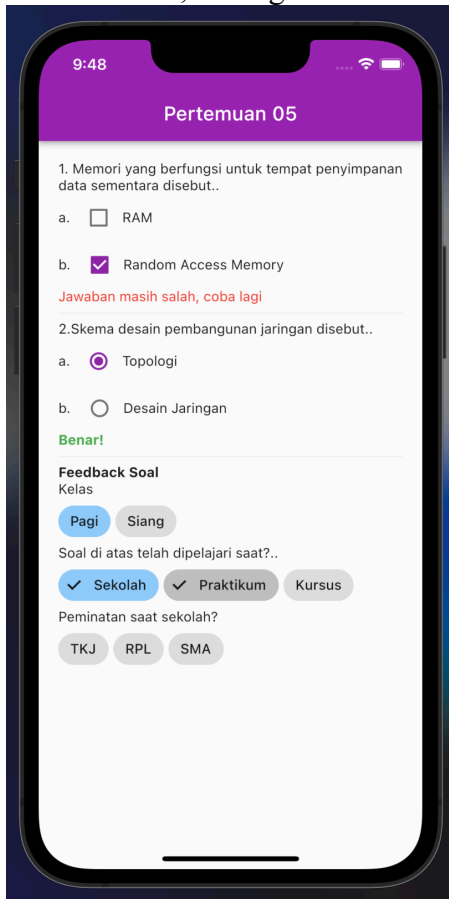
Setelah mempelajari dan mempraktikkan modul ini, mahasiswa diharapkan dapat:

- Menerapkan widget checkbox, radiobutton dan chip pada UI flutter
- Menerapkan style pada widget checkbox, radiobutton dan chip
- Menerapkan setState atau Provider pada widget

## CheckBox, RadioButton dan Chips

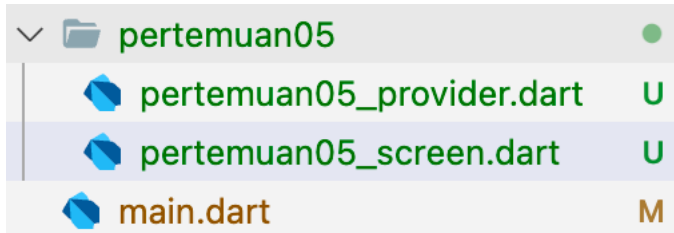
### Membuat aplikasi quiz

Aplikasi ini menggunakan menerapkan widget checkbox, radiobutton dan chip pada UI / Screen flutter, sedangkan untuk handel UI state menggunakan setState() dan Provider.



Langkah-langah pembuatan:

1. Buatlah sebuah folder dan file dengan struktur file seperti gambar berikut:



pada folder **pertemuan05** buat 2 file, **pertemuan5\_provider.dart** untuk provider dan **pertemuan05\_screen.dart** sebagai screen.

2. Buatlah sebuah screen menggunakan **statefulWidget** dengan nama class **Pertemuan05Screen()**.

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

class Pertemuan05Screen extends StatefulWidget {
  Pertemuan05Screen({Key? key}) : super(key: key);

  @override
  State<Pertemuan05Screen> createState() => _Pertemuan05ScreenState();
}

class _Pertemuan05ScreenState extends State<Pertemuan05Screen> {
  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

3. Gunakan **scaffold** pada widget build agar aplikasi dapat menggunakan material component. **Ganti return container()** dengan scaffold.

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar( // AppBar ...
    body: SingleChildScrollView( //
    ); // Scaffold
  );
}
```

4. Pada property appBar gunakan Widget **AppBar()** untuk menampilkan title.

```
appBar: AppBar(
  title: Text('title'),
),
```

5. Pada property body, tambahkan gunakan Widget `SingleChildScrollView() > Padding() > Column()`

```
-body: SingleChildScrollView(
  child: Padding(
    padding: EdgeInsets.all(20),
    child: Column( // Column ...
  )), // Padding // SingleChildScrollView
```

6. Pada widget column, beberapa widget akan ditambahkan seperti checkbox, radiobutton, dan chips, berikut adalah gambaran besarnya.

```
-body: SingleChildScrollView(
  scrollDirection: Axis.vertical,
  child: Padding(
    padding: EdgeInsets.all(20),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        //Soal 1: Menggunakan Checkbox
        const Text( // Text --
        Row( // Row --
        Row( // Row --

        //Respon jawaban soal1
        if (soal1a == false && soal1b == false)
          Container()
        else if (soal1a == true && soal1b == true)
          const Text( // Text --
        else
          const Text( // Text --

        const Divider(),
        //Soal 2: Menggunakan Checkbox
        const Text('2.Skema desain pembangunan jaringan disebut..'),
        Row( // Row --
        Row( // Row --

        //CheckJawaban
        if (soal2 == 'answ')
          Container()
        else if (soal2 == 'topologi')
          const Text( // Text --
        else
          const Text( // Text --

        //Chips
        //ChoiceChip
        const Divider(),
        const Text('Feedback Soal', // Text --
        const Text('Kelas'),
        Row( // Row --
        const Text('Soal di atas telah dipelajari saat?..'),
        Row( // Row --

        //InputChip
        const Text('Peminatan saat sekolah?'),
        Container( // Container --
        Row( // Row --

      ],
    ), // Column
```

7. Menggunakan **widget CheckBox**. Lengkapi di bawah baris "Soal 1: Menggunakan checkbox".

```
//Soal 1: Menggunakan Checkbox
const Text(
  '1. Memori yang berfungsi untuk tempat penyimpanan data sementara disebut..'), // Text
Row(
  children: [
    Text('a.'),
    SizedBox(width: 5),
    Checkbox(
      value: soal1a,
      onChanged: (val) {
        setState(() {
          soal1a = val;
        });
      },
    ), // Checkbox
    Text('RAM'),
  ],
), // Row
Row(
  children: [
    Text('b.'),
    SizedBox(width: 5),
    Checkbox(
      value: soal1b,
      onChanged: (val) {
        setState(() {
          soal1b = val;
        });
      },
    ), // Checkbox
    Text('Random Access Memory'),
  ],
), // Row
```

Pada masing-masing checkbox, dibutuhkan value dan onChange untuk memperbaharui kondisi checkbox. Pada case ini digunakan state dan setState untuk listen perubahan dan mempengaruhi UI.

Atur **state soal1a dan soal1b** pada class state screen ini.

```
class _Pertemuan05ScreenState extends State<Pertemuan05Screen> {
  //Status soal 1
  bool? soal1a = false;
  bool? soal1b = false;
```

Hasil:

1. Memori yang berfungsi untuk tempat penyimpanan data sementara disebut..

- a. ☒ RAM
- b. ☐ Random Access Memory

8. Menggunakan **widget RadioButton()**.

Di bawah bari "Soal 2: Menggunakan CheckBox", tambahkan **widget RadioButton**

//Soal 2: Menggunakan Checkbox

```
—const Text('2.Skema desain pembangunan jaringan disebut..'),
```

```
—Row(  
  |
```

```
  | children: [  
  |
```

```
  |   Text('a.'),  
  |
```

```
  |   SizedBox(width: 5),  
  |
```

```
  |   Radio(  
  |
```

```
  |     value: 'topologi',  
  |
```

```
  |     groupValue: soal2,  
  |
```

```
  |     onChanged: (val) {  
  |
```

```
  |       setState(() {  
  |
```

```
  |         soal2 = 'topologi';  
  |
```

```
  |       });  
  |
```

```
  |     },  
  |
```

```
  |   ), // Radio  
  |
```

```
  |   Text('Topologi'),  
  |
```

```
  | ],  
  |
```

```
  | ), // Row  
  |
```

```
—Row(  
  |
```

```
  | children: [  
  |
```

```
  |   Text('b.'),  
  |
```

```
  |   SizedBox(width: 5),  
  |
```

```
  |   Radio(  
  |
```

```
  |     groupValue: soal2,  
  |
```

```
  |     value: 'desain jaringan',  
  |
```

```
  |     onChanged: (val) {  
  |
```

```
  |       setState(() {  
  |
```

```
  |         soal2 = 'desain jaringan';  
  |
```

```
  |       });  
  |
```

```
  |     },  
  |
```

```
  |   ), // Radio  
  |
```

```
  |   Text('Desain Jaringan'),  
  |
```

```
  | ],  
  |
```

```
  | ), // Row
```

Ingat! kondisi opsi radio yaitu hanya salah satu, atau single answer. Maka tambahkan state berikut pada class state:

```
class _Pertemuan05ScreenState extends State<Pertemuan05Screen> {
  //Status soal 1
  bool? soal1a = false;
  bool? soal1b = false;

  //Status soal 2
  String soal2 = 'answ';
}
```

Gunakan value yang tidak sesuai pada pilihan value radiobutton agar state awal radiobutton tidak terpilih.

Hasil:

2. Skema desain pembangunan jaringan disebut..

- a. ☐ Topologi
- b. ☒ Desain Jaringan

9. Menggunakan ChoiceChip. Tambahkan Widget ini dibawah text("Kelas").

```
-Row(
  children: [
    ChoiceChip(
      label: Text('Pagi'),
      selectedColor: Colors.blue[200],
      selected: kelasPagi,
      onSelect: (val) {
        setState(() {
          kelasPagi = val;
        });
      },
    ), // ChoiceChip
    const SizedBox(width: 5),
    ChoiceChip(
      label: Text('Siang'),
      selectedColor: Colors.blue[200],
      selected: kelasSiang,
      onSelect: (val) {
        setState(() {
          kelasSiang = val;
        });
      },
    ), // ChoiceChip
  ],
), // Row
```

SITAS  
MIKROSKIL

Atur state untuk provide kondisi mula-mula choiceChip.

```
class _Pertemuan05ScreenState extends State<Pertemuan05Screen> {  
  //Status soal 1  
  bool? soall1a = false;  
  bool? soall1b = false;  
  
  //Status soal 2  
  String soal2 = 'answ';  
  
  //Status pilih kelas  
  bool kelasPagi = false;  
  bool kelasSiang = false;  
}
```

#### 10. Menggunakan widget **FilterChip** dengan **Provider**.

Buat sebuah file untuk provider, **pertemuan05\_provider.dart**

```
import 'package:flutter/material.dart';  
  
class Pertemuan05Provider extends ChangeNotifier {  
  // status soal telah dipelajari saat?  
  
  // Inisialisasi nilai awal  
  bool _diSekolah = false;  
  bool _diPraktik = true;  
  bool _diKursus = false;  
  
  // Ini akan return nilai dari disekolah, dipraktik, dst. Ingat konsep OOP setter || getter  
  bool get statusSekolah => _diSekolah;  
  bool get statusPraktik => _diPraktik;  
  bool get statusKursus => _diKursus;  
  
  // Perubahan state, ingat konsep Listen pada StateManajemen  
  set setSekolah(val) {  
    _diSekolah = val;  
    notifyListeners();  
  }  
  
  set setPraktik(val) {  
    _diPraktik = val;  
    notifyListeners();  
  }  
  
  set setKursus(val) {  
    _diKursus = val;  
    notifyListeners();  
  }  
}
```

Class **Pertemuan05Provider** menggunakan **ChangeNotifier** agar bisa memanggil **notifyListener()**. Pada class ini semua state, listener atau method di tulis.



Agar bisa menggunakan Provider ini, maka daftar pada root widgetnya yaitu pada file **main.dart**.

```
Run | Debug | Profile
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return MultiProvider(
      providers: [
        // Tambahkan provider disini!
        ChangeNotifierProvider(create: (_) => Pertemuan05Provider()),
      ],
      child: MaterialApp(
        title: 'Flutter Demo',
        debugShowCheckedModeBanner: false,
        theme: ThemeData(
          primarySwatch: Colors.purple,
        ), // ThemeData
        home: Pertemuan05Screen(), // MaterialApp
      ); // MultiProvider
    }
  }
}
```

Gunakan widget **MultiProvider** dari package provider, kemudian tambahkan provider **Pertemuan05Provider()**. Pastikan default **home** yaitu **Pertemuan05Screen()**.

### Tambahkan FilterChips()

```
Row(
  children: [
    FilterChip(
      label: Text('Sekolah'),
      //Atur color disini
      selectedColor: Colors.blue[200],
      selected: prov.statusSekolah,
      onSelected: (val) {
        prov.setSekolah = val;
      },
    ), // FilterChip
    const SizedBox(width: 5),
    FilterChip(
      label: Text('Praktikum'),
      selected: prov.statusPraktik,
      onSelected: (val) {
        prov.setPraktik = val;
      },
    ), // FilterChip
    const SizedBox(width: 5),
    FilterChip(
      label: Text('Kursus'),
      selected: prov.statusKursus,
      onSelected: (val) {
        prov.setKursus = val;
      },
    ), // FilterChip
  ],
), // Row
```

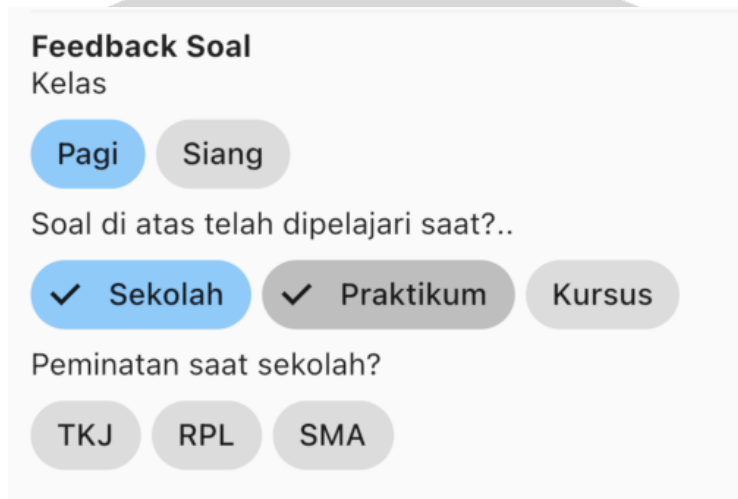
Gunakan provider pada setiap FilterChips. Karena provider sering digunakan pada class ini, maka definisi Provider dengan prov seperti gambar berikut:

```
@override
Widget build(BuildContext context) {
  // class ini menggunakan provider Pertemuan05Provider
  final prov = Provider.of<Pertemuan05Provider>(context);

  return Scaffold(
    appBar: AppBar( // AppBar ...
```

Disini prov dapat memanggil seluruh property pada **Pertemuan05Provider**. sehingga, untuk menggunakan state statusPraktik cukup dengan **prov.statusPratik**.

Hasil:



**Feedback Soal**

Kelas

☒ Pagi ☐ Siang

Soal di atas telah dipelajari saat?..

☒ Sekolah ☒ Praktikum ☐ Kursus

Peminatan saat sekolah?

☐ TKJ ☐ RPL ☐ SMA

### Latihan:

1. Menggunakan **Provider** atau **setState()** lengkapi InputChips dibawah baris "Peminatan saat sekolah?" agar InputChips ini dapat berinteraksi dengan ketentuan:

Ketentuan:

- Menggunakan Provider.
- Ketika chips di klik, maka statusnya true: "Berwarna biru".
- Selain berwarna biru, chips akan bertambah pada Container.

```
//InputChip
const Text('Peminatan saat sekolah?'),
Container(
  child: Row(
    children: [
      Container(),
      // letakkan chip terpilih disini!
      // atur state menggunakan setState atau Provider
    ],
  ), // Row
), // Container

Row(
  children: [
    InputChip(
      label: Text('TKJ'),
      onPressed: (val) {
        print(val);
        //atur state menggunakan setState atau Provider
      },
    ), // InputChip
    SizedBox(width: 5),
    InputChip(
      label: Text('RPL'),
      onPressed: (val) {
        print(val);
        //atur state menggunakan setState atau Provider
      },
    ), // InputChip
    SizedBox(width: 5),
    InputChip(
      label: Text('SMA'),
      onPressed: (val) {
        print(val);
        //atur state menggunakan setState atau Provider
      },
    ), // InputChip
  ],
), // Row
```