

Modul 15

Komponen dalam mendesain Aplikasi

Adding Package: ImagePicker, Carousel, dan

WebView

Module Overview

Mengenal dan mengimplementasikan plugin atau package flutter yang terdapat di website pub.dev pada aplikasi flutter.

Module Objectives

Setelah mempelajari dan mempraktikkan modul ini, mahasiswa diharapkan dapat:

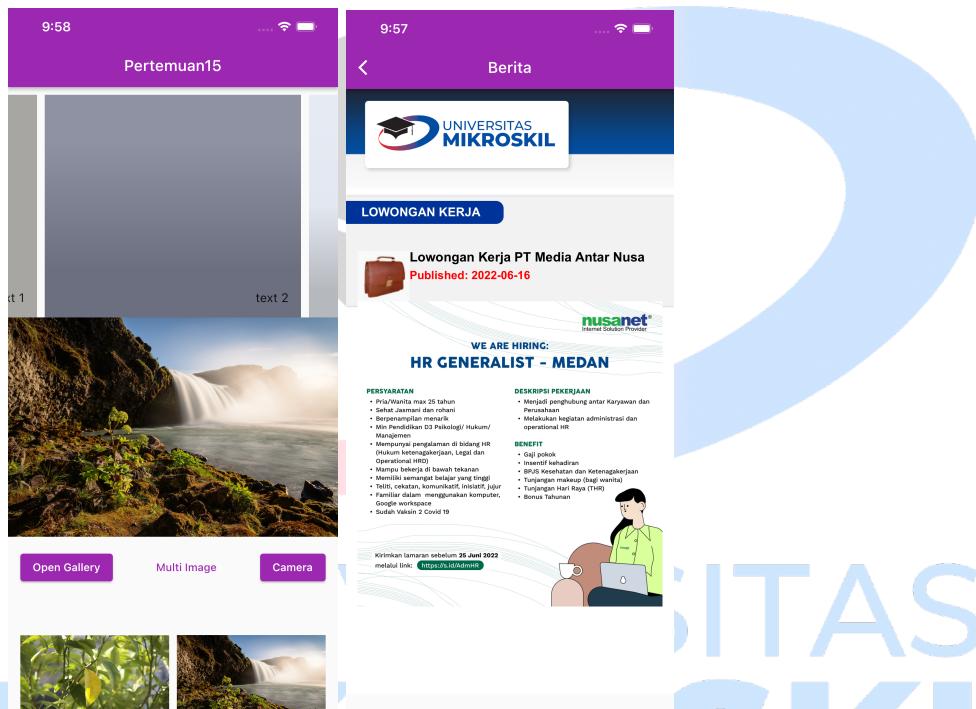
- Mengetahui cara menginstall plugin dari pub.dev.
- Mampu menerapkan plugin / package pada aplikasi flutter.

ImagePicker, Carousel dan WebView

Menerapkan Widget ImagePicker, Carousel dan WebView.

Keterangan:

Luaran aplikasi ini akan memiliki fitur untuk memilih gambar dari gallery atau capture dari fitur kamera, menampilkan gambar dengan carousel dan fitur webview.



1. Berikut adalah struktur folder untuk pertemuan 15.

✓	└ pertemuan15	•
✓	└ components	•
↳	button_image_picker.dart	U
↳	carousel_slider.dart	U
↳	webview_widget.dart	U
↳	pertemuan15_provider.dart	U
↳	pertemuan15_screen.dart	U

Keterangan:

- pertemuan15_screen.dart merupakan file untuk screen UI flutter (root home)
 - pertemuan15_provider.dart merupakan file untuk logic yang di provide untuk view.
 - pada folder component, terdapat beberapa widget yang untuk kebutuhan webview, carousel slider dan tombol image picker.
2. Buatlah screen baru dengan nama file **pertemuan15_screen.dart** pada folder pertemuan15. Import **material**, Gunakan **StatefulWidget**, namai class **statefulWidget** dengan **Pertemuan15Screen** dan gunakan **Scaffold** untuk memulai mendesain komponen.



```
1 import 'package:flutter/material.dart';
2
3 class Pertemuan15Screen extends StatefulWidget {
4   const Pertemuan15Screen({Key? key}) : super(key: key);
5
6   @override
7   State<Pertemuan15Screen> createState() => _Pertemuan15ScreenState();
8 }
9
10 class _Pertemuan15ScreenState extends State<Pertemuan15Screen> {
11   @override
12   Widget build(BuildContext context) {
13     return Scaffold(
14       appBar: AppBar(
15         title: const Text('Pertemuan15'),
16       ),
17       body: const Text('Pertemuan15'));
18   }
19 }
```

Keterangan:

- Update **main.dart** menjadi "home: Pertemuan15Screen()"
- Pada terminal root folder ketik "**flutter run**" untuk menjalankan aplikasi.

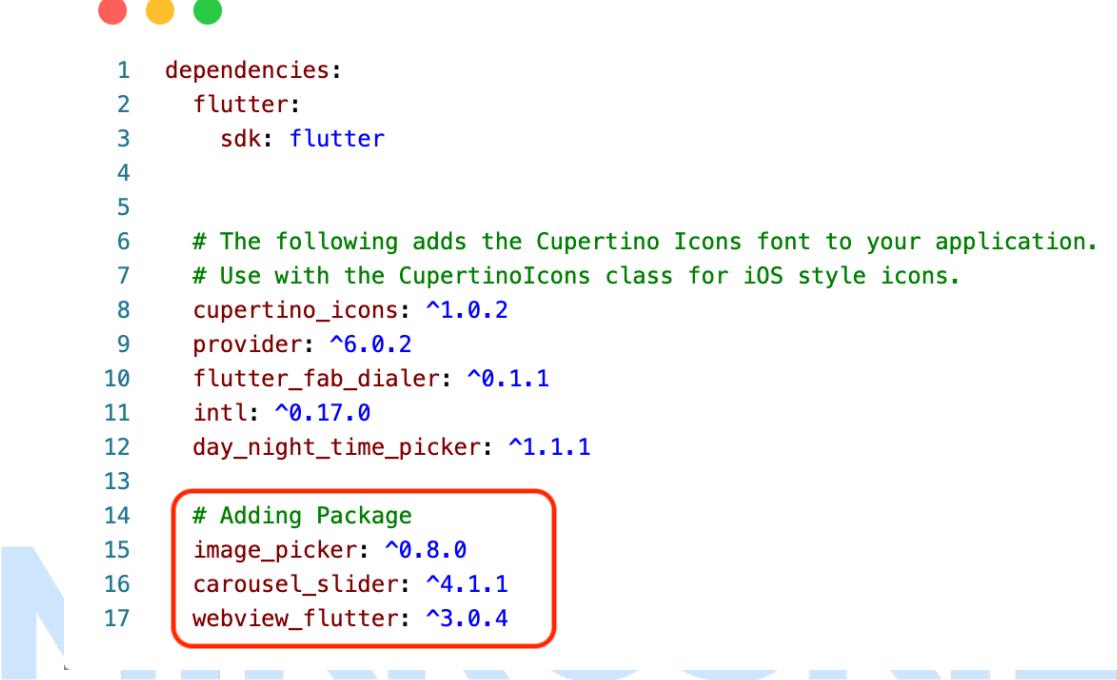
3. Buatlah class untuk provider pada file pertemuan15provider.dart dan tambahkan pada folder main.dart, seperti gambar berikut:



4. **Adding Package.** Pada pubspec.yaml tambahkan package version yang akan ditambahkan:

- https://pub.dev/packages/image_picker
- https://pub.dev/packages/carousel_slider
- https://pub.dev/packages/webview_flutter

Catatan: Silahkan baca dokumentasi pada tab read me sebelum menggunakan setiap plugin, berikutnya tambahkan pada file **pubspec.yaml** seperti gambar berikut:



```
1 dependencies:
2   flutter:
3     sdk: flutter
4
5
6   # The following adds the Cupertino Icons font to your application.
7   # Use with the CupertinoIcons class for iOS style icons.
8   cupertino_icons: ^1.0.2
9   provider: ^6.0.2
10  flutter_fab_dialer: ^0.1.1
11  intl: ^0.17.0
12  day_night_time_picker: ^1.1.1
13
14  # Adding Package
15  image_picker: ^0.8.0
16  carousel_slider: ^4.1.1
17  webview_flutter: ^3.0.4
```

Simpan (*otomatis akan terinstall*), jika tidak otomatis, maka pada terminal ketik: **flutter pub get**. Kemudian, jalankan ulang "**flutter run**"

5. **Menggunakan image_picker.** Sesuai dokumentasi pada image_picker, maka perlu menyiapkan fungsi untuk pilih gambar dari gallery atau camera. Lengkapi fungsi tersebut pada provider, seperti gambar berikut.



```
1 import 'package:flutter/material.dart';
2 import 'package:image_picker/image_picker.dart';
3
4 class Pertemuan15Provider extends ChangeNotifier {
5     bool _isImageLoaded = false;
6     bool get isImageLoaded => _isImageLoaded;
7     set setIsImageLoaded(val) {
8         _isImageLoaded = val;
9         notifyListeners();
10    }
11
12 //ImagePicker
13 final ImagePicker _imagePicker = ImagePicker();
14
15 XFile? _img;
16 XFile? get img => _img;
17 set setImg(val) {
18     _img = val;
19     notifyListeners();
20 }
21
22 pickImage(bool isGallery) async {
23     try {
24         var res = await _imagePicker.pickImage(
25             source: isGallery == true ? ImageSource.gallery : ImageSource.camera);
26         if (res != null) {
27             setImg = res;
28             setIsImageLoaded = true;
29         }
30     } catch (e) {
31         setIsImageLoaded = false;
32         print(e.toString());
33     }
34 }
35 }
36 }
```



Keterangan:

- **Baris 5 - 10**, adalah sebuah setter dan getter yang digunakan untuk set status mula-mula apakah ada gambar atau tidak.
- **Baris 13**, sebuah initialisasi ImagePicker.

- **Baris 15 - 20**, Setter dan getter file gambar dengan tipe data **XFile**.
- **Baris 22 - 34**, sebuah method untuk menjalankan fungsi memilih gambar dari gallery atau camera (tergantung pilihan user).

6. lengkapi button_image_picker.dart dengan **stateless** dan **elevated button**, seperti gambar berikut:



```
1 import 'package:flutter/material.dart';
2 import 'package:flutter_application/pertemuan15/pertemuan15_provider.dart';
3 import 'package:provider/provider.dart';
4
5 class ButtonImagePicker extends StatelessWidget {
6   final String title;
7   final bool isGallery;
8   const ButtonImagePicker(
9     {Key? key, required this.isGallery, required this.title})
10    : super(key: key);
11
12   @override
13   Widget build(BuildContext context) {
14     final prov = Provider.of<Pertemuan15Provider>(context);
15     return ElevatedButton(
16       onPressed: () async {
17         await prov.pickImage(isGallery);
18       },
19       child: Text(title));
20   }
21 }
```



- ElevatedButton akan mengeksekusi fungsi pickImage yang tadi telah disediakan pada provider.

Letakkan **ButtonImagePicker()** sebagai child dari listview pada body seperti gambar berikut:



```
1  @override
2  Widget build(BuildContext context) {
3      final prov = Provider.of<Pertemuan15Provider>(context);
4      return Scaffold(
5          appBar: AppBar(
6              title: const Text('Pertemuan15'),
7          ),
8          body: ListView(
9              children: [
10                  //Image Picker
11                  prov.isImageLoaded == true
12                      ? Image.file(File(prov.img!.path))
13                      : Container(),
14                  Padding(
15                      padding: const EdgeInsets.all(16.0),
16                      child: Row(
17                          mainAxisAlignment: MainAxisAlignment.spaceBetween,
18                          children: const [
19                              ButtonImagePicker(
20                                  isGallery: true,
21                                  title: 'Open Gallery',
22                              ),
23                              ButtonImagePicker(
24                                  isGallery: false,
25                                  title: 'Camera',
26                              ),
27                          ],
28                      ),
29                  ),
30                  ],
31              ),
32          );
33      );
34  }
35
```



Keterangan:

- **baris 11 -13**, akan menampilkan gambar jika ada gambar terpilih. Gambar akan ditampilkan menggunakan **Image.file()**

- reload flutter dan silahkan coba 2 tombol tersebut untuk memilih gambar seperti gambar berikut.



Open Gallery

Camera

7. Menggunakan fungsi **multimage** pada **imagePicker** dan menampilkan list gambar menggunakan **grid**.

Tambahkan variable listImage dalam bentuk list<XFile> sebagai local state seperti gambar berikut.



```
1 class _Pertemuan15ScreenState extends State<Pertemuan15Screen> {  
2     List<XFile>? listImg;  
3  
4     @override  
5     Widget build(BuildContext context) {
```

Tambahkan sebuah textButton diantara BottonImagePicker, seperti gambar berikut: ****cukup ketik yang bagian tanda merah.**



```
1 const ButtonImagePicker(  
2   isGallery: true,  
3   title: 'Open Gallery',  
4 ),  
5 TextButton(  
6   onPressed: () async {  
7     var pilihGambar = ImagePicker();  
8     var hasil = await pilihGambar.pickMultiImage();  
9  
10    if (hasil == null) {  
11      print('Tidak ada foto');  
12    } else {  
13      setState(() {  
14        listImg = hasil;  
15      });  
16    }  
17  },  
18  child: Text('Multi Image'),  
19 ),  
20 const ButtonImagePicker(  
21   isGallery: false,  
22   title: 'Camera',  
23 ),
```



Keterangan:

- Ini adalah contoh jika ingin menggunakan imagePicker dengan konsep setState local dari flutter.
- Sama seperti imagePicker sebelumnya, perlu pengecekan apakah ada **hasil** foto terpilih. Jika ada maka ubah state listImg ke hasil.

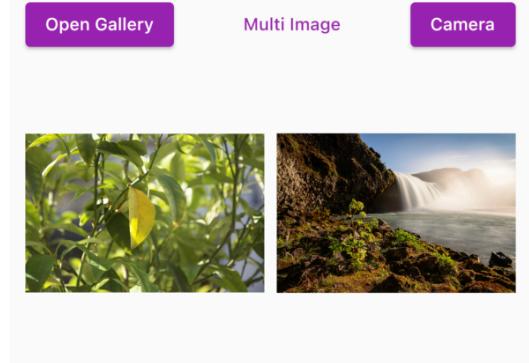
Menampilkan List<XFile> dalam grid.

Letakkan kode berikut dibawah button sebagai child dari Column.



```
1  listImg != null
2      ? Padding(
3          padding: const EdgeInsets.all(16.0),
4          child: SizedBox(
5              height: 300,
6              width: MediaQuery.of(context).size.width,
7              child: GridView.count(
8                  crossAxisCount: 2,
9                  crossAxisSpacing: 10,
10                 mainAxisSpacing: 4,
11                 children: listImg!.map((e) {
12                     return Image.file(File(e.path));
13                 }).toList(),
14             )));
15 : Container(),
```

reload untuk melihat hasilnya:



8. Menggunakan **carousel**. Pada file **carousel_slider.dart**, tambahkan caraousel seperti gambar berikut:



Keterangan:

- **CarouselSlider** membutuhkan list item, maka pada implementasi di atas item dari carousel adalah list yang di map untuk return sebuah builder.

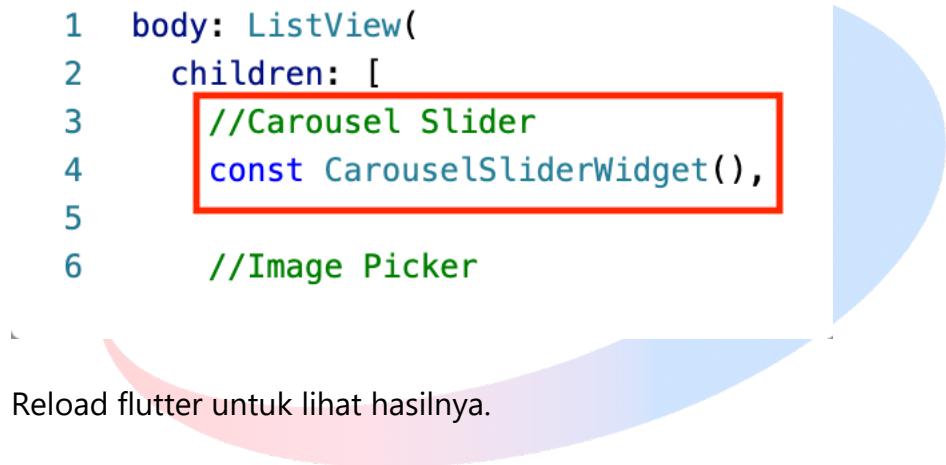
- Sebagai contoh, maka pada container diterapkan **DecorationImage** yang berasal dari **picsum.photos**.

Tambahkan Widget yang telah dibuat pada tahapan sebelumnya pada pertemuan15screen sebagai child dari listview



```
1 body: ListView(  
2   children: [  
3     //Carousel Slider  
4     const CarouselSliderWidget(),  
5  
6     //Image Picker
```

Reload flutter untuk lihat hasilnya.



UNIVERSITAS
MIKROSKIL

Pertemuan15



Open Gallery

Multi Image

Camera



9. **Menggunakan ListView.** Tambahkan ListView pada sebuah halaman yaitu webview_widget.dart seperti gambar berikut:



```
1 import 'dart:io';
2
3 import 'package:flutter/material.dart';
4 import 'package:webview_flutter/webview_flutter.dart';
5
6 class WebViewWidget extends StatefulWidget {
7   final String url;
8   const WebViewWidget({Key? key, required this.url}) : super(key: key);
9
10  @override
11  State<WebViewWidget> createState() => _WebViewWidgetState();
12 }
13
14 class _WebViewWidgetState extends State<WebViewWidget> {
15   @override
16   void initState() {
17     super.initState();
18     // Enable virtual display.
19     if (Platform.isAndroid) WebView.platform = AndroidWebView();
20   }
21
22   @override
23   Widget build(BuildContext context) {
24     return Scaffold(
25       appBar: AppBar(
26         title: const Text('Berita'),
27       ),
28       body: SafeArea(child: WebView(initialUrl: widget.url)),
29     );
30   }
31 }
32 }
```

Keterangan:

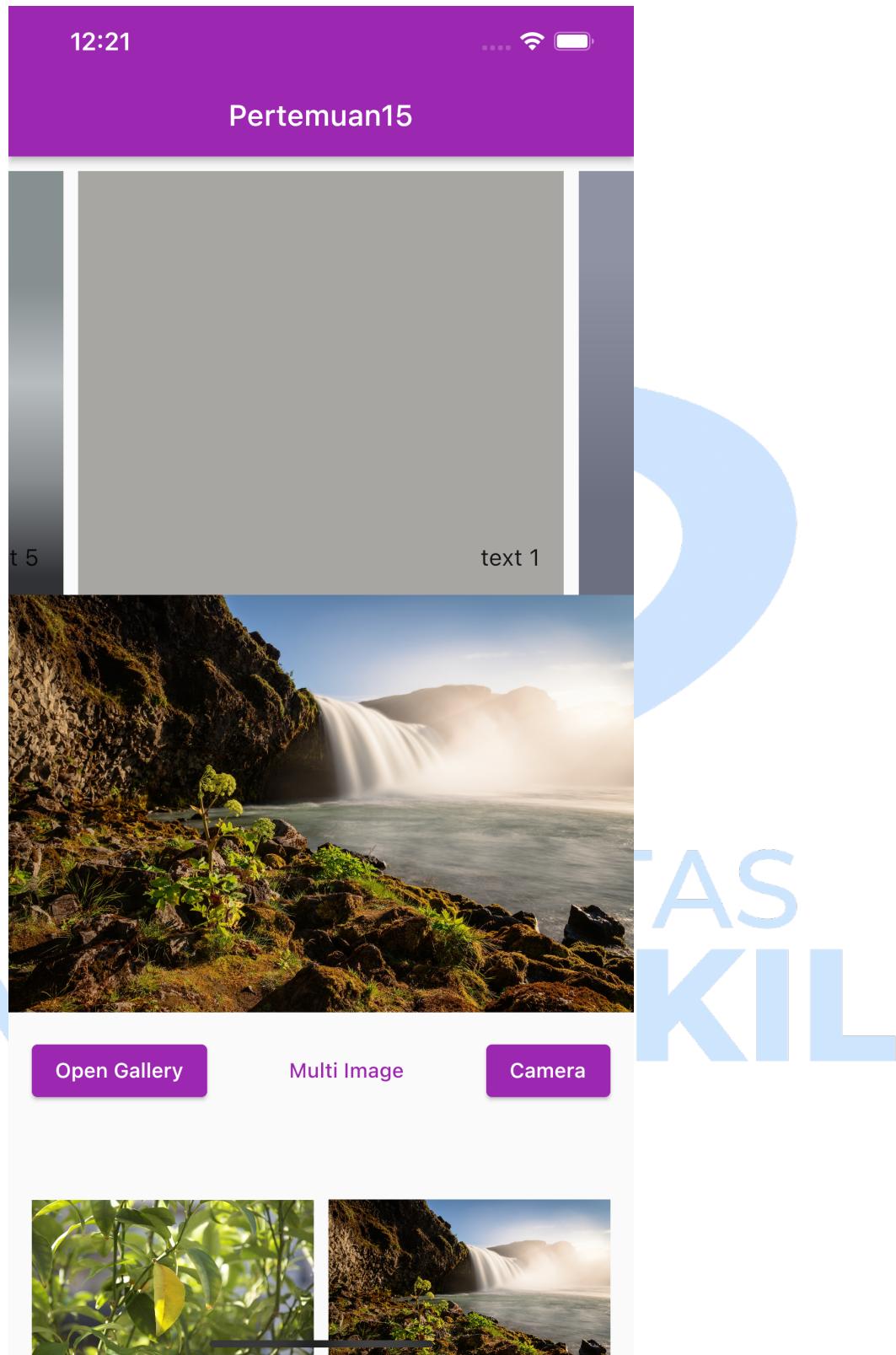
- Page ini akan digunakan menampilkan konten web dari url yang dikirim untuk tampil sebagai webview menggunakan plugin **WebView**
- Sesuai dokumentasi dari webview, pada initState() perlu cek platform yang digunakan.

Terapkan navigator pada gesture yang terdapat pada carousel_slider.dart pada tambahan sebelumnya, seperti gambar berikut:



Keterangan:

Cukup update bagian yang ditandai warnah merah. Reload untuk melihat hasilnya.



Latihan Praktek

1. Pada carousel slider, tampilkan gambar sesuai gambar yang terpilih dari multimage().
2. Cari package / plugin pada pub.dev untuk memilih file dan menampilkan video.



UNIVERSITAS
MIKROSKIL