

# Komponen dalam Mendesain Aplikasi

TI2137 - Pengembangan Aplikasi Mobil Front-End



# Team Teaching

- Sunaryo Winardi, S.Kom., M.TI.
- Sio Jurnal Pipin, S.Kom., M.Kom.
- M. Taufiq Hidayat Pohan, S.Kom.
- Richy Rotuahta Saragih, S.Kom.



# Komponen dalam Mendesain Aplikasi Part. 12

- Adding packager
- Image Picker
- Carousel Slider
- Webview Flutter



# Adding Packager



# Packages

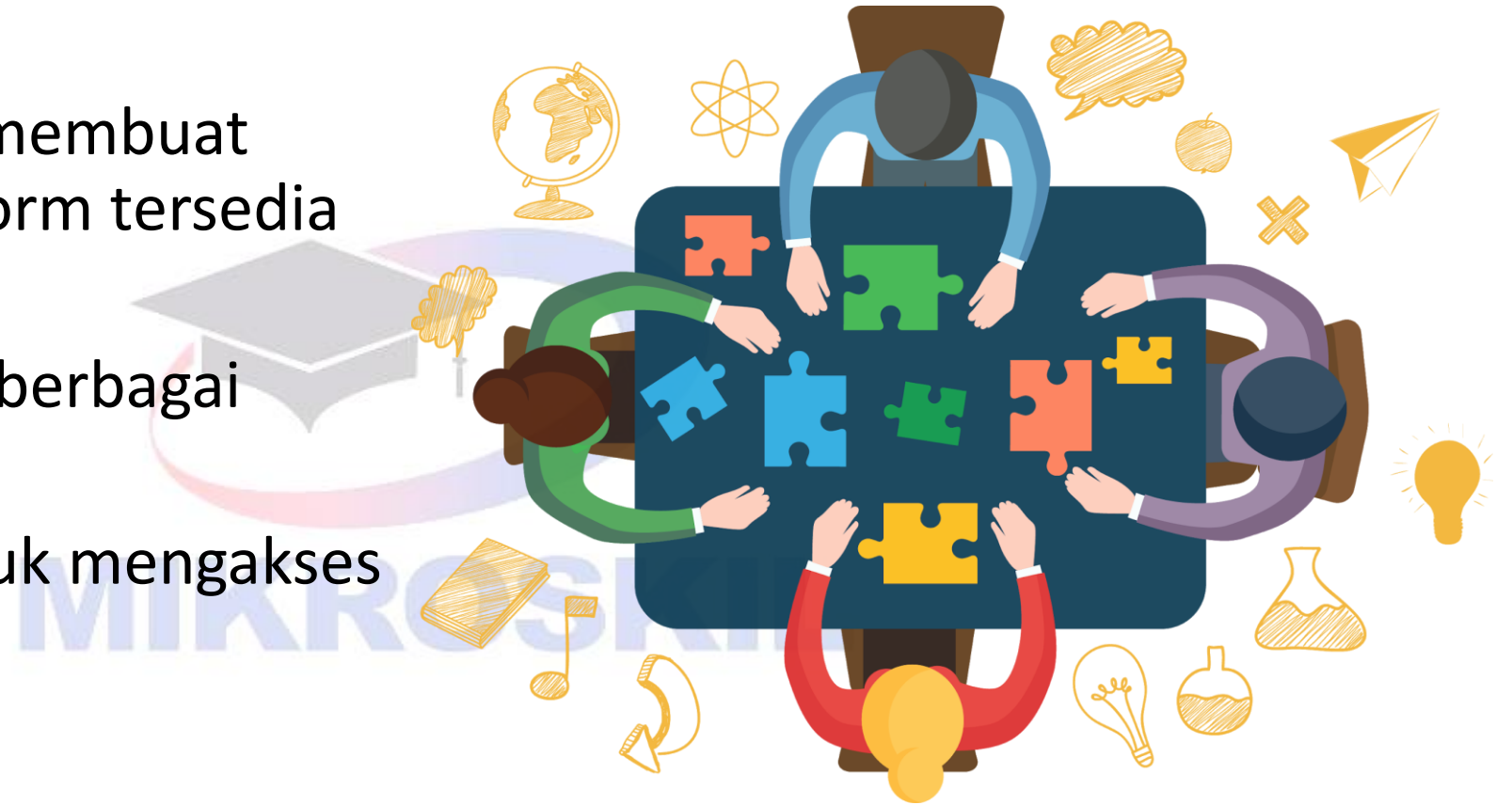
- Direktori yang memiliki sebuah pubspec.yaml
- Terdapat dependencies berupa pustaka Dart, aplikasi resource, contoh, pengujian dan lainnya.

MIKRO



# Plugins

- Paket khusus yang membuat fungsionalitas platform tersedia untuk aplikasi
- Dapat ditulis untuk berbagai platform
- Contoh : plugin untuk mengakses kamera

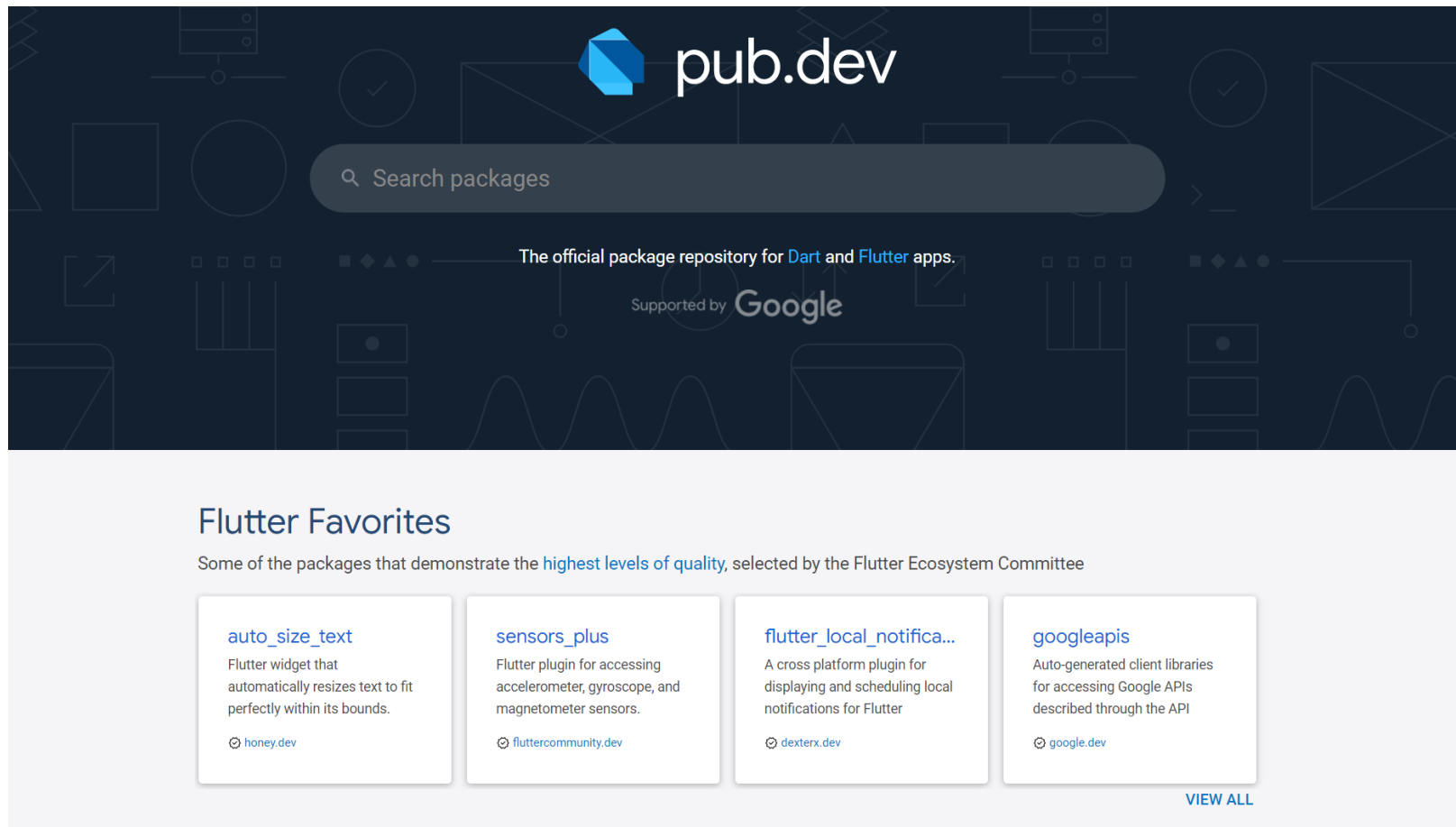


# Package Flutter

- Dapat diakses pada : [Pub.dev](https://pub.dev)
- Tersedia berbagai package untuk kasus tertentu, misalnya :
  - Http -> keperluan network request
  - Fluro -> Penanganan route pada aplikasi
  - FlutterFire -> menghubungkan flutter dengan Firebase

MIKROSKIL

# Pub.dev





# Adding a Package (#1)

- Buka file pubspec.yaml
- Tambahkan nama package dan versi di dalam list dependencies
  - Format :  
    [package\_name] : [version]
  - Contoh :  
    provider : ^6.0.2
- Buka terminal pada folder project
- Jalankan perintah **flutter pub get**

```
dependencies:  
  flutter:  
    sdk: flutter  
  
  cupertino_icons: ^1.0.2  
  provider: ^6.0.2  
  intl: ^0.17.0
```

# Adding a Package (#1)

- Tambah package pada file dengan **import**
  - Contoh
    - `import 'package:provider/provider.dart';`
- Jalankan kembali aplikasi dengan **Hot Restart**, untuk menghindari terjadinya error **MissingPluginExeption**.

MIKROSKIL

# Conflict Resolution

- Terjadinya konflik pada versi sebuah package.
- Misalnya package\_a dan package\_b membutuhkan image\_picker dengan versi yang berbeda.
- Diselesaikan dengan menggunakan rentang versi (^), artinya menggunakan versi tersebut atau lebih tinggi akan otomatis di upgrade
- Contoh  
**provider : ^6.0.2**

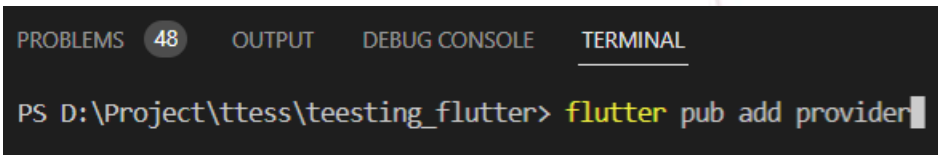
MIKROSKIL

# Adding a Package (#2)

- Buka terminal atau cmd pada direktori project
- Jalankan perintah flutter pub add [nama\_package]

**Contoh :**

flutter pub add provider

A screenshot of an IDE's terminal window. The terminal has tabs for 'PROBLEMS' (with a count of 48), 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'TERMINAL' tab is active. The command prompt shows the path 'PS D:\Project\ttess\teesting\_flutter>' followed by the command 'flutter pub add provider' which is partially typed, with a cursor at the end.

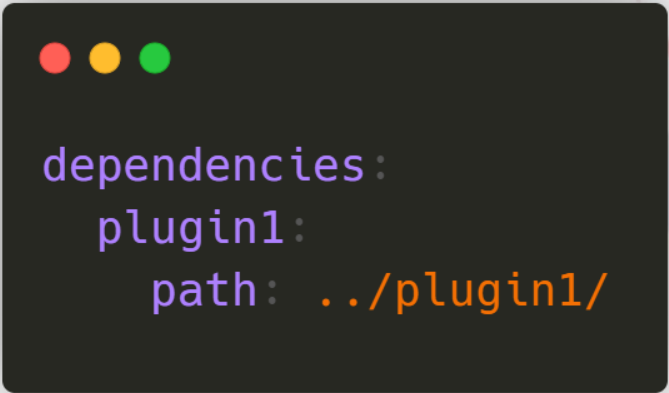
```
PROBLEMS 48 OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Project\ttess\teesting_flutter> flutter pub add provider
```

- Panggil package di file menggunakan import

# Menambah Unpublished Packages

## Path dependency

Memanggil package dari path tertentu



```
dependencies:  
  plugin1:  
    path: ../plugin1/
```



# Menambah Unpublished Packages

## Git dependency

Memanggil package dari repository git tertentu



```
dependencies:  
  plugin1:  
    git:  
      url: git://github.com/flutter/plugin1.git
```

# Menambah Unpublished Packages

## Git dependency on a Package in a folder

Memanggil package dari repository git tertentu, namun package bukan terletak di root folder, melainkan di folder lain pada repository tersebut.

```
dependencies:  
  package1:  
    git:  
      url: git://github.com/flutter/packages.git  
      path: packages/package1
```

# Memanggil Packages

- Pastikan packages sudah terdaftar di list dependencies pada pubspec.yaml
- Buka file dart yang ingin memanggil package
- Tambahkan import lalu direktori package pada bagian atas file (biasanya sebelum membuat sebuah class)

home\_screen.dart X

lib > pertemuan13 > home\_screen.dart > HomeScreen

```
1  import 'package:flutter/material.dart';  
2  import 'package:provider/provider.dart';  
3
```



# Memanggil Packages

- Setelah mengimport package, sekarang package dapat dipanggil dengan diawali nama classnya.
- Misalnya **Provider.of(context)**



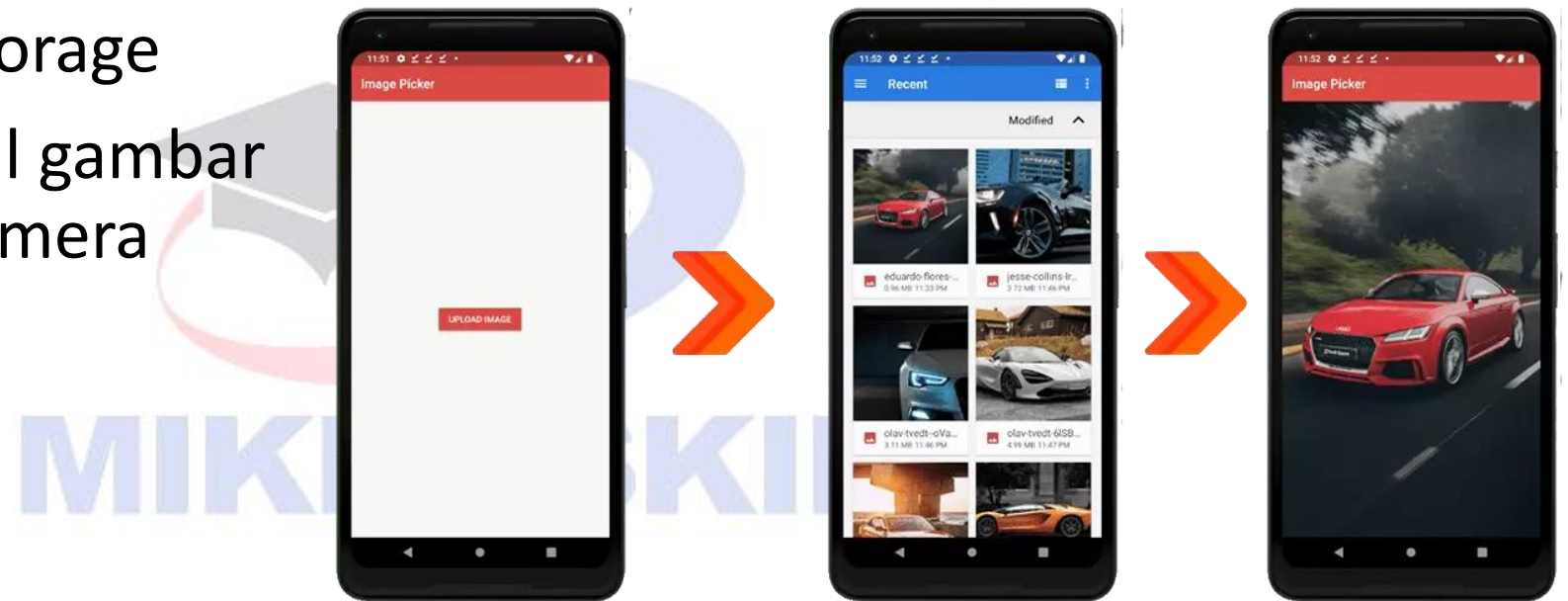
```
final prov = Provider.of<HomeProvider>(context);
```

# Image Picker



# ImagePicker adalah...

- Plugin untuk mengambil gambar dari local storage
- Bisa juga mengambil gambar langsung melalui kamera device



# Instalasi


Jalankan Perintah berikut ini di terminal / command prompt



```
flutter pub add image_picker
```

# Instalasi

Atau tambahkan baris berikut ini di file **pubspec.yaml** dan jalankan perintah **flutter get pub** di terminal / command prompt

A dark-themed terminal window with three colored window control buttons (red, yellow, green) in the top-left corner. It contains two lines of code in a light purple font. The first line is 'dependencies:' and the second line is 'image\_picker: ^0.8.5+3'.

```
dependencies:  
  image_picker: ^0.8.5+3
```

# Konfigurasi (iOS)

Jika aplikasi akan di pasang di sistem operasi iOS, tambahkan baris berikut ini di dalam file **<project root>/ios/Runner/Info.plist** agar aplikasi dapat akses mengambil gambar dan dapat lolos dari pemeriksaan Apple Developer. Package dapat berjalan di iOS versi **9.0 keatas**.

```
<dict>
.
.
<key>NSPhotoLibraryUsageDescription</key>
<string>Upload images for screen background</string>
<key>NSCameraUsageDescription</key>
<string>Upload image from camera for screen background</string>
<key>NSMicrophoneUsageDescription</key>
<string>Post videos to profile</string>
.
.
</dict>
```

# Konfigurasi (iOS)

## Penjelasan

1. **NSPhotoLibraryUsageDescription** - Why app needs gallery access?
2. **NSCameraUsageDescription** - Why app needs access to camera?
3. **NSMicrophoneUsageDescription** - Why app needs microphone access?

Meskipun sedang tidak membutuhkan akses kamera atau mic, konfigurasi tersebut tetap harus dipasang.

# Konfigurasi (Android)

- Untuk mulai versi **Android 4.3 (Jelly Bean) keatas**, Android sudah mengimplementasikan fitur pick multiple image.
- **Tidak perlu lakukan konfigurasi khusus.**
- Gambar yang diambil dari kamera akan disimpan di local cache, sehingga bersifat sementara.
- *Optional* : Direkomendasikan untuk membuat persiapan ketika suatu saat android akan mematikan app saat low memory. Untuk caranya akan dibahas nanti.



# Implementasi

```
/// Get from gallery
_getFromGallert() async {
  XFile? pickedFile = await ImagePicker().pickImage(
    source: ImageSource.gallery,
    maxWidth: 1800,
    maxHeight: 1800
  );
  if (pickedFile != null) {
    File imageFile = File(pickedFile.path);
  }
}
```

**Reminder : Jangan lupa import**

**import 'dart:io';**

**import 'package:image\_picker/image\_picker.dart';**

# Property di pickImage

Property	Class/DataType	Deskripsi
source	ImageSource (@required)	Set sumber gambar. Ada 2 properti, yaitu <b>ImageSource.gallery</b> dan <b>ImageSource.camera</b>
maxWidth	double?	Set lebar maksimum gambar (jika melebihi maksimum, maka akan otomatis di rescale)
maxHeight	double?	Set tinggi maksimum gambar (jika melebihi maksimum, maka akan otomatis di rescale)
imageQuality	int?	Set kualitas gambar yang di import (range 0-100)
preferredCameraDevice	CameraDevice	Set kamera yang akan dipakai jika source gambar adalah <b>ImageSource.camera</b> . Ada 2 properti, yaitu <b>CameraDevice.rear</b> (Kamera Belakang) dan <b>CameraDevice.front</b> (Kamera Depan)

# Implementasi

```
/// Get from camera
_getFromGallert() async {
  XFile? pickedFile = await ImagePicker().pickImage(
    source: ImageSource.camera,
    maxWidth: 1800,
    maxHeight: 1800,
    preferredCameraDevice: CameraDevice.front
  );
  if (pickedFile != null) {
    File imageFile = File(pickedFile.path);
  }
}
```

# Implementasi



```
// Pick a video
final XFile? image = await _picker.pickVideo(source: ImageSource.gallery);
// Capture a video
final XFile? video = await _picker.pickVideo(source: ImageSource.camera);
```

# Implementasi

```

List<XFile>? _imageFileList;

// Multiple images
_getFromGallery() async {
  List<XFile>? pickedFiles =await ImagePicker().pickMultiImage(
    maxWidth: 1800,
    maxHeight: 1800);
  if (pickedFile != null) {
    setState(() {
      _imagefileList = pickedFiles;
    });
  }
}
```

# Recover Lost Data

- Android dapat mematikan app yang menggunakan image\_picker.
- Di Android, image\_picker menggunakan intent default `Intent.ACTION_GET_CONTENT` atau `MediaStore.ACTION_IMAGE_CAPTURE`.
- Ketika screen image picker muncul, app utama akan pindah ke background dan dapat otomatis dimatikan oleh sistem android ketika low memory.

# Recover Lost Data

- Ketika image picker selesai dilakukan, android akan merestart app utama.
- Karena data tidak akan dikembalikan ke fungsi asli pemanggilnya, gunakan **ImagePicker.retriveLostData()** untuk mengembalikannya.



MIKROSKIL

# Recover Lost Data

```
Future<void> getLostData() async {  
    final LostDataResponse response =  
        await picker.retrieveLostData();  
    if (response.isEmpty) {  
        return;  
    }  
    if (response.files != null) {  
        for (final XFile file in response.files) {  
            _handleFile(file);  
        }  
    } else {  
        _handleError(response.exception);  
    }  
}
```

Disarankan dijalankan ketika app baru dibuka untuk handle kejadian seperti ini



# Carousel Slider



# Carousel adalah...

- Carousel adalah fitur yang memungkinkan pengguna mengunggah lebih dari satu foto atau video dalam sekali postingan.
- Carousel bersifat interaktif
- Dapat melihat deretan gambar atau video di dalamnya sesuai urutan dengan menge-scroll ke kiri atau kanan



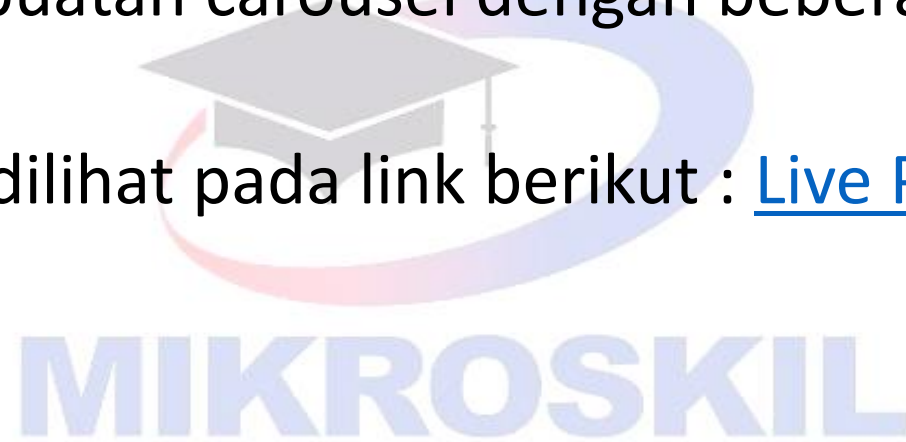
MIKROSKIL

# Contoh Carousel



# Carousel Slider

- Sebuah package flutter dari [pub.dev](https://pub.dev)
- Memudahkan pembuatan carousel dengan beberapa fitur dan multiplatform
- Live preview dapat dilihat pada link berikut : [Live Preview](#)



# Features

- Infinite Scroll
- Custom child widgets
- Auto play



# Supported Platforms

- Flutter Android
- Flutter iOS
- Flutter web
- Flutter desktop



# Installation (#1)

- Buka file pubspec.yaml
- Tambahkan **carousel\_slider: ^4.1.1** pada list dependencies
- Save file dan jalankan **flutter pub get** pada terminal
- Import package pada file dart



```
import 'package:carousel_slider/carousel_slider.dart';
```

# Installation (#2)

- Buka terminal pada folder project
- Jalankan perintah **flutter pub add carousel\_slider**
- Import package pada file dart



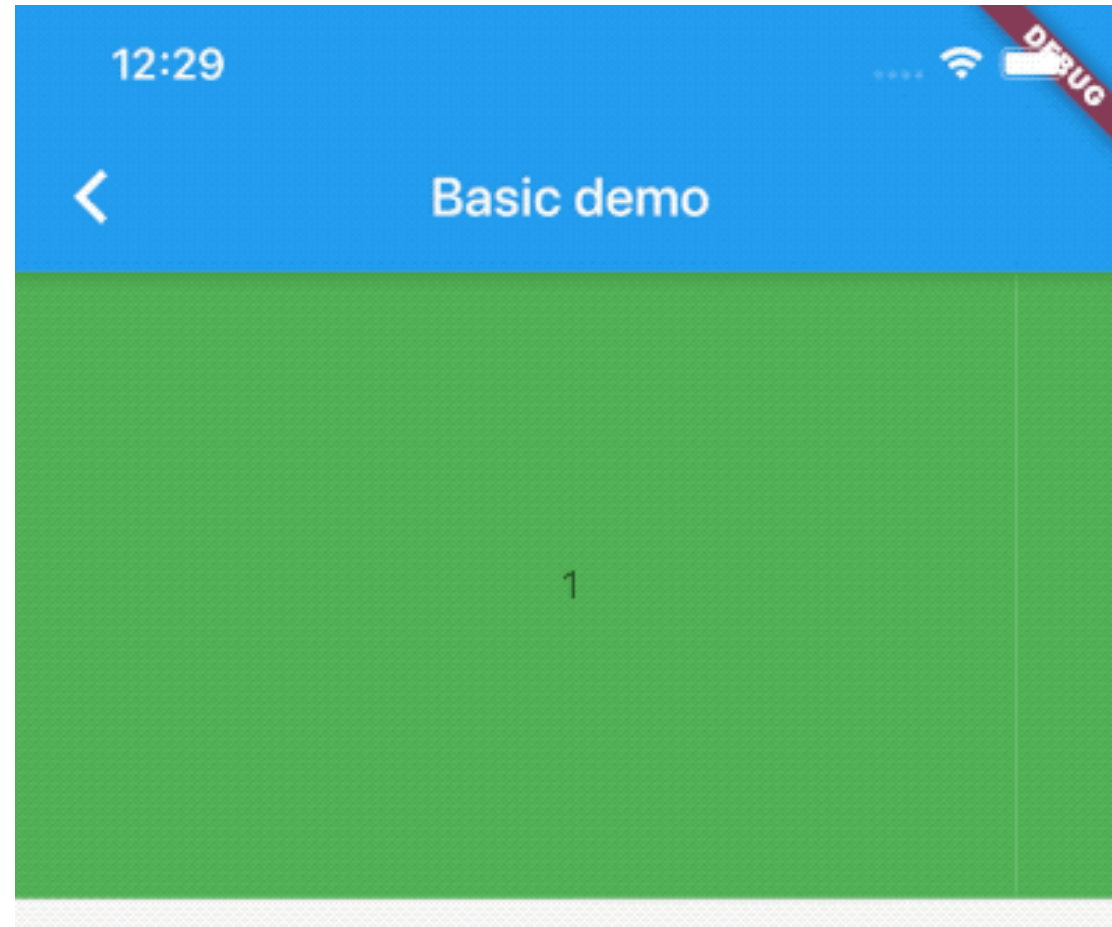
```
import 'package:carousel_slider/carousel_slider.dart';
```



# Simple Widget

```
CarouselSlider(  
  options: CarouselOptions(height: 400.0),  
  items: [1,2,3,4,5].map((i) {  
    return Builder(  
      builder: (BuildContext context) {  
        return Container(  
          width: MediaQuery.of(context).size.width,  
          margin: EdgeInsets.symmetric(horizontal: 5.0),  
          decoration: BoxDecoration(  
            color: Colors.amber  
          ),  
          child: Text('text $i', style: TextStyle(fontSize: 16.0),)  
        );  
      },  
    );  
  }).toList(),  
)
```

# Simple Widget



# Options pada CarouselSlider

Property	Class/DataType	Deskripsi
aspectRatio	double	Digunakan jika height tidak di deklarasikan
autoPlay	bool	Melakukan slide secara otomatis
autoPlayAnimationDuration	Duration	Menentukan durasi slide otomatis
height	double?	Menentukan tinggi widget
initialPage	Int	Menentukan konten yang pertama ditampilkan
reverse	Bool	Membalik urutan dari konten
scrollDirection	Axis	Menentukan axis untuk scroll atau slide konten
enableInfiniteScroll	Bool	Jika true, slide atau scroll dilakukan secara berulang
enlargeCenterPage	Bool	Konten akan lebih besar daripada konten sebelahnya

Lebih lengkap dapat dibuka pada link berikut : [CarouselSlider Options](#)

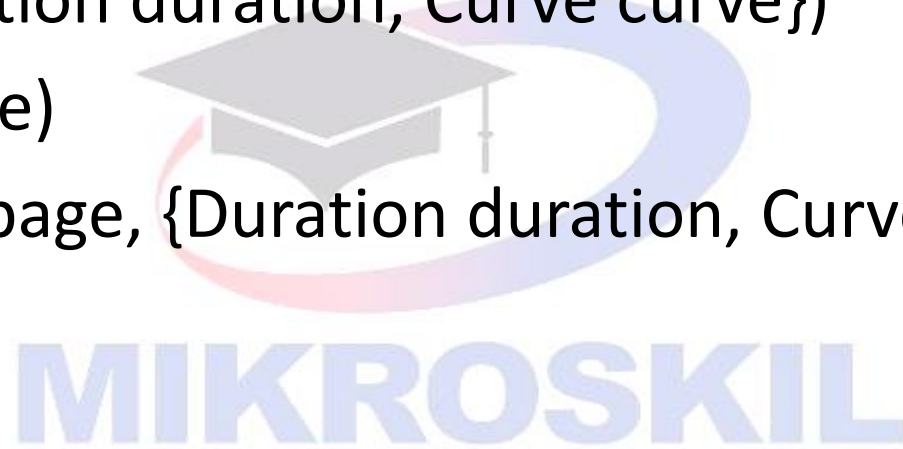
# Carousel Controller

- Berfungsi untuk mengontrol posisi konten yang ditampilkan.
- Terdapat method yang mengatur posisi dan aksi pada carousel



# Method pada CarouselController

- nextPage({Duration duration, Curve curve})
- previousPage({Duration duration, Curve curve})
- jumpToPage(int page)
- animateToPage(int page, {Duration duration, Curve curve})



# Method pada CarouselController

## .nextPage

- Parameters :
  - **Duration** duration
  - **Curve** curve
- Fungsi
  - Berfungsi untuk menampilkan **konten selanjutnya** jika masih ada.
  - Jika konten habis dan **infiniteLoop aktif**, maka akan menampilkan **konten awal**.



# Method pada CarouselController

## . previousPage

- Parameters :

- **Duration** duration
- **Curve** curve

- Fungsi

- Berfungsi untuk menampilkan konten **sebelumnya** jika masih ada.
- Jika konten habis dan **infiniteLoop aktif**, maka akan menampilkan **konten akhir**.



# Method pada CarouselController

## . jumpToPage

- Parameters :

- **int** page

- Fungsi

- Lompat ke konten pada index sesuai parameter page
  - Jika **page > length**, dapat menimbulkan **exception**





# Method pada CarouselController

## .nextPage

- Parameters :
  - **int** page
  - **Duration** duration
  - **Curve** curve
- Fungsi
  - Berfungsi untuk menampilkan **konten sesuai index** jika ada dengan **animasi tertentu**.



# Implementasi



# Implementasi

- Buat list string untuk menampung url image

```
final List<String> imgList = [  
    'https://bit.ly/3u4jXuk',  
    'https://bit.ly/3A39Fi5',  
    'https://bit.ly/3ymuELj',  
    'https://bit.ly/39QLicN',  
    'https://bit.ly/30r1ZdF',  
    'https://bit.ly/3A2Zyty',  
];
```

# Implementasi

- Buat list widget untuk memanggil semua gambar melalui link

```
final List<Widget> imageSliders = imgList
.map((item) => Container(
  child: Container(
    margin: const EdgeInsets.all(5.0),
    child: ClipRRect(
      borderRadius: const BorderRadius.all(Radius.circular(5.0)),
      child: Stack(
        children: <Widget>[
          Image.network(item, fit: BoxFit.cover, width: 1000.0),
          Positioned(
            bottom: 0.0,
            left: 0.0,
            right: 0.0,
            child: Container(
              decoration: const BoxDecoration(
                gradient: LinearGradient(
                  colors: [
                    Color.fromARGB(200, 0, 0, 0),
                    Color.fromARGB(0, 0, 0, 0)
                  ],
                  begin: Alignment.bottomCenter,
                  end: Alignment.topCenter,
                ),
              ),
            ),
            padding: const EdgeInsets.symmetric(
              vertical: 10.0, horizontal: 20.0),
            child: Text(
              'No. ${imgList.indexOf(item)} image',
              style: const TextStyle(
                color: Colors.white,
                fontSize: 20.0,
                fontWeight: FontWeight.bold,
              ),
            ),
          ),
        ],
      ),
    ),
  ),
).toList();
```

# Implementasi

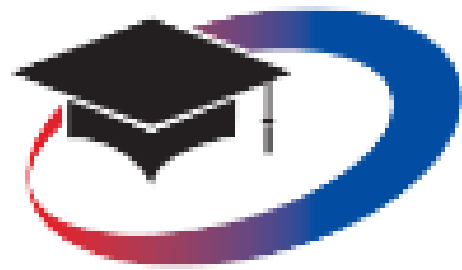
- Selanjutnya, buat column untuk memanggil widget list dan indicator dari slidernya.

```
Column(children: [
  Expanded(
    child: CarouselSlider(
      items: imageSliders,
      carouselController: _controller,
      options: CarouselOptions(
        autoPlay: true,
        enlargeCenterPage: true,
        aspectRatio: 2.0,
        onPageChanged: (index, reason) {
          setState(() {
            _current = index;
          });
        },
      ),
    ),
  ),
  Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: imgList.asMap().entries.map((entry) {
      return GestureDetector(
        onTap: () => _controller.animateToPage(entry.key),
        child: Container(
          width: 12.0,
          height: 12.0,
          margin:
            const EdgeInsets.symmetric(vertical: 8.0, horizontal: 4.0),
          decoration: BoxDecoration(
            shape: BoxShape.circle,
            color: (Theme.of(context).brightness == Brightness.dark
              ? Colors.white
              : Colors.black)
              .withOpacity(_current == entry.key ? 0.9 : 0.4)),
          ),
        ),
      }).toList(),
    ),
  ),
]),
```

# Implementasi

MIL

# Webview Flutter



UNIVERSITAS  
**MIKROSKIL**

PRODI. TEKNIK INFORMATIKA (S-1)

# Webview adalah...

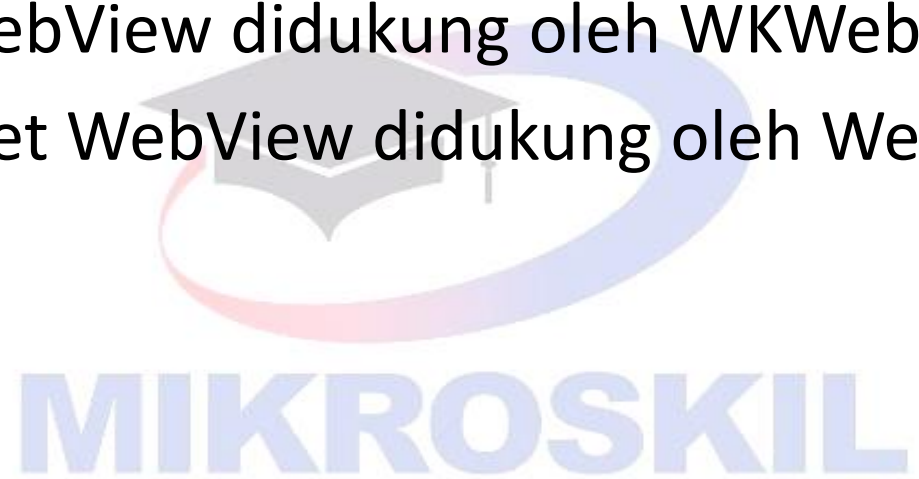
Komponen yang memungkinkan aplikasi dapat menampilkan konten website pada flutter





# WebView Flutter

- Plugin Flutter yang menyediakan widget WebView
- Pada iOS, widget WebView didukung oleh WKWebView
- Pada Android, widget WebView didukung oleh WebView



# Spesifikasi Minimum

	Android	iOS
Mendukung	SDK 19+ atau 20+	9.0+

# Installation (#1)

- Buka file pubspec.yaml
- Tambahkan **webview\_flutter: ^3.0.4** pada list dependencies
- Save file dan jalankan **flutter pub get** pada terminal
- Import package pada file dart



```
import 'package:webview_flutter/webview_flutter.dart';
```

# Installation (#2)

- Buka terminal pada folder project
- Jalankan perintah **flutter pub add webview\_flutter**
- Import package pada file dart



```
import 'package:webview_flutter/webview_flutter.dart';
```

# Tampilan Platform

## Hybrid Composition

- Memiliki dukungan keyboard default
- Membutuhkan Android SDK 19+



# Tampilan Platform


## Virtual Display

- Keyboard default sering bermasalah
- Membutuhkan Android SDK 20+



# Initial Setup

1. Update **minSdkVersion** pada build.gradle di folder **android/app/** menjadi **19**

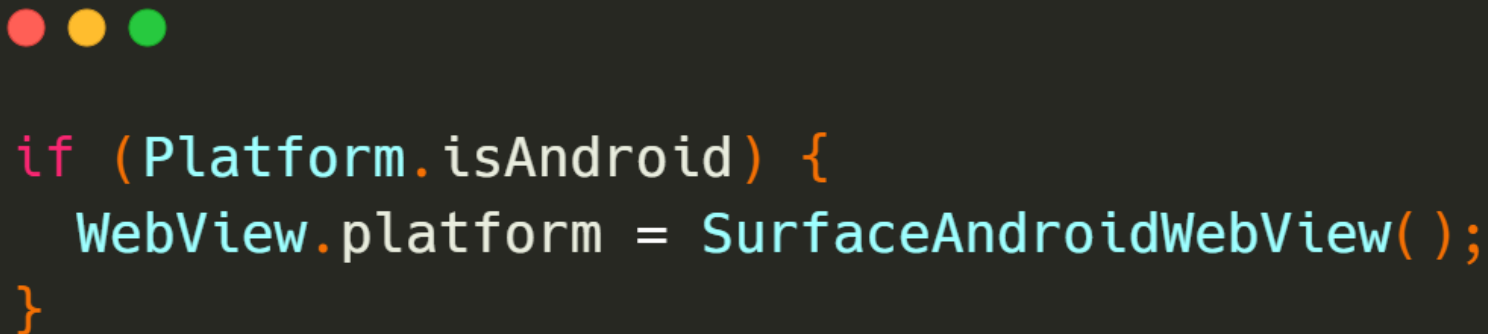


```
android {  
    defaultConfig {  
        minSdkVersion 19  
    }  
}
```

# Initial Setup

2. Ketik kode berikut pada **initState**

Kode ini untuk mengecek jika platform **android**, akan menggunakan class **SurfaceAndroidWebView**



```
if (Platform.isAndroid) {  
    WebView.platform = SurfaceAndroidWebView();  
}
```



# Implementasi

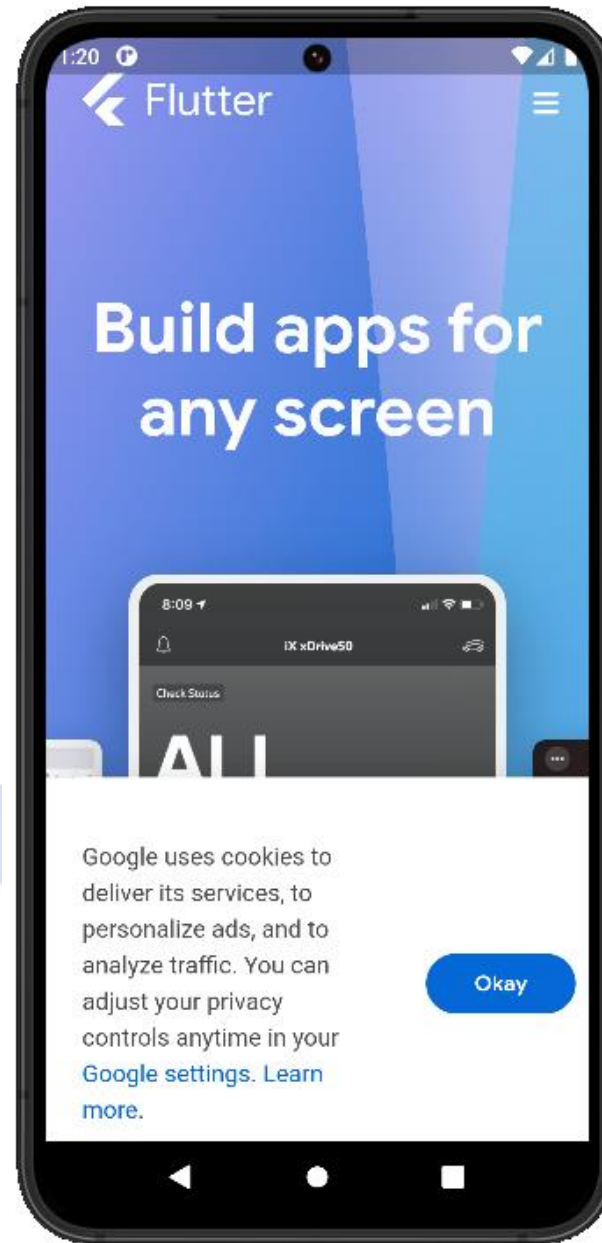


# Demo WebView

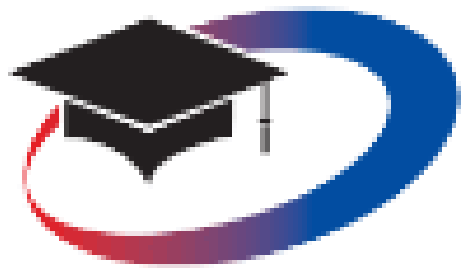


```
Container(  
  child: const WebView(  
    initialUrl: 'https://flutter.dev',  
    javascriptMode: JavascriptMode.unrestricted,  
  ),  
)
```

# Demo WebView



# Terima Kasih



UNIVERSITAS  
**MIKROSKIL**

PRODI. TEKNIK INFORMATIKA (S-1)