

**TUGAS GUIDED
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL IX
API Perangkat Keras**



**Disusun Oleh :
Althafia Defiyandrea Laskanadya Wibowo
2211104011 / SE06-01**

**Asisten Praktikum :
Muhammad Faza Zulian Gesit Al Barru
Aisyah Hasna Aulia**

**Dosen Pengampu :
Yudha Islami Sulistya, S.Kom., M.Cs.**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2024

1. Camera API

Camera API berfungsi untuk memungkinkan developer (pengembang) untuk mengakses dan mengontrol kamera perangkat. Flutter menyediakan paket camera yang memudahkan implementasi fitur kamera untuk mengambil foto, merekam video, dan mengakses umpan kamera secara langsung. Paket ini sangat berguna untuk membuat aplikasi yang membutuhkan pengambilan gambar atau video, seperti aplikasi media sosial atau e-commerce.

2. Media API

Media API adalah sekumpulan alat dan pustaka yang mendukung pengelolaan dan interaksi dengan berbagai jenis media, seperti gambar, video, dan audio. Flutter tidak memiliki API media bawaan untuk semua kebutuhan media, tetapi dapat menggunakan paket-paket tambahan untuk mengakses fitur media yang umum di aplikasi.

GUIDED

Main.dart

```
1 import 'package:flutter/material.dart';
2 import 'myapi_page.dart';
3
4 void main() {
5   runApp(const MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   const MyApp({super.key});
10
11   @override
12   Widget build(BuildContext context) {
13     return MaterialApp(
14       title: 'Flutter Demo',
15       theme: ThemeData(
16         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
17         useMaterial3: true,
18       ),
19       home: MyApiPage(),
20     );
21   }
22 }
```

Image_picker_screen.dart

```
1  import 'dart:io';
2  import 'package:flutter/material.dart';
3  import 'image_picker.dart';
4
5  class ImageFromGalleryEx extends StatefulWidget {
6    final ImageSourceType type;
7    ImageFromGalleryEx(this.type);
8
9    @override
10   ImageFromGalleryExState createState() => ImageFromGalleryExState(this.type);
11 }
12
13 class ImageFromGalleryExState extends State<ImageFromGalleryEx> {
14   File? _image;
15   late ImagePicker imagePicker;
16   final ImageSourceType type;
17
18   ImageFromGalleryExState(this.type);
19
20   @override
21   void initState() {
22     super.initState();
23     imagePicker = ImagePicker();
24   }
25
26   @override
27   Widget build(BuildContext context) {
28     return Scaffold(
29       appBar: AppBar(
30         title: Text(type == ImageSourceType.camera
31           ? "Image from Camera"
32           : "Image from Gallery"),
33       ),
34       body: Column(
35         children: <Widget>[
36           SizedBox(height: 52),
37           Center(
38             //mengambil gambar dari camera atau gallery
39             child: GestureDetector(
40               onTap: () async {
41                 //operasi ternary untuk memilih sumber gambar
42                 var source = type == ImageSourceType.camera
43                   ? ImageSource.camera
44                   : ImageSource.gallery;
45
46                 //menyimpan gambar pada variabel image
47                 XFile? image = await imagePicker.pickImage(
48                   source: source,
49                   imageQuality: 50,
50                   preferredCameraDevice: CameraDevice.front);
```

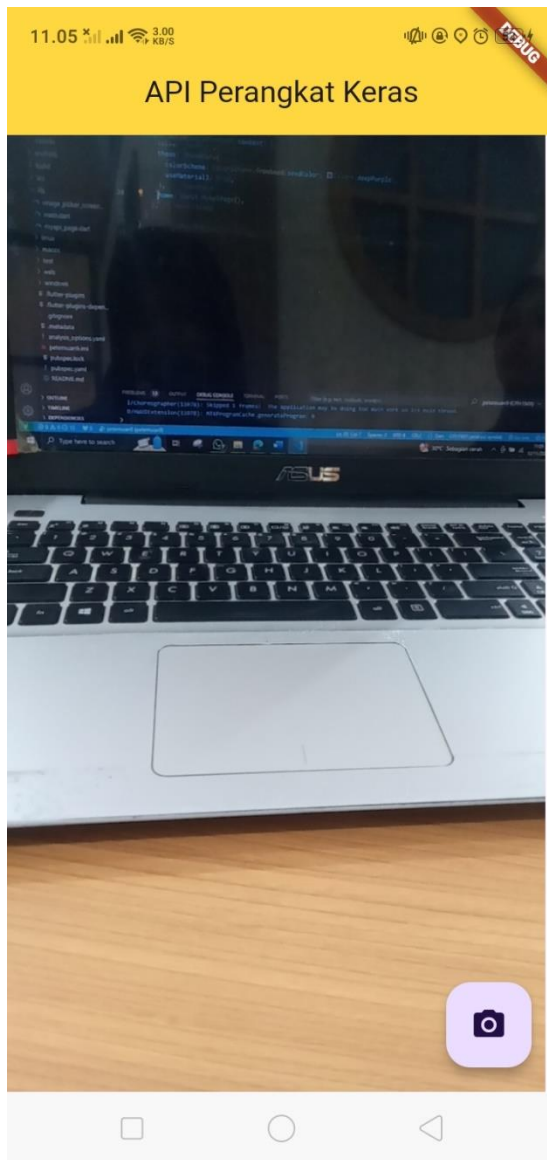
```

51
52         if (image != null) {
53             setState(() {
54                 _image = File(image.path);
55             });
56         }
57     },
58     child: Container(
59         width: 200,
60         height: 200,
61         decoration: BoxDecoration(
62             color: Colors.red[200],
63         ),
64
65         // menampilkan gambar dari camera atau gallery
66         child: _image != null
67             ? Image.file(
68                 _image!,
69                 width: 200.0,
70                 height: 200.0,
71                 fit: BoxFit.fitHeight,
72             )
73
74             // jika tidak ada gambar yang dipilih
75             : Container(
76                 decoration: BoxDecoration(
77                     color: Colors.red[200],
78                 ),
79                 width: 200,
80                 height: 200,
81                 child: Icon(
82                     Icons.camera_alt,
83                     color: Colors.grey[800],
84                 ),
85             ),
86     ),
87 ),
88 ],
89 ),
90 );
91 );
92 }
93 }
94
95 enum ImageSourceType { camera, gallery }
96

```

Myapi_page

```
1 import 'package:camera/camera.dart';
2 import 'package:flutter/material.dart';
3
4 class MyApiPage extends StatefulWidget {
5   @override
6   _MyApiPageState createState() => _MyApiPageState();
7 }
8
9 class _MyApiPageState extends State<MyApiPage> {
10   late CameraController _controller;
11   Future<void>? _initializeControllerFuture;
12
13   @override
14   void initState() {
15     super.initState();
16     _initializeControllerFuture = _initializeCamera();
17   }
18
19   Future<void> _initializeCamera() async {
20     final cameras = await availableCameras();
21     final firstCamera = cameras.first;
22
23     _controller = CameraController(
24       firstCamera,
25       ResolutionPreset.medium,
26     );
27
28     await _controller.initialize();
29   }
30
31   @override
32   void dispose() {
33     _controller.dispose();
34     super.dispose();
35   }
36
37   @override
38   Widget build(BuildContext context) {
39     return Scaffold(
40       appBar: AppBar(
41         title: Text("Api Perangkat Keras"),
42         centerTitle: true,
43         backgroundColor: Color(0xFFFFD700),
44       ),
45       body: FutureBuilder<void>(
46         future: _initializeControllerFuture,
47         builder: (context, snapshot) {
48           if (snapshot.connectionState == ConnectionState.done) {
49             return CameraPreview(_controller);
50           } else {
51             return Center(child: CircularProgressIndicator());
52           }
53         },
54       ),
55
56       // floatingActionButton: FloatingActionButton(
57       //   onPressed: () {
58       //     Navigator.pop(context);
59       //   },
60       //   child: Icon(Icons.arrow_back),
61       // ),
62     );
63   }
64 }
```



Hasil run dari program ini akan menghasilkan sebuah aplikasi Flutter dengan dua fitur utama: akses ke kamera perangkat dan galeri untuk memilih gambar. Ketika aplikasi dijalankan, pengguna akan disambut oleh antarmuka yang menampilkan tombol untuk mengambil gambar melalui kamera atau memilih gambar dari galeri. Saat tombol kamera ditekan, aplikasi menggunakan plugin `camera` untuk membuka kamera perangkat, memungkinkan pengguna mengambil foto, dan menampilkan foto tersebut di dalam sebuah container. Aplikasi ini mengimplementasikan logika asynchronous untuk memastikan pengalaman pengguna tetap lancar meskipun ada proses yang memerlukan waktu, seperti inisialisasi kamera atau pengambilan gambar.