

**LAPORAN PRAKTIKUM  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL IX  
API PERANGKAT KERAS**



**Disusun Oleh :**  
**Devrin Anggun Saputri / 2211104001**  
**SE0601**

**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**

# GUIDED

## 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.

### Cara Instalasi:

- Tambahkan paket camera yang ada pada Pub Dev di pubspec.yaml

```
dependencies:  
  flutter:  
    sdk: flutter  
  
  # The following adds the Cupertino Icons package, which includes many icons from the  
  # platform's Iкони. For more information, see: https://github.com/flutter/flutter/tree/master/packages/cupertino_icons  
  # Use with the CupertinoIcons class for iOS style icons.  
  cupertino_icons: ^1.0.8  
  camera: ^0.11.0+2  
  image_picker: ^1.1.2
```

- Lalu jalankan perintah ‘flutter pub get’
- Izinkan akses kamera pada AndroidManifest.xml

```
android > app > src > main > AndroidManifest.xml  
1   <manifest xmlns:android="http://schemas.android.com/apk/res/android">  
2     <uses-permission android:name="android.permission.CAMERA" />  
3     <uses-feature android:name="android.hardware.camera" />
```

- Ubah minimum versi Android sdk ke 21 pada file **android/app/build.gradle**

```
minSdkVersion 21
```

- Implementasi kamera pada halaman flutter di file myapi\_page.dart

```
● ● ●  
1 import 'dart:io';  
2 import 'package:flutter/material.dart';  
3 import 'package:camera/camera.dart';  
4  
5 class MyApiPage extends StatefulWidget {  
6   const MyApiPage({super.key});  
7  
8   @override  
9   State<MyApiPage> createState() => _MyApiPageState();  
10 }  
11  
12 class _MyApiPageState extends State<MyApiPage> {  
13   late CameraController _controller;  
14   late Future<void> _initializeControllerFuture;  
15 }
```

```
15
16     @override
17     void initState() {
18         super.initState();
19         _initializeCamera();
20     }
21
22     Future<void> _initializeCamera() async {
23         // Ambil daftar kamera yang tersedia di perangkat
24         final cameras = await availableCameras();
25         final firstCamera = cameras.first;
26
27         // Buat kontroler kamera dan mulai kamera
28         _controller = CameraController(
29             firstCamera,
30             ResolutionPreset.high,
31         );
32
33         _initializeControllerFuture = _controller.initialize();
34         setState(() {});
35     }
36
37     @override
38     void dispose() {
39         // Bersihkan kontroler ketika widget dihapus
40         _controller.dispose();
41         super.dispose();
42     }
43
44     @override
45     Widget build(BuildContext context) {
46         return Scaffold(
47             appBar: AppBar(
48                 title: Text("API Perangkat Keras"),
49                 centerTitle: true,
50                 backgroundColor: Colors.amberAccent,
51             ),

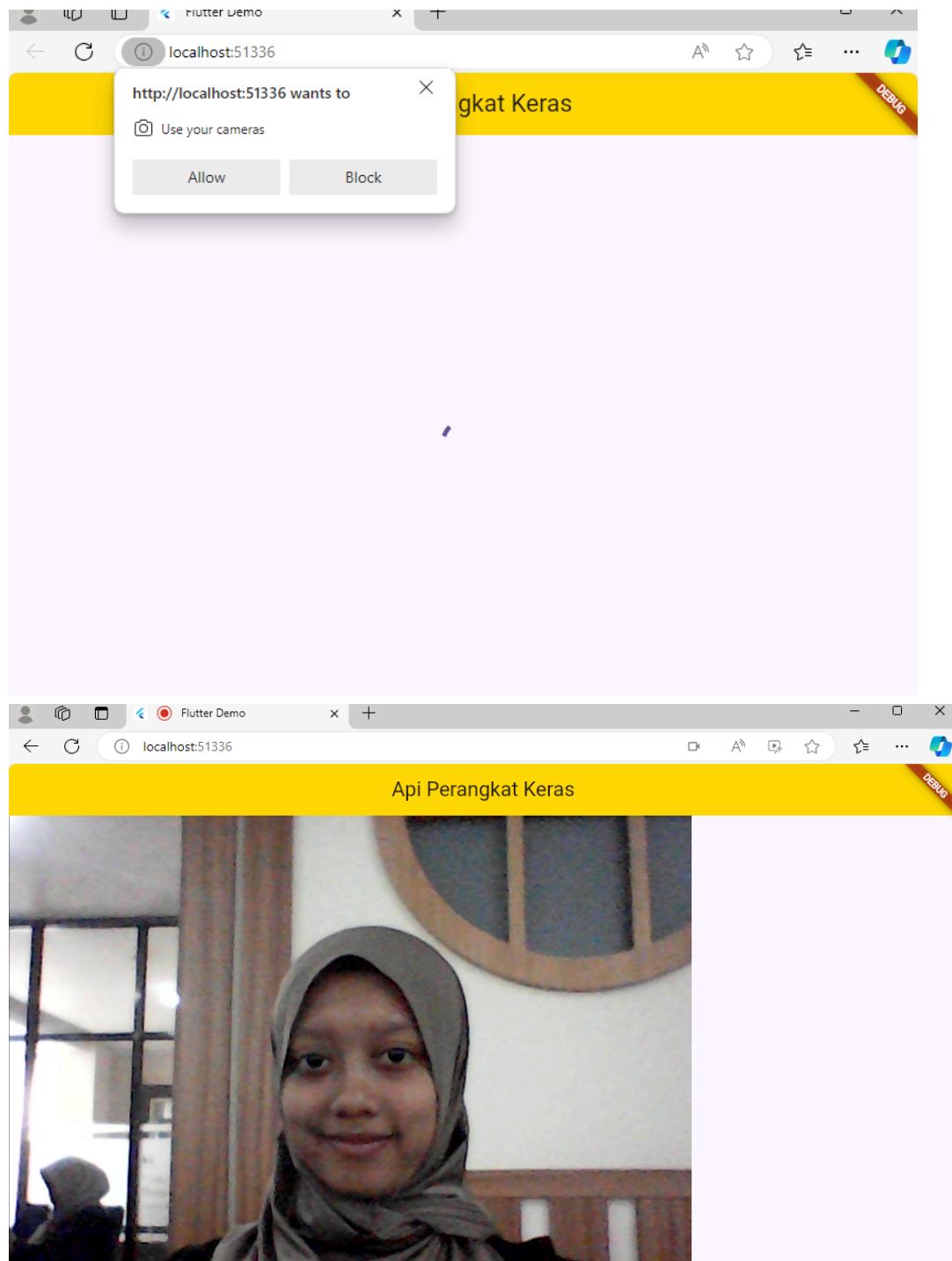
```



```
1     body: FutureBuilder<void>(
2         future: _initializeControllerFuture,
3         builder: (context, snapshot) {
4             if (snapshot.connectionState == ConnectionState.done) {
5                 // Jika kamera sudah siap, tampilkan pratinjau
6                 return CameraPreview(_controller);
7             } else {
8                 // Jika kamera belum siap, tampilkan loading
9                 return Center(child: CircularProgressIndicator());
10            }
11        },
12    ),
13    floatingActionButton: FloatingActionButton(
14        onPressed: () async {
15            try {
```

```
16          // Pastikan kamera sudah diinisialisasi
17          await _initializeControllerFuture;
18          // Ambil gambar
19          final image = await _controller.takePicture();
20          // Tampilkan atau gunakan gambar
21          Navigator.push(
22              context,
23              MaterialPageRoute(
24                  builder: (context) =>
25                      DisplayPictureScreen(imagePath: image.path),
26                  ),
27              );
28          } catch (e) {
29              print(e);
30          }
31      },
32      child: Icon(Icons.camera_alt),
33      ),
34  );
35 }
36 }
37
38 class DisplayPictureScreen extends StatelessWidget {
39     final String imagePath;
40
41     const DisplayPictureScreen({Key? key, required this.imagePath})
42         : super(key: key);
43
44     @override
45     Widget build(BuildContext context) {
46         return Scaffold(
47             appBar: AppBar(title: Text('Display Picture')),
48             body: Image.file(File(imagePath)),
49         );
50     }
51 }
```

## Output:



## 2. Media API

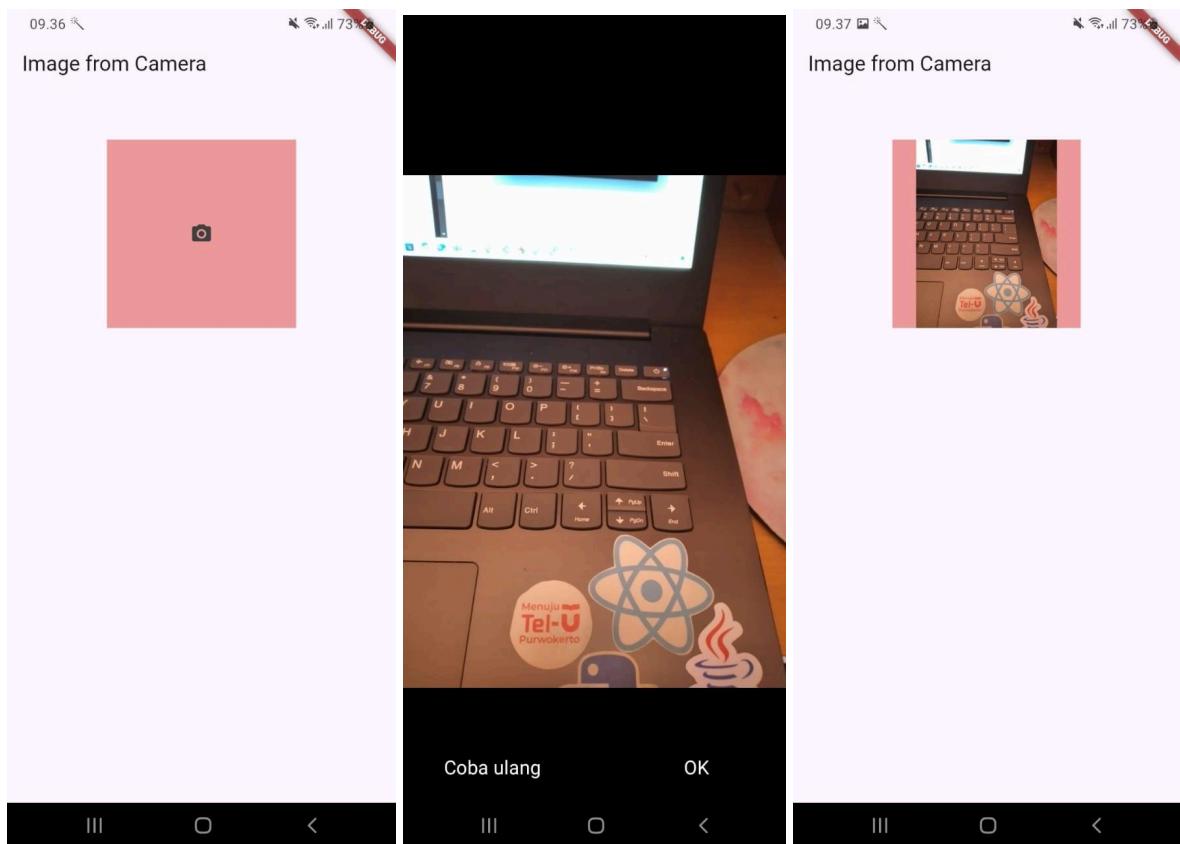
Media API adalah sekumpulan alat dan pustaka yang mendukung pengelolaan dan interaksi dengan berbagai jenis media, seperti gambar, video, dan audio. Flutter tidak memiliki API media bawaan untuk semua kebutuhan media, tetapi dapat menggunakan paket-paket tambahan untuk mengakses fitur media yang umum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

### a. File `image_picker_screen.dart`

```
1 import 'dart:io';
2 import 'package:flutter/material.dart';
3 import 'package:image_picker/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           )
53         )
54       ],
55     );
56   }
57 }
```

```
1          if (_image != null) {
2              setState(() {
3                  _image = File(image.path);
4              });
5          }
6      },
7      child: Container(
8          width: 200,
9          height: 200,
10         decoration: BoxDecoration(
11             color: Colors.red[200],
12         ),
13
14         // menampilkan gambar dari camera atau gallery
15         child: _image != null
16             ? Image.file(
17                 _image!,
18                 width: 200.0,
19                 height: 200.0,
20                 fit: BoxFit.fitHeight,
21             )
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44 enum ImageSourceType { camera, gallery }
```

## Output:



Halaman tersebut menampilkan fungsionalitas dasar dari sebuah kamera pada aplikasi mobile. Widget Camera digunakan untuk mengakses kamera perangkat dan menampilkan preview gambar secara real-time. Tombol IconButton dengan ikon kamera bertindak sebagai trigger untuk mengambil foto. Setelah foto diambil, gambar tersebut ditampilkan menggunakan widget Image. Desain antarmuka pengguna yang sederhana dengan kotak berwarna pink dan tombol yang jelas membuat fitur ini mudah digunakan oleh pengguna.

## UNGUIDED

1. (Soal) Modifikasi project pemilihan gambar yang telah dikerjakan pada Tugas Pendahuluan Modul 09 agar fungsionalitas tombol dapat berfungsi untuk mengunggah gambar.
  - Ketika tombol Gallery ditekan, aplikasi akan mengambil gambar dari galeri, dan setelah gambar dipilih, gambar tersebut akan ditampilkan di dalam container.
  - Ketika tombol Camera ditekan, aplikasi akan mengambil gambar menggunakan kamera, dan setelah pengambilan gambar selesai,gambar tersebut akan ditampilkan di dalam container.
  - Ketika tombol Hapus Gambar ditekan, gambar yang ada pada container akan dihapus.

*Note: Jangan lupa sertakan source code, screenshoot output, dan deskripsi program. Kreatifitas menjadi nilai tambah.*

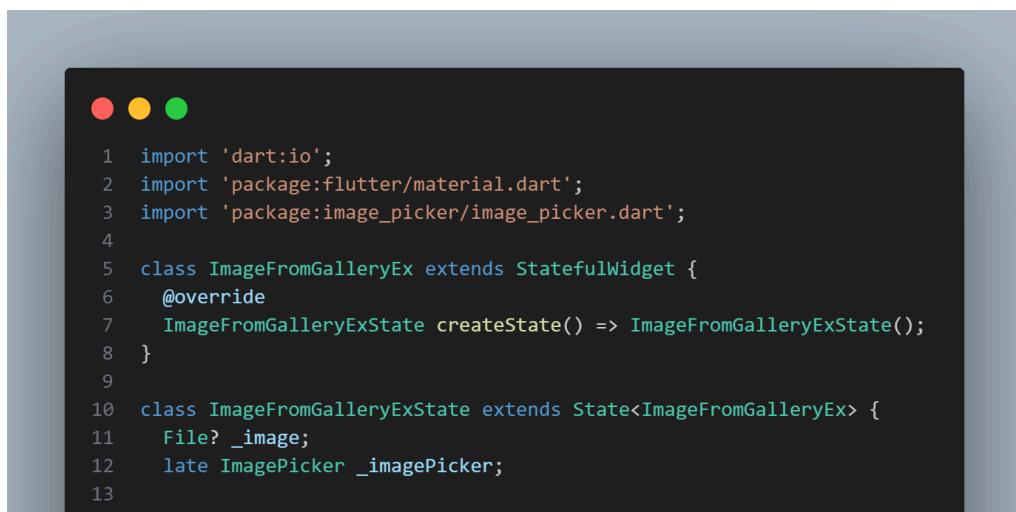
### Source Code:

#### a. main.dart



```
1 import 'package:flutter/material.dart';
2 import 'package:unguided_9/image_picker_screen.dart';
3
4 void main() => runApp(const MyApp());
5
6 class MyApp extends StatelessWidget {
7   const MyApp({super.key});
8
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      title: 'Material App',
13      home: ImageFromGalleryEx(),
14      // MyApiPage(),
15    );
16  }
17 }
```

#### b. image\_picker\_screen.dart



```
1 import 'dart:io';
2 import 'package:flutter/material.dart';
3 import 'package:image_picker/image_picker.dart';
4
5 class ImageFromGalleryEx extends StatefulWidget {
6   @override
7   ImageFromGalleryExState createState() => ImageFromGalleryExState();
8 }
9
10 class ImageFromGalleryExState extends State<ImageFromGalleryEx> {
11   File? _image;
12   late ImagePicker _imagePicker;
13 }
```

```
14     @override
15     void initState() {
16         super.initState();
17         _imagePicker = ImagePicker();
18     }
19
20     Future<void> _pickImage(ImageSource source) async {
21         try {
22             final XFile? image = await _imagePicker.pickImage(
23                 source: source,
24                 imageQuality: 50,
25                 preferredCameraDevice: CameraDevice.front,
26             );
27
28             if (image != null) {
29                 setState(() {
30                     _image = File(image.path);
31                 });
32             }
33         } catch (e) {
34             ScaffoldMessenger.of(context).showSnackBar(
35                 SnackBar(content: Text("Error picking image: $e")),
36             );
37         }
38     }
39
40     void _removeImage() {
41         setState(() {
42             _image = null;
43         });
44     }
45 }
```

```
1  @override
2  Widget build(BuildContext context) {
3      return Scaffold(
4          appBar: AppBar(
5              title: const Text('Image Picker'),
6              centerTitle: true,
7              backgroundColor: const Color.fromARGB(255, 235, 172, 57),
8          ),
9          body: Center(
10              child: Padding(
11                  padding: const EdgeInsets.all(16.0),
12                  child: Column(
13                      mainAxisAlignment: MainAxisAlignment.center,
14                      children: [
15                          // Container untuk gambar
16                          Container(
17                              width: 250,
18                              height: 250,
19                              decoration: BoxDecoration(
20                                  border: Border.all(color: Colors.grey),
21                                  borderRadius: BorderRadius.circular(16),
22                              ),
23                          child: _image != null
24                              ? ClipRRect(
25                                  borderRadius: BorderRadius.circular(16),
26                                  child: Image.file(
27                                      _image!,
28                                      fit: BoxFit.cover,
29                                  ),
30                              )
31                              : const Icon(
32                                  Icons.image,
33                                  size: 80,
34                                  color: Color.fromARGB(255, 206, 129, 129),
35                              ),
36                      ),
37                  ),
38              ),
39          ),
40      );
41  }
```

```
37         const SizedBox(height: 30),
38         // Tombol-tombol
39         Wrap(
40             alignment: WrapAlignment.center,
41             spacing: 15,
42             runSpacing: 15,
43             children: [
44                 ElevatedButton.icon(
45                     onPressed: () => _pickImage(ImageSource.gallery),
46                     icon: const Icon(
47                         Icons.photo,
48                         size: 30,
49                         color: Color.fromARGB(255, 131, 118, 82), // Mengubah warna ikon menjadi biru
50                     ),
51                     label: const Text(
52                         "Gallery",
53                         style: TextStyle(
54                             fontSize: 18,
55                             color: Colors.white,
56                         ),
57                     ),
58                     style: ElevatedButton.styleFrom(
59                         backgroundColor: const Color.fromARGB(255, 235, 172, 57),
60                         padding: const EdgeInsets.symmetric(
61                             vertical: 12,
62                             horizontal: 20,
63                         ),
64                     ),
65                 ),
66             ],
67         ),
68     ),
69 
```

```
1         ElevatedButton.icon(
2             onPressed: () => _pickImage(ImageSource.camera),
3             icon: const Icon(
4                 Icons.camera_alt,
5                 size: 30,
6                 color: Color.fromARGB(255, 131, 118, 82), // Mengubah warna ikon menjadi biru
7             ),
8             label: const Text(
9                 "Camera",
10                style: TextStyle(
11                    fontSize: 18,
12                    color: Colors.white,
13                ),
14            ),
15             style: ElevatedButton.styleFrom(
16                 backgroundColor: const Color.fromARGB(255, 235, 172, 57),
17                 padding: const EdgeInsets.symmetric(
18                     vertical: 12,
19                     horizontal: 20,
20                 ),
21             ),
22         ),
23         ElevatedButton.icon(
24             onPressed: _removeImage,
25             icon: const Icon(
26                 Icons.delete,
27                 size: 30,
28                 color: Color.fromARGB(255, 131, 118, 82), // Mengubah warna ikon menjadi biru
29             ),
30             label: const Text(
31                 "Hapus Gambar",
32                 style: TextStyle(
33                     fontSize: 18,
34                     color: Colors.white,
35                 ),
36             ),
37             style: ElevatedButton.styleFrom(
38                 backgroundColor: const Color.fromARGB(255, 235, 172, 57),
39                 padding: const EdgeInsets.symmetric(
40                     vertical: 12,
41                     horizontal: 20,
42                 ),
43             ),
44         ],
45     ],
46     ],
47     ],
48     ],
49     ],
50     ],
51 );
52 }
53 }
```

c. myapi\_page.dart

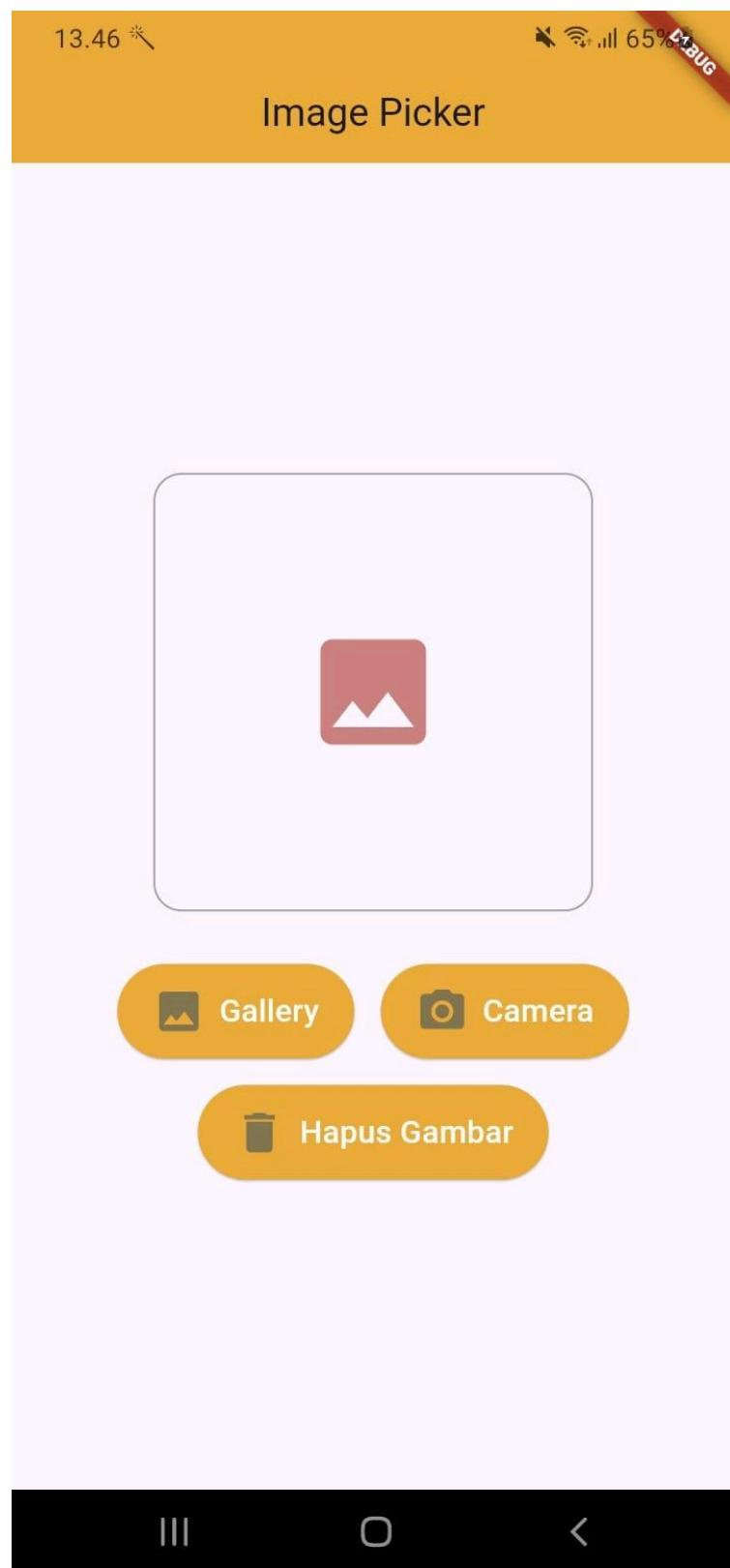
```
1 import 'dart:io';
2
3 import 'package:camera/camera.dart';
4 import 'package:flutter/material.dart';
5
6 class MyApiPage extends StatefulWidget {
7   const MyApiPage({super.key});
8
9   @override
10  State<MyApiPage> createState() => _MyApiPageState();
11 }
12
13 class _MyApiPageState extends State<MyApiPage> {
14   late CameraController _controller;
15   Future<void>? _initializeControllerFuture;
16
17   Future<void> _initializeCamera() async {
18     final cameras = await availableCameras();
19     final firstCamera = cameras.first;
20     _controller = CameraController(
21       firstCamera,
22       ResolutionPreset.high,
23     );
24 }
```

```
1
2   _initializeControllerFuture = _controller.initialize();
3   setState(() {});
4 }
5
6   @override
7   void initState() {
8     _initializeCamera();
9     super.initState();
10 }
11
12   @override
13   void dispose() {
14     _controller.dispose();
15     super.dispose();
16 }
17 }
```

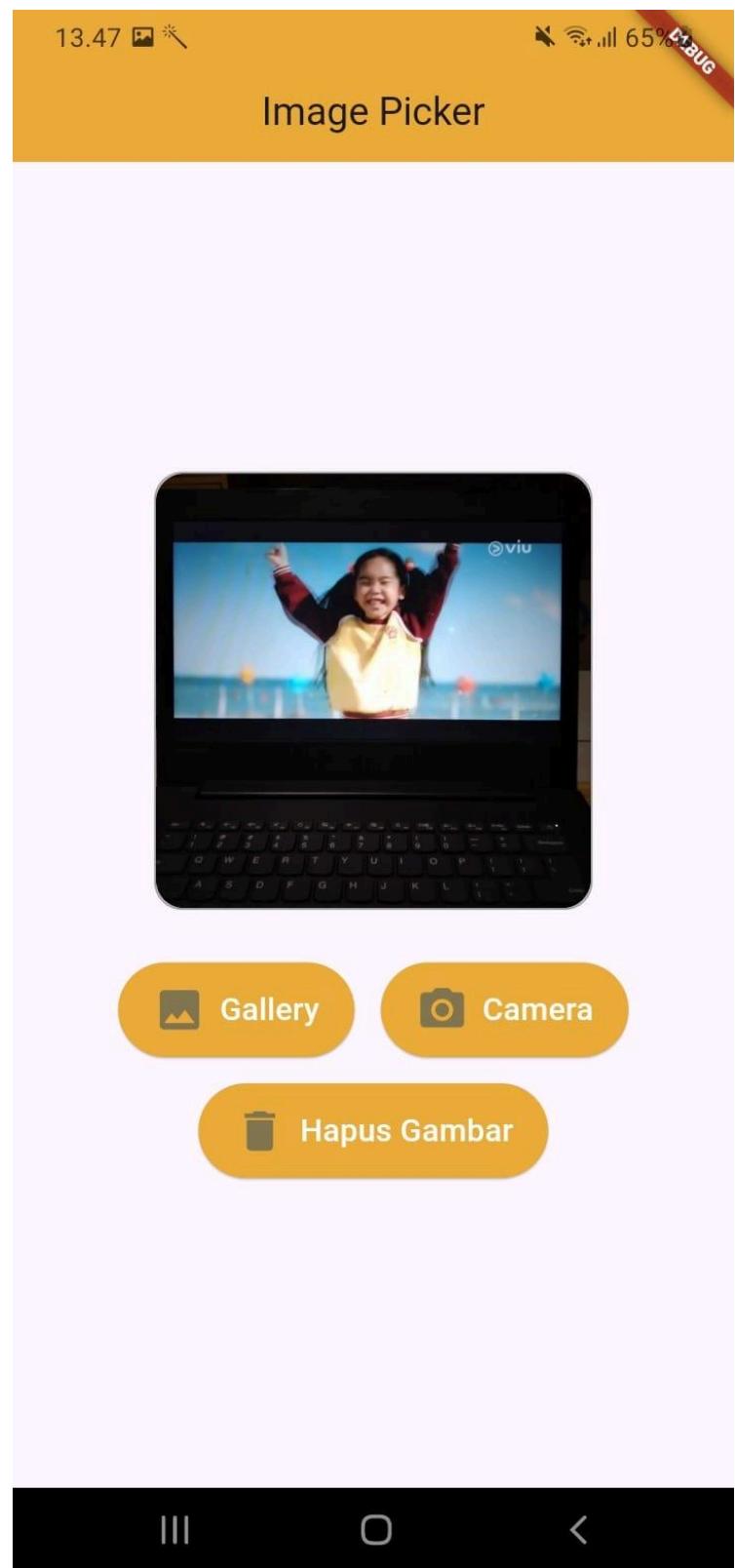
```
18     @override
19     Widget build(BuildContext context) {
20         return Scaffold(
21             appBar: AppBar(
22                 title: Text('API Perangkat Keras'),
23                 centerTitle: true,
24             ),
25             body: FutureBuilder<void>(
26                 future: _initializeControllerFuture,
27                 builder: (context, snapshot) {
28                     if (snapshot.connectionState == ConnectionState.done) {
29                         return CameraPreview(_controller);
30                     } else {
31                         return const Center(child: CircularProgressIndicator());
32                     }
33                 },
34             ),
35             floatingActionButton: FloatingActionButton(
36                 onPressed: () async {
37                     try {
38                         await _initializeControllerFuture;
39                         final Image = await _controller.takePicture();
40                         Navigator.push(
41                             context,
42                             MaterialPageRoute(
43                                 builder: (context) =>
44                                     DisplayPictureScreen(imagePath: Image.path),
45                             )));
46                         await _controller.takePicture();
47                     } catch (e) {
48                         print(e);
49                     }
50                 },
51                 child: Icon(
52                     Icons.camera_alt_rounded,
53                     size: 30,
54                 ),
55             ),
56         );
57     }
58 }
59
60 class DisplayPictureScreen extends StatelessWidget {
61     final String imagePath;
62
63     const DisplayPictureScreen({Key? key, required this.imagePath})
64         : super(key: key);
65
66     @override
67     Widget build(BuildContext context) {
68         return Scaffold(
69             appBar: AppBar(
70                 title: Text('Display Foto'),
71                 centerTitle: true,
72                 backgroundColor: Colors.grey,
73             ),
74             body: Image.file(File(imagePath)),
75         );
76     }
77 }
78 }
```

**Output:**

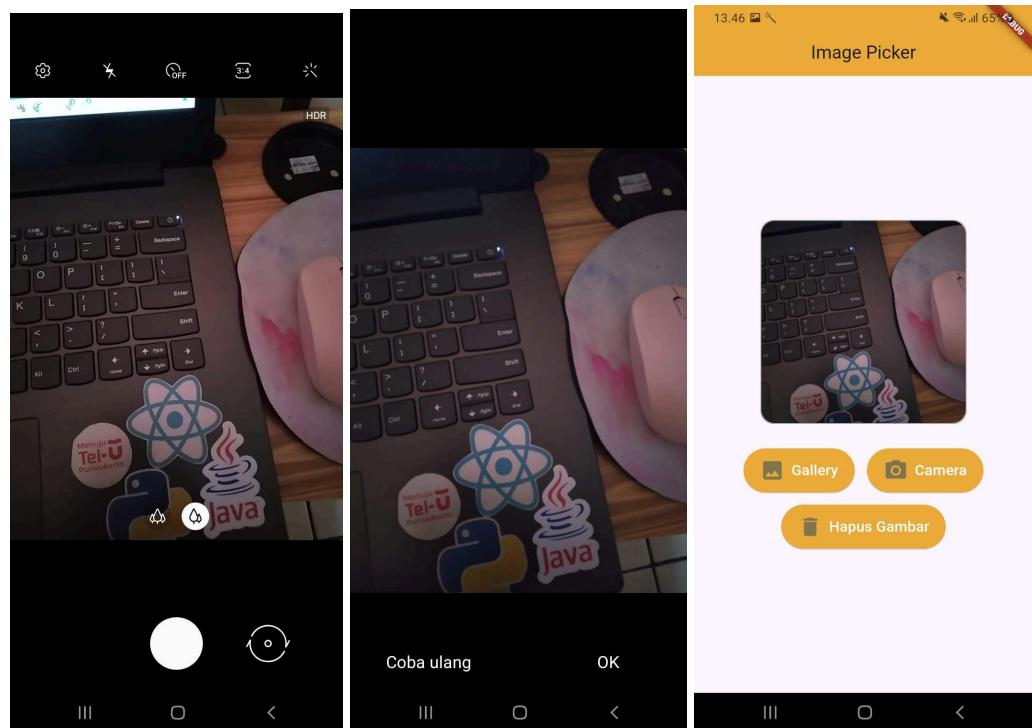
- a. Tampilan utama program.



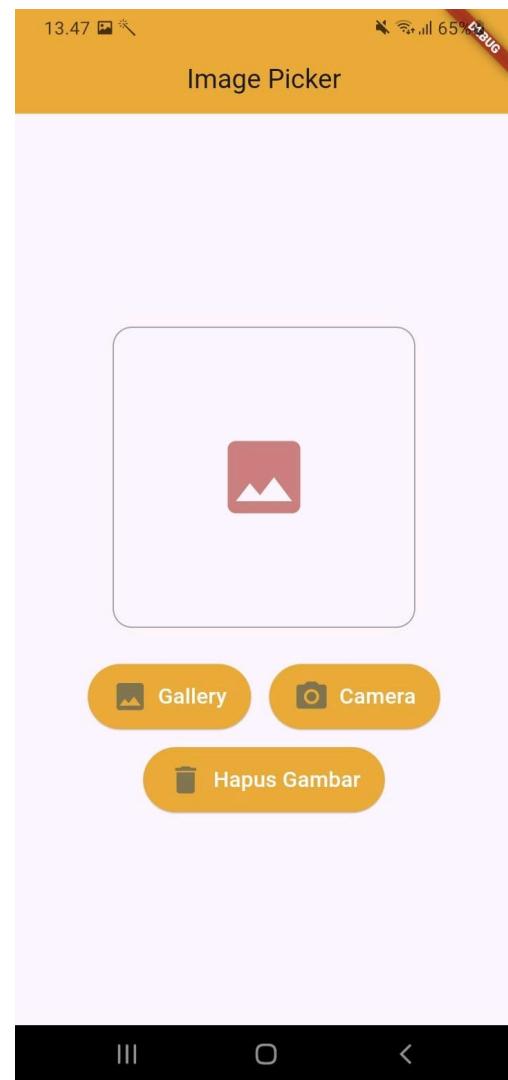
- b. Tombol Gallery. Ketika di klik, maka akan langsung masuk ke bagian gallery dan setelah memilih foto/gambar akan langsung masuk ke halaman gallery.



- c. Tombol Camera. Ketika di klik, maka akan langsung di arahkan ke kamera sehingga menghasilkan gambar yang sudah diambil.



- d. Tombol Hapus. Ketika di klik maka gambar akan hilang karena sudah dihapus.



## **Deskripsi Program:**

Output dari kode tersebut adalah sebuah aplikasi sederhana yang memungkinkan pengguna untuk memilih gambar dari galeri, mengambil gambar menggunakan kamera, atau menghapus gambar yang telah dipilih. Aplikasi ini memiliki antarmuka yang user-friendly dengan elemen-elemen berikut:

### **1. Tampilan Gambar**

Pada bagian utama layar, terdapat sebuah \*container\* yang menampilkan gambar yang telah dipilih atau diambil oleh pengguna. Jika belum ada gambar yang dipilih, ikon gambar kosong berwarna merah muda akan muncul sebagai placeholder.

### **2. Tombol Interaktif**

Di bawah tampilan gambar, terdapat tiga tombol dengan fungsi berbeda:

- **Gallery:** Tombol ini berwarna kuning keemasan dengan ikon foto (berwarna coklat). Ketika ditekan, tombol ini membuka galeri perangkat untuk memilih gambar.
- **Camera:** Tombol ini juga berwarna kuning keemasan dengan ikon kamera (berwarna coklat). Ketika ditekan, tombol ini membuka kamera perangkat untuk mengambil gambar.
- **Hapus Gambar:** Tombol terakhir memiliki ikon tempat sampah (berwarna coklat) dan digunakan untuk menghapus gambar yang telah dipilih atau diambil.

### **3. Tema Warna Aplikasi**

Aplikasi menggunakan palet warna dengan dominasi warna kuning keemasan untuk latar tombol, coklat untuk ikon, dan abu-abu untuk garis pembatas \*container\*. Palet ini memberikan kesan elegan dan hangat pada antarmuka aplikasi.

Aplikasi ini sangat berguna untuk berbagai kebutuhan seperti memilih foto profil, mengunggah dokumen gambar, atau sekadar latihan interaksi dengan galeri dan kamera di Flutter.