

**PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK
TUGAS GUIDED & UNGUIDED**

**MODUL IX
API PERANGKAT KERAS**



Disusun Oleh :
Rizky Hanifa Afania / 2211104017
SE-06-01

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

GUIDED

1. Camera API

Camera API memungkinkan pengembang untuk mengakses dan mengontrol kamera pada perangkat. Dalam Flutter, terdapat paket bernama camera yang memudahkan penerapan fitur kamera untuk mengambil foto, merekam video, dan mengakses tampilan langsung dari kamera. Paket ini sangat bermanfaat untuk membuat aplikasi yang memerlukan pengambilan gambar atau video, seperti aplikasi media sosial atau e-commerce.

Cara instalasi:

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

```
dependencies:  
  flutter:  
    sdk: flutter  
  cupertino_icons: ^1.0.8  
  camera: ^0.11.0+2
```

- b. Lalu jalankan perintah 'flutter pub get'
- c. 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" />
```

- d. Ubah minimum versi Android sdk ke 21 (atau lebih tinggi) pada file **android/app/build.gradle**

```
25  //versi Android sdk  
26  minSdkVersion 21
```

- e. Implementasi kamera pada halaman Flutter di file **myapi_page.dart**

```
import 'package:flutter/material.dart';  
import 'package:camera/camera.dart';  
import 'dart:io';  
  
class MyApiPage extends StatefulWidget {  
  const MyApiPage({super.key});  
  
  @override  
  State<MyApiPage> createState() => _MyApiPageState();  
}  
  
class _MyApiPageState extends State<MyApiPage> {  
  late CameraController _controller;
```

```

late Future<void> _initializeControllerFuture;

@override
void initState() {
  super.initState();
  _initializeCamera(); // Memanggil inisialisasi kamera di sini
}

Future<void> _initializeCamera() async {
  // Ambil daftar kamera yang tersedia di perangkat
  final cameras = await availableCameras();
  final firstCamera = cameras.first;

  // Buat kontroler kamera dan mulai kamera
  _controller = CameraController(
    firstCamera,
    ResolutionPreset.high,
  );

  // Inisialisasi kontroler kamera
  _initializeControllerFuture = _controller.initialize();
  setState(() {}); // Pastikan state diperbarui setelah kamera
  // diinisialisasi
}

@override
void dispose() {
  // Bersihkan kontroler ketika widget dihapus
  _controller.dispose();
  super.dispose();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("API Perangkat Keras"),
      centerTitle: true,
      backgroundColor: Colors.amberAccent,
    ),
    body: FutureBuilder<void>(
      future: _initializeControllerFuture, // Menunggu
      // inisialisasi kamera

```

```

        builder: (context, snapshot) {
          if (snapshot.connectionState == ConnectionState.done) {
            // Jika kamera sudah siap, tampilkan pratinjau
            return CameraPreview(_controller);
          } else {
            // Jika kamera belum siap, tampilkan loading
            return Center(child: CircularProgressIndicator());
          }
        },
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: () async {
          try {
            // Pastikan kamera sudah diinisialisasi
            await _initializeControllerFuture;

            // Ambil gambar
            final image = await _controller.takePicture();

            // Tampilkan / gunakan gambar
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) =>
                  DisplayPictureScreen(imagePath: image.path),
              ),
            );
          } catch (e) {
            print(e);
          }
        },
        child: Icon(Icons.camera_alt),
      ),
    );
  }
}

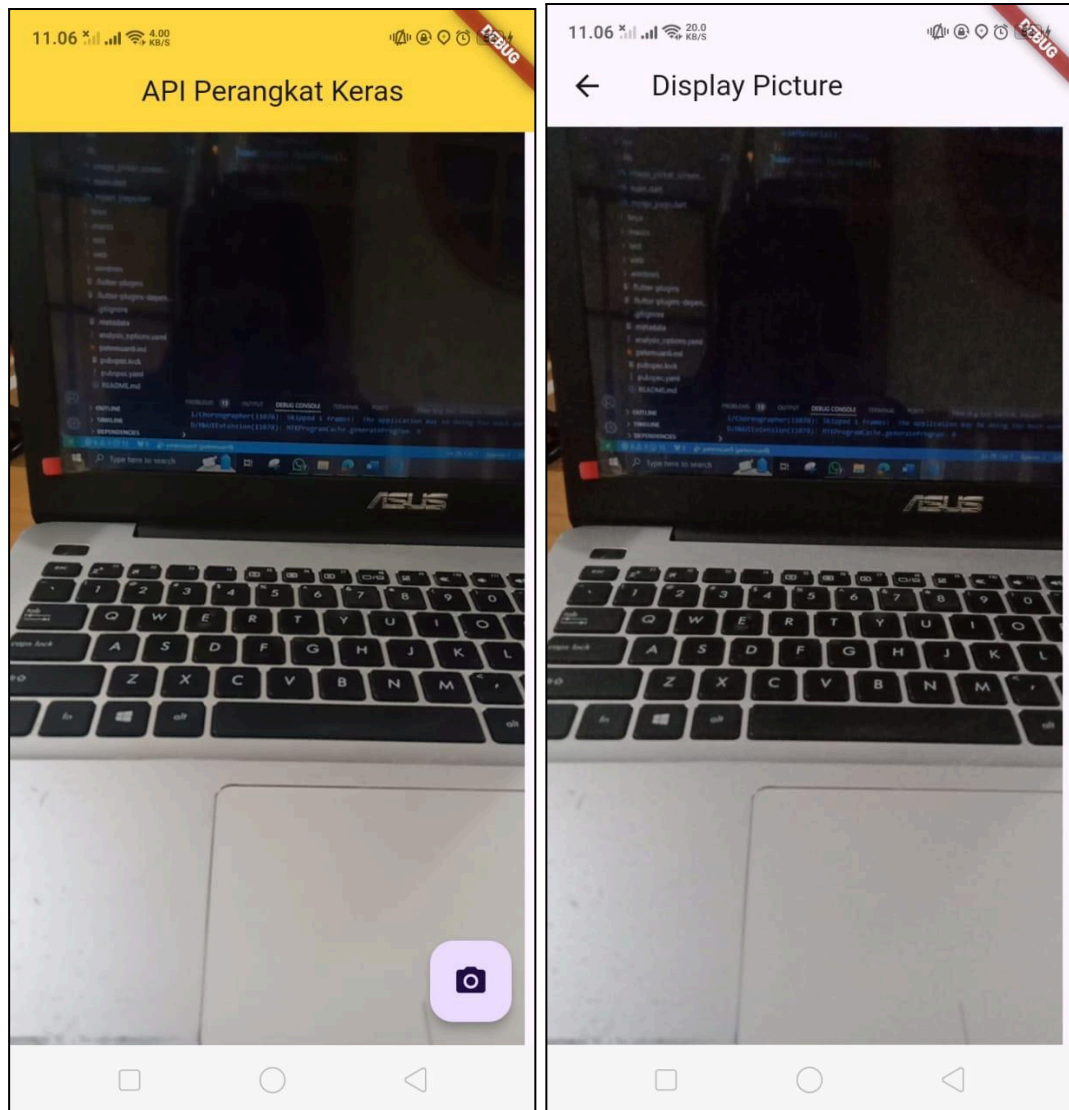
class DisplayPictureScreen extends StatelessWidget {
  final String imagePath;

  const DisplayPictureScreen({Key? key, required this.imagePath})
    : super(key: key);

```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text('Display Picture')),
    body: Image.file(File(imagePath)),
  );
}
```

Output:



Saat ingin mengambil gambar melalui kamera, pengguna bisa menekan icon kamera yang ada di pojok kanan bawah untuk mengakses kamera perangkat. Lalu pengguna bisa memotret dan mengonfirmasi, jika sudah, hasil gambar akan ditampilkan di halaman “Display Picture”.

2. Media API

Media API adalah kumpulan alat dan pustaka yang mendukung pengelolaan serta interaksi dengan media seperti gambar, video, dan audio. Flutter tidak menyediakan API media bawaan untuk semua fungsi ini, tetapi memungkinkan penggunaan paket tambahan, seperti Image Picker, untuk mengakses galeri perangkat. Konfigurasi tambahan diperlukan untuk iOS, sementara Android tidak memerlukannya.

a. File `image_picker_screen.dart`

```
import 'dart:io';

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

class ImageFromGalleryEx extends StatefulWidget {
  final ImageSourceType type;
  ImageFromGalleryEx(this.type);

  @override
  ImageFromGalleryExState createState() =>
    ImageFromGalleryExState(this.type);
}

class ImageFromGalleryExState extends State<ImageFromGalleryEx> {
  File? _image;
  late ImagePicker imagePicker;
  final ImageSourceType type;

  ImageFromGalleryExState(this.type);

  @override
  void initState() {
    super.initState();
    imagePicker = ImagePicker();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(type == ImageSourceType.camera
          ? "Image from Camera"
          : "Image from Gallery"),

```

```

),
body: Column(
  children: <Widget>[
    SizedBox(height: 52),
    Center(
      //mengambil gambar dari camera atau gallery
      child: GestureDetector(
        onTap: () async {
          //operasi ternary untuk memilih sumber gambar
          var source = type == ImageSourceType.camera
            ? ImageSource.camera
            : ImageSource.gallery;

          //menyimpan gambar pada variabel image
          XFile? image = await imagePicker.pickImage(
            source: source,
            imageQuality: 50,
            preferredCameraDevice: CameraDevice.front);

          if (image != null) {
            setState(() {
              _image = File(image.path);
            });
          }
        },
        child: Container(
          width: 200,
          height: 200,
          decoration: BoxDecoration(
            color: Colors.red[200],
          ),

          // menampilkan gambar dari camera atau gallery
          child: _image != null
            ? Image.file(
                _image!,
                width: 200.0,
                height: 200.0,
                fit: BoxFit.fitHeight,
              )

            // jika tidak ada gambar yang dipilih
            : Container(

```

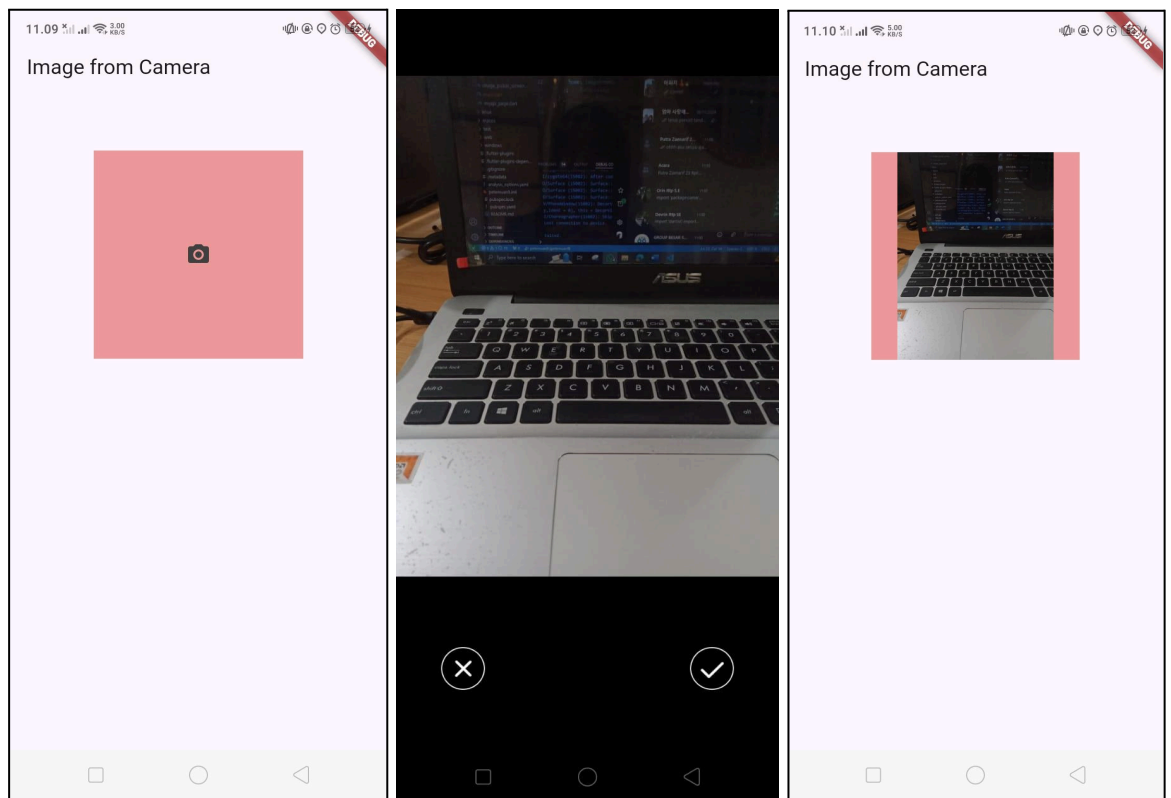
```

        decoration: BoxDecoration(
          color: Colors.red[200],
        ),
        width: 200,
        height: 200,
        child: Icon(
          Icons.camera_alt,
          color: Colors.grey[800],
        ),
      ),
    ),
  ),
],
),
);
}
}

enum ImageSourceType { camera, gallery }

```


Output:



Halaman utama dari bagian ini adalah “Image from Camera” dengan container (kotak) pink yang berisi ikon camera. Saat area kotak pink di klik maka pengguna akan bisa mengakses kamera perangkat. Lalu pengguna bisa memotret dan mengonfirmasi dengan klik tanda centang. Selanjutnya hasil gambar akan ditampilkan di kotak pink.

UNGUIDED

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, screenshot output, dan deskripsi program.

Kreatifitas menjadi nilai tambah

Source Code:

a. main.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:tugas_pendahuluan_9/image_picker_screen.dart';
3
4 void main() {
5   runApp(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: 'Unguided Week 9',
15       theme: ThemeData(
16         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
17         useMaterial3: true,
18       ),
19       home: ImagePickerScreen(),
20     );
21   }
22 }
```

b. image_picker_screen.dart

```
1 import 'dart:io';
2 import 'package:flutter/material.dart';
3 import 'package:image_picker/image_picker.dart';
4 import 'package:dotted_border/dotted_border.dart';
5
6 class ImagePickerScreen extends StatefulWidget {
7   @override
8   _ImagePickerScreenState createState() => _ImagePickerScreenState();
9 }
10
11 class _ImagePickerScreenState extends State<ImagePickerScreen> {
12   File? _image; // Variable untuk menyimpan gambar yang diambil
13   final ImagePicker _picker = ImagePicker();
14
15   Future<void> _pickImage(ImageSource source) async {
16     final XFile? image = await _picker.pickImage(source: source, imageQuality: 50);
17
18     if (image != null) {
19       setState(() {
20         _image = File(image.path);
21       });
22     }
23   }
24
25   void _deleteImage() {
26     setState(() {
27       _image = null;
28     });
29   }
30
31   @override
32   Widget build(BuildContext context) {
33     return Scaffold(
34       appBar: AppBar(
35         title: Center(
36           child: Text('Latihan Memilih Gambar'),
37         ),
38       backgroundColor: Colors.yellow.shade300,
39     ),
40     body: Center(
41       child: Column(
42         children: [
43           SizedBox(height: 30),
44           DottedBorder(
45             color: Colors.grey,
46             strokeWidth: 2,
47             dashPattern: [5, 4],
48             borderType: BorderType.RRect,
49             radius: Radius.circular(12),
50             child: Container(
```

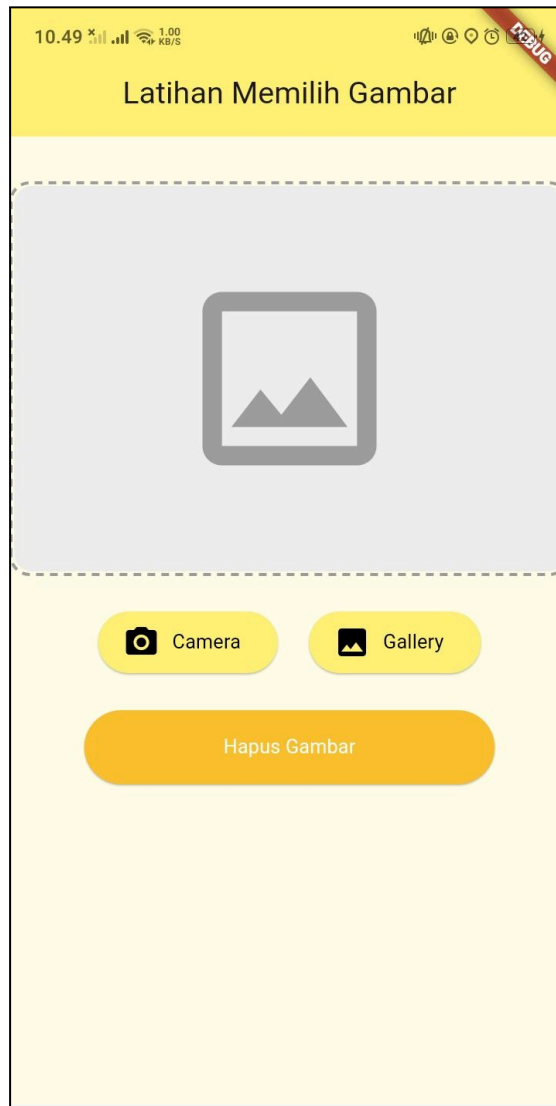
```

1    width: 400,
2      height: 250,
3      decoration: BoxDecoration(
4        color: Colors.grey.shade200,
5        borderRadius: BorderRadius.circular(12),
6      ),
7      child: Center(
8        child: _image != null
9          ? Image.file(
10             _image!,
11             width: 400,
12             height: 250,
13             fit: BoxFit.cover,
14           )
15          : Icon(
16             Icons.image_outlined,
17             size: 150,
18             color: Colors.grey,
19           ),
20      ),
21    ),
22  ),
23  SizedBox(height: 20),
24  Row(
25    mainAxisAlignment: MainAxisAlignment.center,
26    children: [
27      ElevatedButton.icon(
28        onPressed: () => _pickImage(ImageSource.camera),
29        icon: Icon(Icons.camera_alt, color: Colors.black),
30        label: Text(
31          'Camera',
32          style: TextStyle(color: Colors.black),
33        ),
34        style: ElevatedButton.styleFrom(
35          backgroundColor: Colors.yellow[300],
36        ),
37      ),
38      SizedBox(width: 20),
39      ElevatedButton.icon(
40        onPressed: () => _pickImage(ImageSource.gallery),
41        icon: Icon(
42          Icons.photo,
43          color: Colors.black,
44        ),
45        label: Text(
46          'Gallery',
47          style: TextStyle(color: Colors.black),
48        ),
49        style: ElevatedButton.styleFrom(
50          backgroundColor: Colors.yellow[300],
51        ),
52      ),
53    ],
54  ),
55  SizedBox(height: 20),
56  ElevatedButton(
57    onPressed: _deleteImage,
58    child: Text(
59      'Hapus Gambar',
60      style: TextStyle(color: Colors.white),
61    ),
62    style: ElevatedButton.styleFrom(
63      backgroundColor: Colors.yellow.shade700,
64      padding: EdgeInsets.symmetric(horizontal: 90, vertical: 15),
65    ),
66  ),
67 ],
68 ),
69 ),
70 backgroundColor: Colors.yellow.shade50,
71 );
72 }
73 }
74

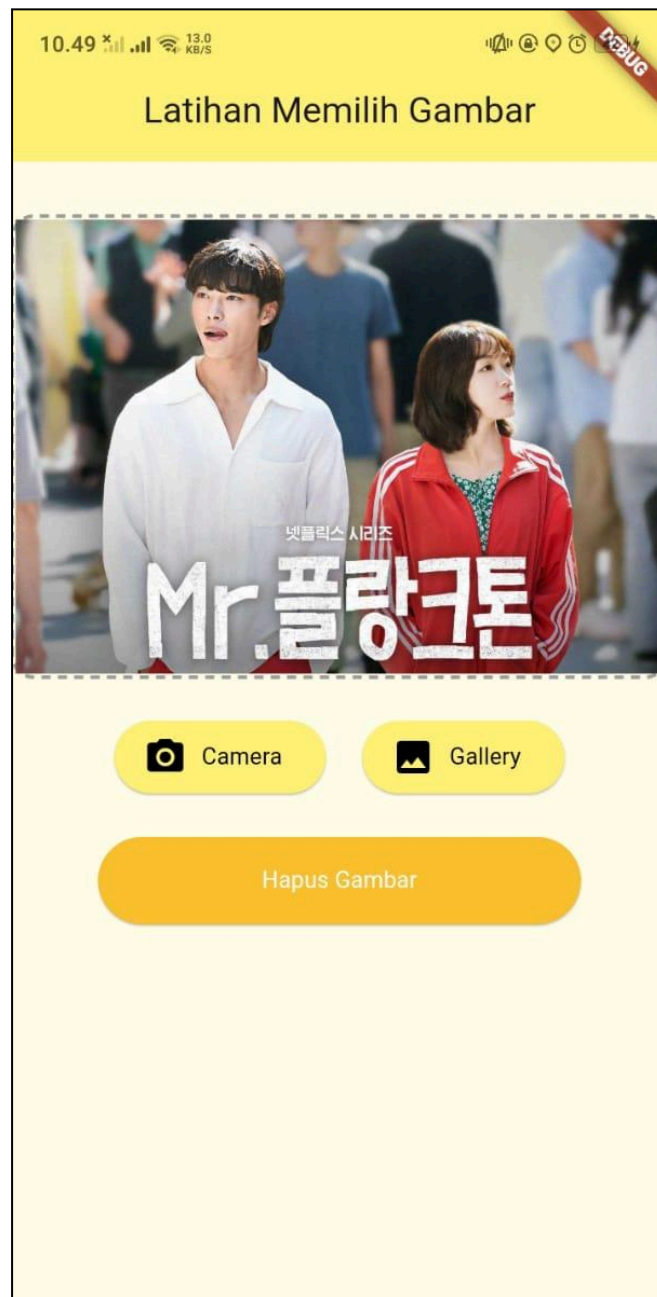
```

Output:

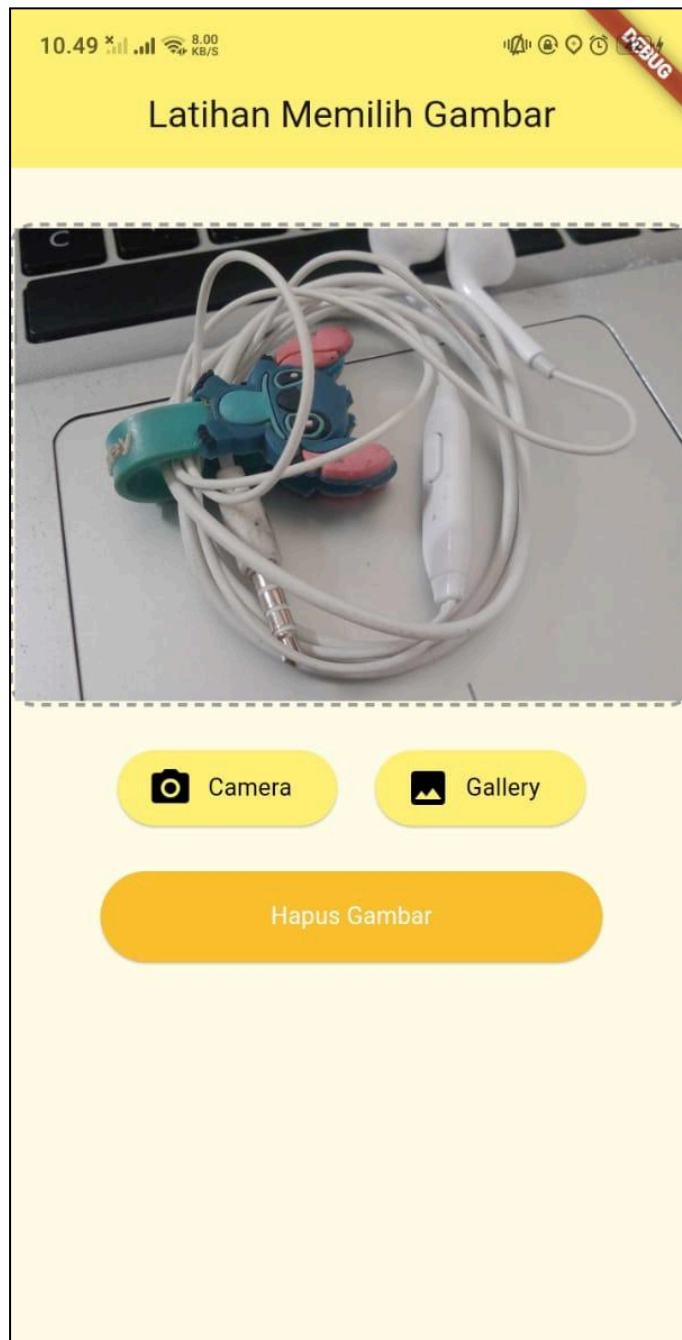
- a. Berikut ini adalah tampilan utama dari program:



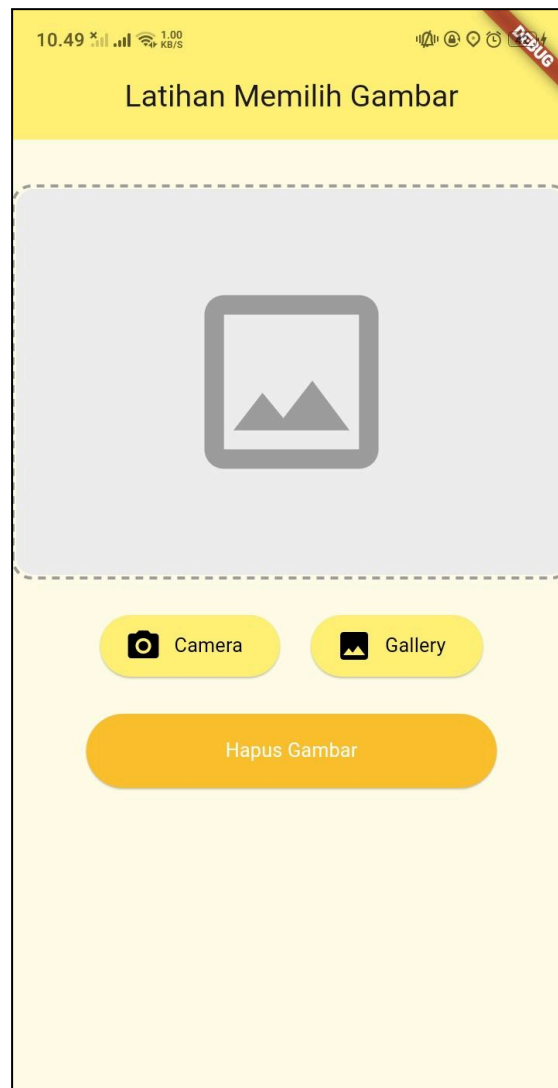
- b. Berikut adalah tampilan ketika tombol Gallery ditekan, aplikasi akan mengambil gambar dari galeri, dan setelah gambar dipilih, gambar tersebut akan ditampilkan di dalam container.



- c. Berikut adalah ketika tombol Camera ditekan, aplikasi akan mengambil gambar menggunakan kamera, dan setelah pengambilan gambar selesai, gambar tersebut akan ditampilkan di dalam container.



- d. Berikut adalah ketika tombol Hapus Gambar ditekan, gambar yang ada pada container akan dihapus.



Deskripsi Program:

Di dalam program ini terdapat tiga poin penting yang memungkinkan pengguna untuk mengambil gambar menggunakan kamera, memilih gambar dari galeri perangkat, dan menghapus gambar. Berikut penjelasan masing-masing point:

a. Mengambil gambar menggunakan kamera.

Saat membuka kamera program akan menjalankan package **image_picker**. Kemudian saat button Camera ditekan, maka akan memanggil fungsi **_pickImage** dengan parameter **ImageSource**. Camera untuk membuka kamera di perangkat. Setelah itu, pengguna bisa memotret dan mengonfirmasi apakah yakin menggunakan gambar tersebut. Lalu gambar akan ditampilkan di dalam container halaman utama.

b. Memilih gambar dari galeri perangkat.

Jika pengguna ingin memilih gambar dari galeri, maka bisa menekan tombol Gallery. Selanjutnya program akan memanggil fungsi **_pickImage** dengan parameter **ImageSource.gallery**, yang membuka galeri perangkat. Lalu pengguna bisa memilih gambar dari galeri, dan gambar yang dipilih akan muncul dalam kotak garis putus-putus di halaman utama program

c. Menghapus gambar yang telah dipilih atau diambil.

Jika pengguna ingin menghapus gambar, maka pengguna bisa menekan tombol Hapus. Ketika tombol ini ditekan, program akan memanggil fungsi **_deleteImage**. Fungsi ini mengatur variabel **_image** menjadi null, yang menghapus tampilan gambar dan mengembalikannya ke ikon placeholder. Fitur ini memberikan fleksibilitas bagi pengguna untuk memilih ulang gambar atau mengambil gambar baru tanpa perlu keluar dari aplikasi.