

**TUGAS PENDAHULUAN / TUGAS UNGUIDED  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL II  
PENGENALAN FLUTTER**



**Disusun Oleh :**

**Farhan Kurniawan / 231110473**

**SE-07-02**

**Asisten Praktikum :**

**Zulfa Mustafa Akhyar Iswahyudi  
Yoga Eka Pratama**

**Dosen Pengampu :**

**Yudha Islami Sulistya, S.Kom., M.Cs.**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

## TUGAS PENDAHULUAN / TUGAS UNGUIDED

### A. SOAL

(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.

### B. JAWABAN

Source code

```
// camera.dart
import 'dart:async';
import 'package:flutter/material.dart';
import 'package:camera/camera.dart';

List<CameraDescription>? cameras;

class CameraApp extends StatefulWidget {
  @override
  _CameraAppState createState() => _CameraAppState();
}

class _CameraAppState extends State<CameraApp> {
  CameraController? controller;

  Future<void> initCamera() async {
    WidgetsFlutterBinding.ensureInitialized();
    cameras = await availableCameras();

    if (cameras != null && cameras!.isNotEmpty) {
```

```

    print(cameras);

    controller = CameraController(cameras![0], ResolutionPreset.max);
    controller?.initialize().then((_) {
      if (!mounted) {
        return;
      }
      setState(() {});
    });
  }
}

@override
void initState() {
  super.initState();
  initCamera();
}

@override
void dispose() {
  controller?.dispose();
  super.dispose();
}

@override
Widget build(BuildContext context) {
  if (controller == null || !controller!.value.isInitialized) {
    return Container();
  }
  return MaterialApp(home: Scaffold(body: CameraPreview(controller!)));
}

// image_picker.dart
import 'dart:io';

```

```

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

enum ImageSourceType { camera, gallery }

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]),
      ),
    ),
  ),
),
],
),
);
}
}

```

**// main.dart**

```

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

```

```

List<CameraDescription> cameras = [];

```

```

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();

  try {
    cameras = await availableCameras();
  } catch (e) {
    print("Error initializing cameras: $e");
  }
}

```

```
runApp(const MainApp());
}

class MainApp extends StatelessWidget {
  const MainApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'API Perangkat Keras',
      theme: ThemeData(primarySwatch: Colors.blueGrey),
      home: const HomePage(),
    );
  }
}

class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  XFile? _selectedImage;
  final ImagePicker picker = ImagePicker();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("API Perangkat Keras"),
        centerTitle: true,
```

```

),
body: Center(
  child: Padding(
    padding: const EdgeInsets.all(24.0),
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        // Tampilkan gambar jika ada
        Container(
          width: 220,
          height: 220,
          decoration: BoxDecoration(
            border: Border.all(color: Colors.blueGrey),
            borderRadius: BorderRadius.circular(12),
          ),
          child: _selectedImage != null
            ? Image.file(File(_selectedImage!.path), fit: BoxFit.cover)
            : const Center(
                child: Text(
                  "Belum ada gambar",
                  style: TextStyle(color: Colors.grey),
                ),
              ),
        ),
      ],
    ),
  ),
  const SizedBox(height: 20),

  /// Tombol Gallery
  ElevatedButton(
    onPressed: () async {
      final image = await picker.pickImage(
        source: ImageSource.gallery,
        imageQuality: 75,
      );
    },
  ),

```

```

        if (image != null) {
          setState(() {
            _selectedImage = image;
          });
        }
      },
      child: const Text("Gallery"),
    ),

    const SizedBox(height: 16),

    /// Tombol Camera
    ElevatedButton(
      onPressed: () async {
        final image = await picker.pickImage(
          source: ImageSource.camera,
          imageQuality: 75,
        );

        if (image != null) {
          setState(() {
            _selectedImage = image;
          });
        }
      },
      child: const Text("Camera"),
    ),

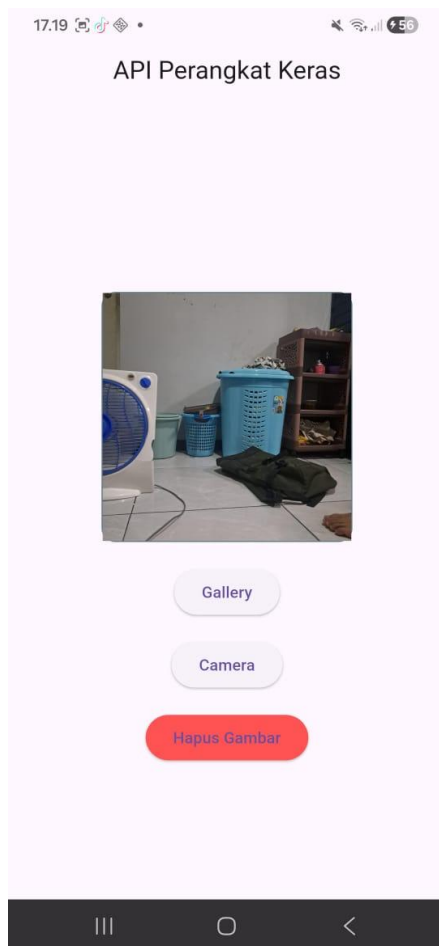
    const SizedBox(height: 16),

    /// Tombol Hapus Gambar
    ElevatedButton(
      onPressed: () {
        setState(() {
          _selectedImage = null;

```

```
        });  
    },  
    style: ElevatedButton.styleFrom(  
        backgroundColor: Colors.redAccent,  
    ),  
    child: const Text("Hapus Gambar"),  
),  
],  
),  
),  
),  
);  
}  
}
```

## Screenshot Output



## **Deskripsi Program**

Program ini merupakan aplikasi Flutter yang memanfaatkan API perangkat keras untuk mengambil atau memilih gambar menggunakan kamera dan galeri pada perangkat Android. Aplikasi ini menggunakan algoritma pemrosesan input berupa pemanggilan fungsi `pickImage()` dari pustaka `image_picker`, yang bekerja dengan menentukan sumber gambar apakah dari kamera atau galeri, kemudian menyimpan hasilnya dalam variabel `_selectedImage`. Ketika pengguna menekan tombol Gallery atau Camera, program menjalankan algoritma pemilihan sumber gambar, membuka galeri atau kamera, menunggu pengguna memilih atau mengambil gambar, lalu menampilkan hasilnya dalam sebuah container. Program juga memiliki algoritma pembersihan data, yaitu ketika tombol Hapus Gambar ditekan, variabel yang menyimpan gambar diatur menjadi null sehingga container kembali kosong. Pada akhirnya, output yang dihasilkan berupa tampilan gambar yang dipilih atau diambil oleh pengguna, atau container kosong jika gambar sudah dihapus, sehingga aplikasi berjalan secara interaktif sesuai tindakan pengguna.