

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

**MODUL IX
API PERANGKAT KERAS**



Disusun Oleh :

Nadia Putri Rahmaniar / 2211104012

S1 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 memberikan kemampuan bagi pengembang untuk mengakses dan mengelola kamera pada perangkat. Dalam Flutter, tersedia paket bernama ****camera**** yang mempermudah implementasi fitur kamera, seperti mengambil foto, merekam video, dan menampilkan feed langsung dari kamera. Paket ini sangat berguna untuk aplikasi yang membutuhkan fungsi pengambilan gambar atau video, misalnya pada 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' di terminal
- c) Implementasi kamera pada halaman Flutter di file myapi_page.dart

```
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:camera/camera.dart';  
import 'package:flutter/material.dart';  
  
class MyApiPage extends StatefulWidget {  
  @override  
  _MyApiPageState createState() => _MyApiPageState();  
}
```

```

}

class _MyApiPageState extends State<MyApiPage> {
  late CameraController _controller;
  Future<void>? _initializeControllerFuture;

  @override
  void initState() {
    super.initState();
    _initializeControllerFuture = _initializeCamera();
  }

  Future<void> _initializeCamera() async {
    final cameras = await availableCameras();
    final firstCamera = cameras.first;

    _controller = CameraController(
      firstCamera,
      ResolutionPreset.medium,
    );

    await _controller.initialize();
  }

  @override
  void dispose() {
    _controller.dispose();
    super.dispose();
  }

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

```

```
        backgroundColor: Color(0xFFFFD700),
      ),
      body: FutureBuilder<void>(
        future: _initializeControllerFuture,
        builder: (context, snapshot) {
          if (snapshot.connectionState == ConnectionState.done) {
            return CameraPreview(_controller);
          } else {
            return Center(child: CircularProgressIndicator());
          }
        },
      ),

      // floatingActionButton: FloatingActionButton(
      //   onPressed: () {
      //     Navigator.pop(context);
      //   },
      //   child: Icon(Icons.arrow_back),
      // ),
    );
  }
}
```

Output :



2. Media API

Media API menyediakan berbagai alat dan pustaka untuk mengelola dan berinteraksi dengan media, seperti foto, video, dan audio. Dalam Flutter, tidak semua fungsi media disediakan secara bawaan, tetapi pengembang dapat menggunakan paket tambahan seperti ****Image Picker**** untuk mengakses galeri perangkat. Untuk perangkat iOS, diperlukan pengaturan tambahan, sedangkan pada Android, konfigurasi tersebut tidak dibutuhkan.

a. image_picker.dart

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

class ImagePickerScreen extends StatefulWidget {
  @override
  _ImagePickerScreenState createState() =>
    _ImagePickerScreenState();
}

class _ImagePickerScreenState extends State<ImagePickerScreen> {
  File? _image; // Variable untuk menyimpan gambar yang diambil
  final ImagePicker _picker = ImagePicker();

  Future<void> _pickImage(ImageSource source) async {
    final XFile? image = await _picker.pickImage(source: source,
imageQuality: 50);

    if (image != null) {
      setState(() {
        _image = File(image.path);
      });
    }
  }

  void _deleteImage() {
```

```
    setState(() {  
      _image = null;  
    });  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Center(  
          child: Text('Latihan Memilih Gambar'),  
        ),  
        backgroundColor: Colors.yellow.shade300,  
      ),  
      body: Center(  
        child: Column(  
          children: [  
            SizedBox(height: 30),  
            DottedBorder(  
              color: Colors.grey,  
              strokeWidth: 2,  
              dashPattern: [5, 4],  
              borderType: BorderType.RRect,  
              radius: Radius.circular(12),  
              child: Container(  
                width: 400,  
                height: 250,  
                decoration: BoxDecoration(  
                  color: Colors.grey.shade200,  
                  borderRadius: BorderRadius.circular(12),  
                ),  
                child: Center(  
                  child: _image != null  
                    ? Image.file(  
                      _image!,  
                      width: 400,
```

```

        height: 250,
        fit: BoxFit.cover,
      )
      : Icon(
        Icons.image_outlined,
        size: 150,
        color: Colors.grey,
      ),
    ),
  ),
),
SizedBox(height: 20),
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    ElevatedButton.icon(
      onPressed: () =>
_pickImage(ImageSource.camera),
      icon: Icon(Icons.camera_alt, color:
Colors.black),
      label: Text(
        'Camera',
        style: TextStyle(color: Colors.black),
      ),
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.yellow[300],
      ),
    ),
    SizedBox(width: 20),
    ElevatedButton.icon(
      onPressed: () =>
_pickImage(ImageSource.gallery),
      icon: Icon(
        Icons.photo,
        color: Colors.black,
      ),

```

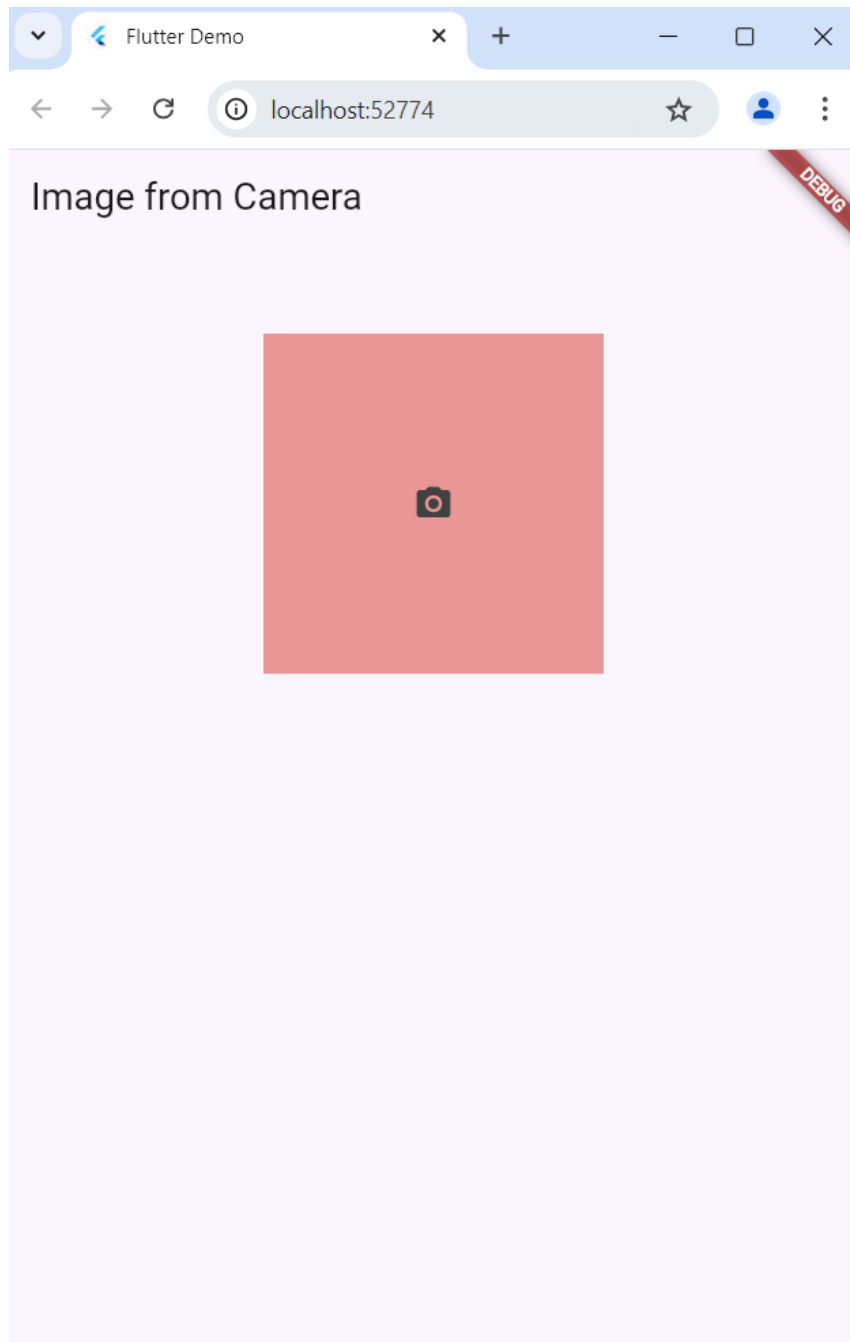


```

        label: Text(
          'Gallery',
          style: TextStyle(color: Colors.black),
        ),
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.yellow[300],
        ),
      ),
    ],
  ),
  SizedBox(height: 20),
  ElevatedButton(
    onPressed: _deleteImage,
    child: Text(
      'Hapus Gambar',
      style: TextStyle(color: Colors.white),
    ),
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.yellow.shade700,
      padding: EdgeInsets.symmetric(horizontal: 90,
vertical: 15),
    ),
  ),
),
],
),
),
backgroundColor: Colors.yellow.shade50,
);
}
}

```

Output :



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. Kreativitas menjadi nilai tambah

1. Source Code

a. main.dart

```
import 'package:flutter/material.dart';
import 'package:unguided_9/image_picker_screen.dart';

void main() => runApp(const MyApp());

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Material App',
      home: ImageFromGalleryEx(),
      // MyApiPage(),
    );
  }
}
```

b. image_picker.dart

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

class ImageFromGalleryEx extends StatefulWidget {
  @override
  ImageFromGalleryExState createState() =>
  ImageFromGalleryExState();
}

class ImageFromGalleryExState extends
  State<ImageFromGalleryEx> {
  File? _image;
  late ImagePicker _imagePicker;

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

  Future<void> _pickImage(ImageSource source) async {
    try {
      final XFile? image = await _imagePicker.pickImage(
        source: source,
        imageQuality: 50,
        preferredCameraDevice: CameraDevice.front,
      );

      if (image != null) {
        setState(() {
          _image = File(image.path);
        });
      }
    } catch (e) {
```

```

        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text("Error picking image: $e")),
        );
      }
    }

void _removeImage() {
  setState(() {
    _image = null;
  });
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Image Picker'),
      centerTitle: true,
      backgroundColor: const Color.fromARGB(255, 235, 172,
57),
    ),
    body: Center(
      child: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            // Container untuk gambar
            Container(
              width: 250,
              height: 250,
              decoration: BoxDecoration(
                border: Border.all(color: Colors.grey),
                borderRadius: BorderRadius.circular(16),
              ),
              child: _image != null

```

```

        ? ClipRRect(
            borderRadius:
BorderRadius.circular(16),
            child: Image.file(
                _image!,
                fit: BoxFit.cover,
            ),
        )
        : const Icon(
            Icons.image,
            size: 80,
            color: Color.fromARGB(255, 206, 129,
129),
        ),
    ),
    const SizedBox(height: 30),
    // Tombol-tombol
    Wrap(
        alignment: WrapAlignment.center,
        spacing: 15,
        runSpacing: 15,
        children: [
            ElevatedButton.icon(
                onPressed: () =>
_pickImage(ImageSource.gallery),
                icon: const Icon(
                    Icons.photo,
                    size: 30,
                    color: Color.fromARGB(255, 131, 118,
82), // Mengubah warna ikon menjadi biru
                ),
                label: const Text(
                    "Gallery",
                    style: TextStyle(
                        fontSize: 18,
                        color: Colors.white,

```

```

        ),
    ),
    style: ElevatedButton.styleFrom(
        backgroundColor: const
Color.fromARGB(255, 235, 172, 57),
        padding: const EdgeInsets.symmetric(
            vertical: 12,
            horizontal: 20,
        ),
    ),
),
),
ElevatedButton.icon(
        onPressed: () =>
_pickImage(ImageSource.camera),
        icon: const Icon(
            Icons.camera_alt,
            size: 30,
            color: Color.fromARGB(255, 131, 118,
82), // Mengubah warna ikon menjadi biru
        ),
        label: const Text(
            "Camera",
            style: TextStyle(
                fontSize: 18,
                color: Colors.white,
            ),
        ),
    ),
    style: ElevatedButton.styleFrom(
        backgroundColor: const
Color.fromARGB(255, 235, 172, 57),
        padding: const EdgeInsets.symmetric(
            vertical: 12,
            horizontal: 20,
        ),
    ),
),
),
),

```

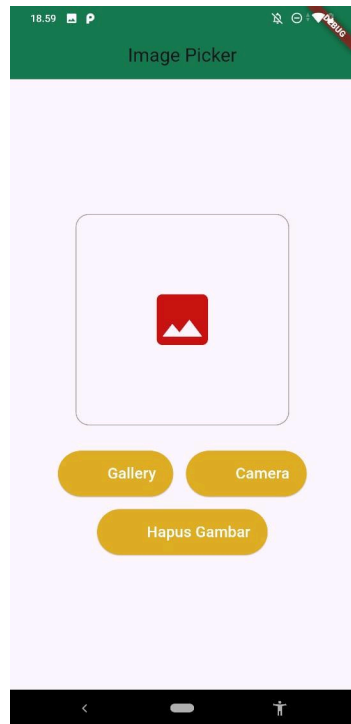
```

        ElevatedButton.icon(
            onPressed: _removeImage,
            icon: const Icon(
                Icons.delete,
                size: 30,
                color: Color.fromARGB(255, 131, 118,
82), // Mengubah warna ikon menjadi biru
            ),
            label: const Text(
                "Hapus Gambar",
                style: TextStyle(
                    fontSize: 18,
                    color: Colors.white,
                ),
            ),
            style: ElevatedButton.styleFrom(
                backgroundColor: const
Color.fromARGB(255, 235, 172, 57),
                padding: const EdgeInsets.symmetric(
                    vertical: 12,
                    horizontal: 20,
                ),
            ),
        ),
    ),
    1,
    ),
    1,
    ),
    ),
    ),
    );
}
}

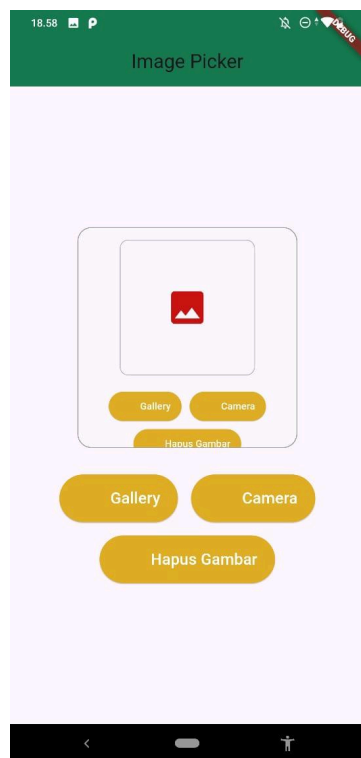
```


c. Screenshot output

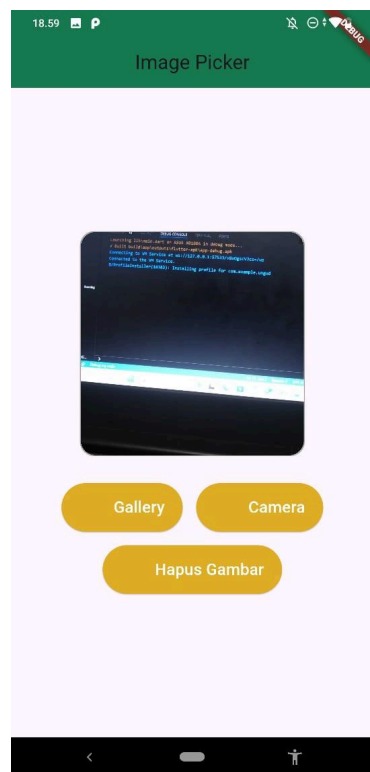
Tampilan utama dari program:



Tampilan gambar yang diambil dari galeri:



Tampilan camera:



2. Deskripsi Program

- a. **Mengambil gambar melalui kamera:** package `image_picker` akan dijalankan dalam program saat kamera dibuka. Setelah itu, ketika tombol kamera ditekan, program akan memanggil fungsi `_pickImage` dengan parameter `ImageSource` untuk membuka kamera. Setelah itu, pengguna dapat memotret untuk memastikan bahwa mereka menggunakan gambar tersebut. Setelah itu, gambar akan ditampilkan di halaman utama.
- b. **Memilih gambar dari galeri:** jika pengguna ingin memilih gambar dari galeri, mereka dapat menekan tombol Gallery. Kemudian program akan memanggil fungsi `_pickImage` dengan parameter `ImageSource.gallery`, yang membuka galeri perangkat. Setelah itu, pengguna dapat memilih gambar dari galeri dan gambar tersebut akan muncul dalam kotak garis putus-putus di halaman utama aplikasi.
- c. **Hapus gambar yang telah dipilih:** untuk hapus gambar, pengguna dapat menekan tombol "Hapus Gambar". Ketika tombol ini ditekan, program akan memanggil fungsi `_deleteImage`, yang mengatur variabel `_image` menjadi null. Ini akan menghapus tampilan gambar dan mengembalikannya ke ikon placeholder. Fitur ini memungkinkan pengguna memilih ulang gambar atau mengambil gambar baru tanpa meninggalkan aplikasi.