

LAPORAN PRAKTIKUM
MODUL IX
API PERANGKAT KERAS



Disusun Oleh :

Dwi Candra Pratama / 2211104035 SE-06-02

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

Tujuan Praktikum

- Mahasiswa mampu memahami konsep layout pada Flutter
- Mahasiswa dapat mengimplementasikan desain user interface pada Flutter

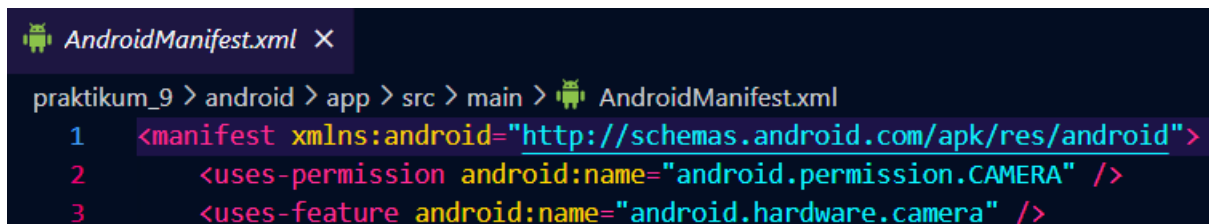
1. Camera API

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:

- a. Tambahkan paket camera yang ada pada Pub Dev di pubspec.yaml
- b. Lalu jalankan perintah 'flutter pub get'
- c. Izinkan akses kamera pada AndroidManifest.xml

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
```



```
praktikum_9 > 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" />
```

a) Android

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

```
minSdkVersion 21
27 minSdkVersion 21
```

- Implementasi kamera pada halaman Flutter

```
import 'package:camera/camera.dart';
import 'package:flutter/material.dart';
import 'package:praktikum_9/display_screen.dart';

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

  @override
  State<MyCameraScreen> createState() => _MyCameraScreenState();
}

class MyCameraScreenState extends State<MyCameraScreen> {
```

```

late CameraController _controller;
Future<void>? _initializeControllerFuture;

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

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

  _initializeControllerFuture = _controller.initialize();
  setState(() {});
}

@override
void initState() {
  super.initState(); // super.initState() harus dipanggil pertama
  _initializeCamera();
}

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

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text("Camera Implementation"),
      centerTitle: true,
      backgroundColor: Colors.greenAccent,
    ),
    body: FutureBuilder(
      future: _initializeControllerFuture,
      builder: (context, snapshot) {
        if (snapshot.connectionState == ConnectionState.done) {
          return CameraPreview(_controller);
        } else {
          return const Center(
            child: CircularProgressIndicator(),
          );
        }
      },
    ),
    floatingActionButton: FloatingActionButton(
      onPressed: () async {

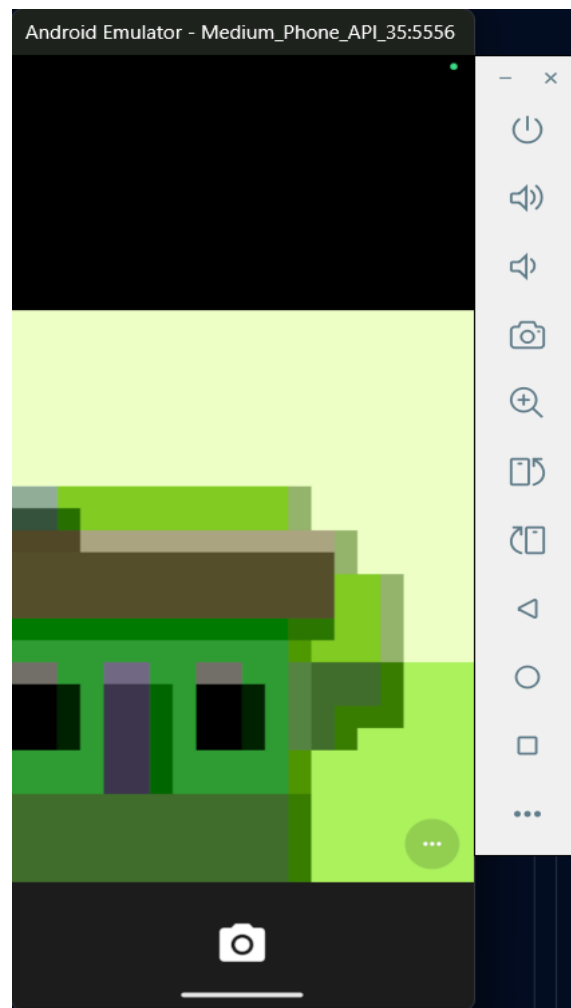
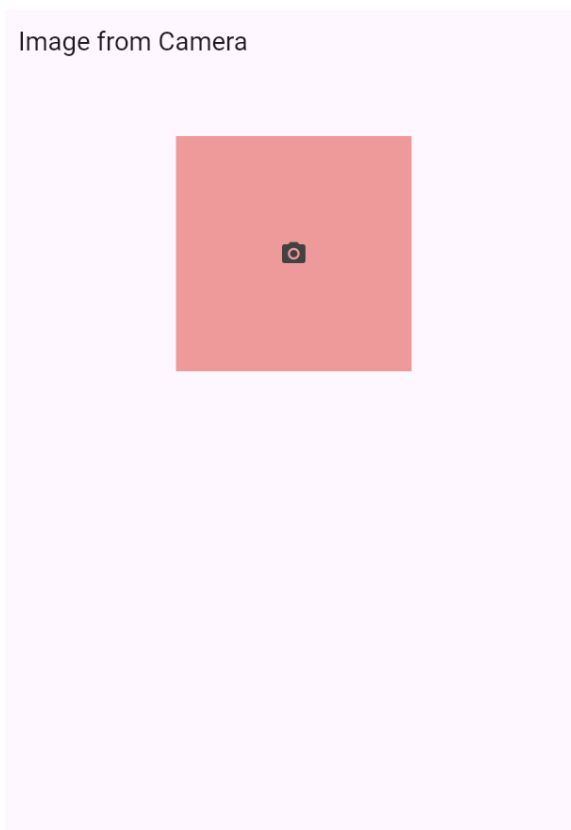
```

```

try {
  await _initializeControllerFuture;
  final image = await _controller.takePicture();
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (_) => DisplayScreen(
        imagePath: image.path,
      ),
    ),
  );
} catch (e) {
  print(e);
}
},
child: const Icon(Icons.camera),
),
);
}
}

```

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.

- Untuk penggunaan image picker, dapat dilihat pada potongan kode berikut :

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

class ImagePickerScreen extends StatefulWidget {
  final ImageSourceType type;

  ImagePickerScreen(this.type);

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

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

  ImagePickerScreenState(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(
            child: GestureDetector(
              onTap: () async {
```

```

// Determine whether to use camera or gallery
var source = type == ImageSourceType.camera
  ? ImageSource.camera
  : ImageSource.gallery;

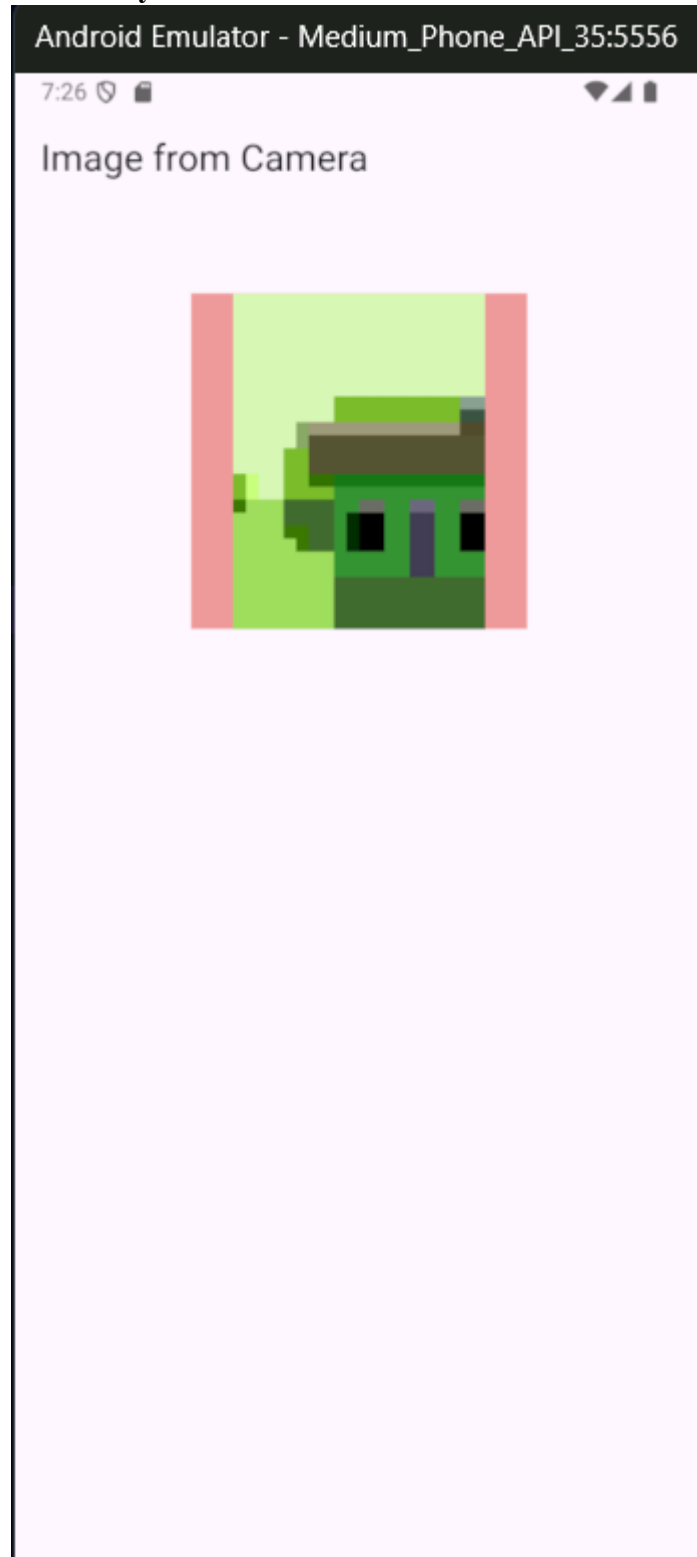
// Pick image from the selected source
XFile? image = await imagePicker.pickImage(
  source: source,
  imageQuality: 50,
  preferredCameraDevice: CameraDevice.front,
);

if (image != null) {
  setState(() {
    _image = File(image.path); // Update image file
  });
} else {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text('No image selected!')),
  );
}
},
child: Container(
  width: 200,
  height: 200,
  decoration: BoxDecoration(
    color: Colors.red[200],
  ),
  child: _image != null
    ? Image.file(
        _image!,
        width: 200.0,
        height: 200.0,
        fit: BoxFit.fitHeight,
      )
    : Icon(
        Icons.camera_alt,
        color: Colors.grey[800],
      ),
    ),
  ),
),
],
),
);
}
}

```

```
enum ImageSourceType { camera, gallery }
```

Out Putnya:



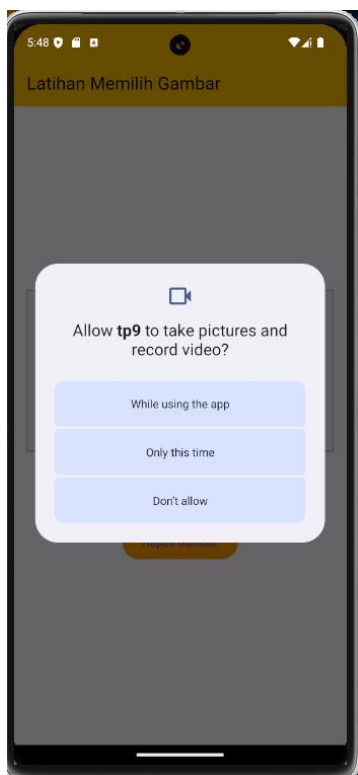
UNGUIDED

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.

OutPutnya:

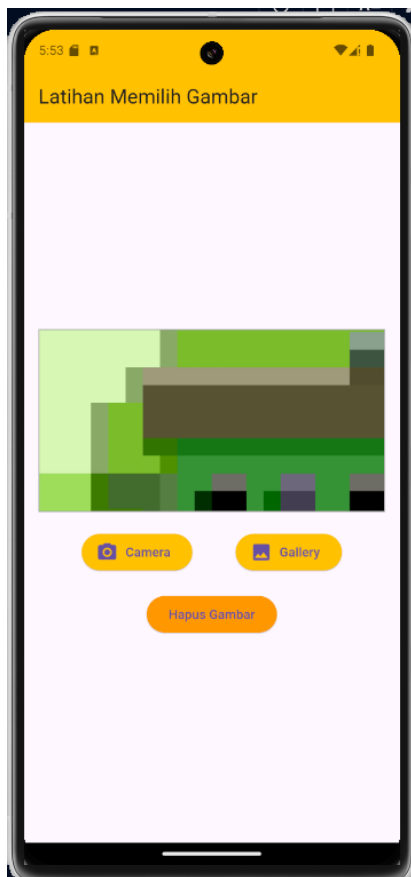
1. klik kamera



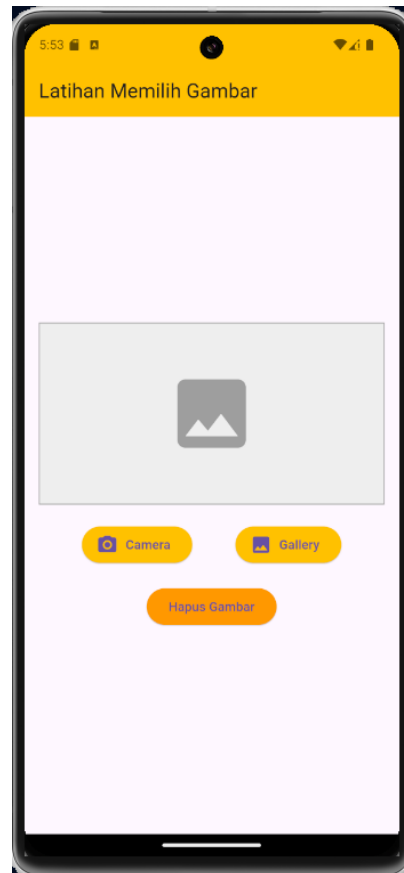
2. Masuk kamera dan ambil gambar



3.gambar tersimpan



4. Klik hapus gambar ketika gambar mau di hapus



Source Code

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: ImagePickerExample(),
    );
  }
}
```

```

class ImagePickerExample extends StatefulWidget {
  @override
  ImagePickerExampleState createState() => ImagePickerExampleState();
}

class ImagePickerExampleState extends State<ImagePickerExample> {
  File? _selectedImage;
  final ImagePicker _picker = ImagePicker();

  Future<void> _pickImageFromGallery() async {
    final pickedFile = await _picker.pickImage(source: ImageSource.gallery);
    if (pickedFile != null) {
      setState(() {
        _selectedImage = File(pickedFile.path);
      });
    }
  }

  Future<void> _pickImageFromCamera() async {
    final pickedFile = await _picker.pickImage(source: ImageSource.camera);
    if (pickedFile != null) {
      setState(() {
        _selectedImage = File(pickedFile.path);
      });
    }
  }

  void _clearImage() {
    setState(() {
      _selectedImage = null;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Latihan Memilih Gambar'),
        backgroundColor: Colors.amber,
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Container(
              height: 200,
              width: double.infinity,
              decoration: BoxDecoration(
                color: Colors.grey[200],

```

```

        border: Border.all(color: Colors.grey),
      ),
      child: _selectedImage != null
        ? Image.file(_selectedImage!, fit: BoxFit.cover)
        : const Center(
          child: Icon(
            Icons.image,
            size: 100,
            color: Colors.grey,
          ),
        ),
    ),
    const SizedBox(height: 20),
    Row(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: [
        ElevatedButton.icon(
          onPressed: _pickImageFromCamera,
          icon: const Icon(Icons.camera_alt),
          label: const Text('Camera'),
          style: ElevatedButton.styleFrom(
            backgroundColor: Colors.amber,
          ),
        ),
        ElevatedButton.icon(
          onPressed: _pickImageFromGallery,
          icon: const Icon(Icons.photo),
          label: const Text('Gallery'),
          style: ElevatedButton.styleFrom(
            backgroundColor: Colors.amber,
          ),
        ),
      ],
    ),
    const SizedBox(height: 20),
    ElevatedButton(
      onPressed: _clearImage,
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.orange,
      ),
      child: const Text('Hapus Gambar'),
    ),
  ],
),
);
}

```

Deskripsi Program

Program ini adalah aplikasi Flutter sederhana yang memungkinkan pengguna untuk memilih dan menampilkan gambar dari galeri atau kamera, serta menghapus gambar yang telah dipilih. Dengan memanfaatkan paket *image_picker*, pengguna dapat mengambil gambar langsung dari perangkat mereka dan menampilkannya dalam sebuah container. Aplikasi ini menyediakan tiga tombol utama: "Gallery" untuk memilih gambar, "Camera" untuk mengambil gambar, dan "Hapus Gambar" untuk menghapus gambar dari container tersebut. Antarmuka aplikasi dirancang dengan kesederhanaan, menampilkan ikon dan label yang jelas serta mudah dipahami. Fitur ini juga didukung oleh pengaturan izin pada perangkat Android untuk akses kamera dan galeri, sehingga aplikasi dapat berjalan dengan mulus. Aplikasi ini sangat cocok untuk latihan pengembangan atau sebagai fondasi bagi proyek-proyek berbasis media.