PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK

MODUL IX

API PERANGKAT KERAS

Disusun Oleh:

Aorinka Anendya Chazanah / 2211104013

S1 SE-06-01

Asisten Praktikum:

MuhammadFazaZulian 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 **camera** yang memudahkan pengembang untuk mengimplementasikan fitur kamera, seperti mengambil foto, merekam video, dan mengakses tampilan kamera secara langsung. Paket ini sangat berguna untuk pengembangan aplikasi yang memerlukan fitur pengambilan gambar atau video, seperti aplikasi media sosial atau e-commerce. Berikut adalah langkah instalasinya.

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

```yaml
dependencies:
  flutter:
    sdk: flutter

  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.8
  camera: ^0.11.0+2
  image_picker: ^1.1.2
```

b) Lalu jalankan perintah ‘flutter pub get’

c) Izinkan akses kamera pada AndroidManifest.xml

```xml
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
```

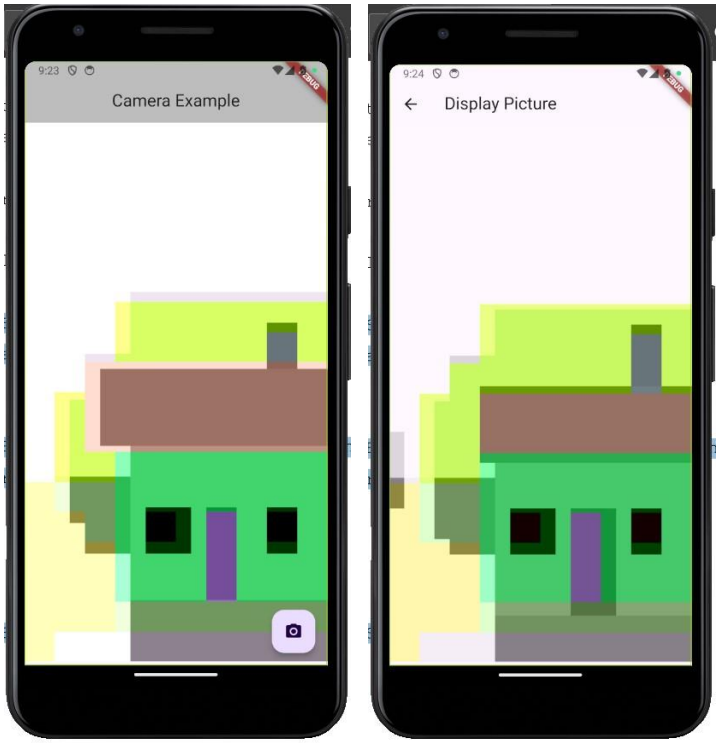
d) Implementasi kamera pada halaman Flutter

```dart
import 'package:camera/camera.dart';
import 'package:image_picker/image_picker.dart';
import 'package:flutter/material.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;
  Future<void>? _initializeControllerFuture;

  @override
  void initState() {
    super.initState();
    _initializeCamera();
```

```
  }

  Future<void> _initializeCamera() async {
    final cameras = await availableCameras();
    final firstCamera = cameras.first;
    _controller = CameraController(firstCamera,
ResolutionPreset.high);
    _initializeControllerFuture = _controller.initialize();
  }

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Camera Example'),
        centerTitle: true,
        backgroundColor: Colors.grey.shade400,
      ),
      body: FutureBuilder<void>(
        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: (context) =>
                    DisplayPictureScreen(imagePath: image.path),
              ),
            );
          } catch (e) {
            print(e);
          }
        },
        child: const Icon(Icons.camera_alt),
      ),
    );
  }
}

class DisplayPictureScreen extends StatelessWidget {
  final String imagePath;
```

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

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

Hasil Output



2. Media API

Media API merupakan kumpulan alat dan pustaka yang memungkinkan pengelolaan serta interaksi dengan berbagai jenis media, seperti gambar, video, dan audio. Meskipun Flutter tidak menyediakan API bawaan untuk semua kebutuhan media, pengembang dapat menggunakan paket tambahan untuk mengakses fitur media yang sering digunakan dalam aplikasi.

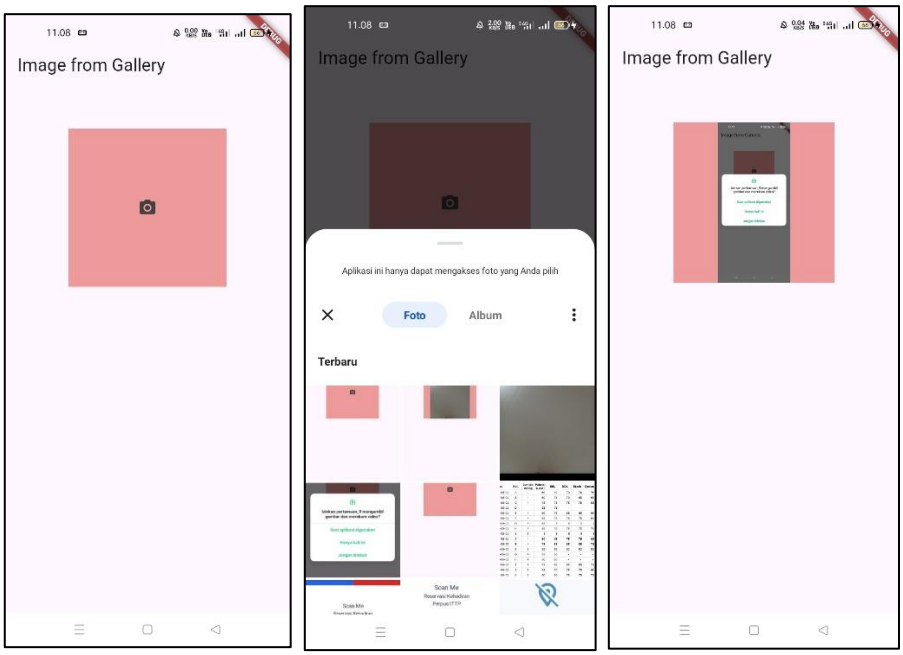
Dalam pembahasan ini, kita akan menggunakan paket Image Picker agar aplikasi dapat mengakses galeri media pada perangkat. Pada platform iOS, diperlukan konfigurasi tambahan, sementara pada Android tidak memerlukan konfigurasi khusus. Berikut adalah langkah-langkah konfigurasi tambahan untuk iOS. Implementasi pada Flutter:

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

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

Hasil 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, screenshoot output, dan deskripsi program. Kreatifitas menjadi nilai tambah.

Source Code

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

enum ImageSourceType { camera, gallery }

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  final cameras = await availableCameras();
  runApp(MyApp(cameras: cameras));
}

class MyApp extends StatelessWidget {
  final List<CameraDescription> cameras;

  const MyApp({Key? key, required this.cameras}) : super(key: key);

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

class ImagePickerExample extends StatefulWidget {
  final List<CameraDescription> cameras;

  const ImagePickerExample({Key? key, required this.cameras}) : super(key: key);

  @override
```

```dart
  _ImagePickerExampleState createState() => _ImagePickerExampleState();
}

class _ImagePickerExampleState extends State<ImagePickerExample> {
  File? _selectedImage;
  final ImagePicker _picker = ImagePicker();
  late CameraController _controller;
  late Future<void> _initializeControllerFuture;

  @override
  void initState() {
    super.initState();
    _controller = CameraController(
      widget.cameras.first,
      ResolutionPreset.medium,
    );
    _initializeControllerFuture = _controller.initialize();
  }

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

  Future<void> _handleImageSelection(ImageSourceType type) async {
    if (type == ImageSourceType.gallery) {
      final XFile? image = await _picker.pickImage(source: ImageSource.gallery);
      if (image != null) {
        setState(() {
          _selectedImage = File(image.path);
        });
      }
    } else {
      await Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => CameraScreen(
            cameras: widget.cameras,
            onImageCaptured: (File image) {
              setState(() {
                _selectedImage = image;
              });
            },
          ),
        ),
      );
    }
  }

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
```

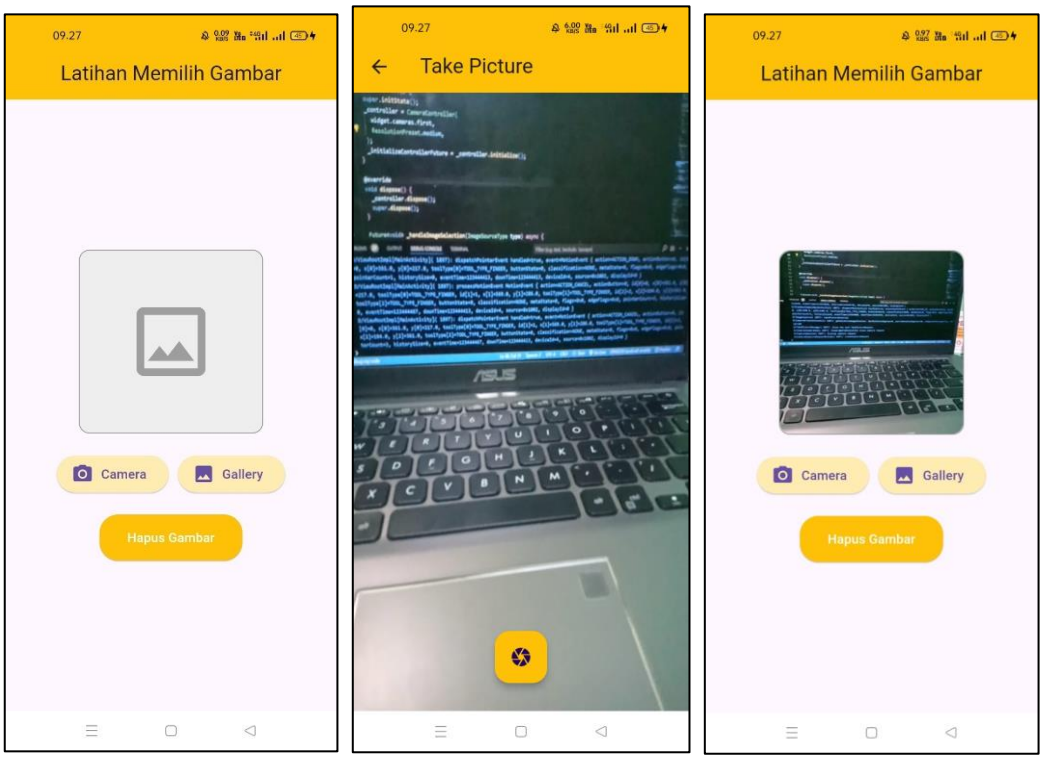
```
      appBar: AppBar(
        backgroundColor: Colors.amber,
        title: Text('Latihan Memilih Gambar'),
        centerTitle: true,
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Container(
              width: 200,
              height: 200,
              decoration: BoxDecoration(
                color: Colors.grey[200],
                borderRadius: BorderRadius.circular(12),
                border: Border.all(color: Colors.grey),
              ),
              child: _selectedImage != null
                  ? ClipRRect(
                      borderRadius: BorderRadius.circular(12),
                      child: Image.file(
                        _selectedImage!,
                        width: 200,
                        height: 200,
                        fit: BoxFit.cover,
                      ),
                    )
                  : Center(
                      child: Icon(
                        Icons.image_outlined,
                        size: 100,
                        color: Colors.grey,
                      ),
                    ),
            ),
            SizedBox(height: 20),
            Row(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                ElevatedButton.icon(
                  onPressed: () =>
                      _handleImageSelection(ImageSourceType.camera),
                  icon: Icon(Icons.camera_alt),
                  label: Text('Camera'),
                  style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.amber[100],
                    shape: RoundedRectangleBorder(
                      borderRadius: BorderRadius.circular(20),
                    ),
                  ),
                ),
                SizedBox(width: 10),
                ElevatedButton.icon(
                  onPressed: () =>
                      _handleImageSelection(ImageSourceType.gallery),
                  icon: Icon(Icons.photo),
                  label: Text('Gallery'),
                  style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.amber[100],
```

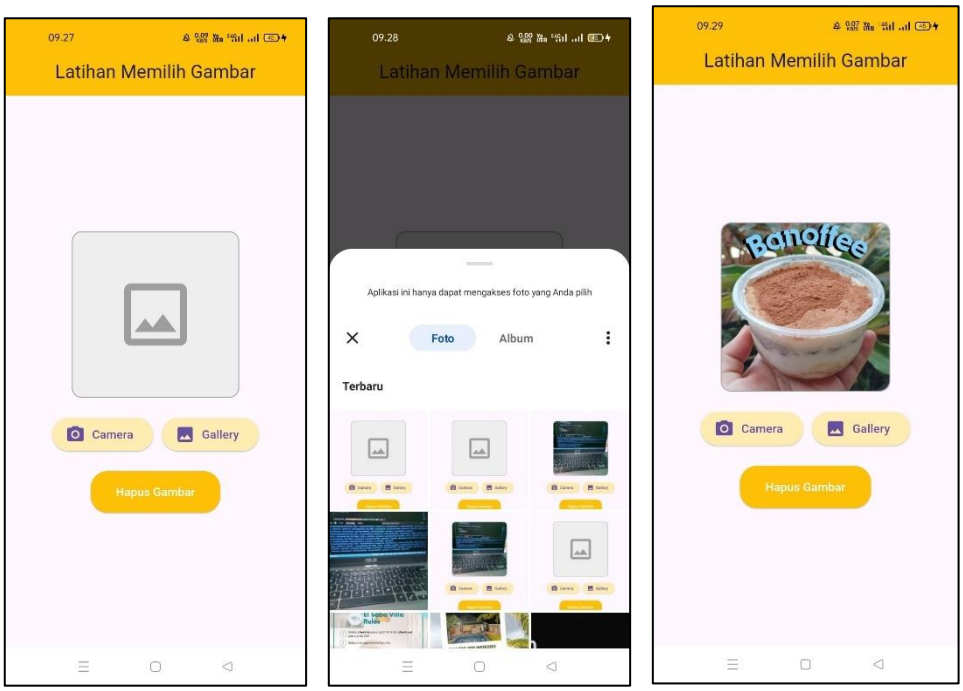
```
                    shape: RoundedRectangleBorder(
                      borderRadius: BorderRadius.circular(20),
                    ),
                  ),
                ),
              ],
            ),
            SizedBox(height: 20),
            ElevatedButton(
              onPressed: _deleteImage,
              child: Text(
                'Hapus Gambar',
                style: TextStyle(color: Colors.white),
              ),
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.amber,
                shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(20),
                ),
                padding: EdgeInsets.symmetric(horizontal: 30, vertical: 15),
              ),
            ),
          ],
        ),
      ),
    );
  }
}

class CameraScreen extends StatefulWidget {
  final List<CameraDescription> cameras;
  final Function(File) onImageCaptured;

  const CameraScreen({
    Key? key,
    required this.cameras,
    required this.onImageCaptured,
  }) : super(key: key);

  @override
  _CameraScreenState createState() => _CameraScreenState();
}

class _CameraScreenState extends State<CameraScreen> {
  late CameraController _controller;
  late Future<void> _initializeControllerFuture;

  @override
  void initState() {
    super.initState();
    _controller = CameraController(
      widget.cameras.first,
      ResolutionPreset.medium,
    );
    _initializeControllerFuture = _controller.initialize();
  }

  @override
  void dispose() {
```

```dart
    _controller.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.amber,
        title: Text('Take Picture'),
      ),
      body: FutureBuilder<void>(
        future: _initializeControllerFuture,
        builder: (context, snapshot) {
          if (snapshot.connectionState == ConnectionState.done) {
            return Stack(
              children: [
                Positioned.fill(
                  child: CameraPreview(_controller),
                ),
                Positioned(
                  bottom: 30,
                  left: 0,
                  right: 0,
                  child: Row(
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: [
                      FloatingActionButton(
                        backgroundColor: Colors.amber,
                        child: Icon(Icons.camera),
                        onPressed: () async {
                          try {
                            await _initializeControllerFuture;
                            final image = await _controller.takePicture();
                            widget.onImageCaptured(File(image.path));
                            Navigator.pop(context);
                          } catch (e) {
                            print(e);
                          }
                        },
                      ),
                    ],
                  ),
                ),
              ],
            );
          } else {
            return Center(child: CircularProgressIndicator());
          }
        },
      ),
    );
  }
}
```

Hasil Output

Gambar: Tampilan awal > halaman ambil gambar > gambar muncul pada container > tampilan setelah klik “Hapus Gambar” > tampilan galeri/album > memilih gambar dari galeri kemudian gambar muncul pada container.





Deskripsi Program

Aplikasi ini menggunakan package **image_picker** untuk mengambil gambar dari galeri dan **camera** untuk menangkap gambar menggunakan kamera perangkat. Aplikasi dimulai dengan menginisialisasi kamera yang tersedia, lalu menjalankan aplikasi yang menampilkan layar utama dengan tiga tombol: *Camera*, *Gallery*, dan *Hapus Gambar*. Gambar yang dipilih atau diambil akan ditampilkan di dalam sebuah kontainer, dengan opsi untuk menghapus gambar yang ditampilkan.

Fungsionalitas utama aplikasi mencakup:

- **Camera Button**: Membuka layar kamera untuk mengambil gambar menggunakan paket **camera**. Gambar yang diambil ditampilkan pada container layar utama.
- **Gallery Button**: Mengakses galeri perangkat menggunakan **image_picker** untuk memilih gambar.
- **Hapus Gambar Button**: Menghapus gambar yang sedang ditampilkan dari layar.