LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK

MODUL 9

API PERANGKAT KERAS

Disusun Oleh :

YOGI HAFIDH MAULANA / 2211104061

SE06-02

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

A. GUIDED

- **Camera API**

  Code

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

void main() {
  runApp(const MyApp());
}

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'E-Commerce App',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home:
          // MyCameraScreen()
          ImagePickerScreen(
        ImageSourceType.gallery,
      ),
    );
  }
}
```

```dart
import 'package:camera/camera.dart';
import 'package:flutter/material.dart';
import 'package:praktikum/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();
    _initializeCamera();
  }

  @override
  void dispose() {
    _controller.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),
      ),
    );
  }
}
```

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

class DisplayScreen extends StatelessWidget {
  final String imagePath;

  const DisplayScreen({
    super.key,
    required this.imagePath,
  });

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Display Screen'),
        centerTitle: true,
        backgroundColor: Colors.greenAccent,
        actions: [
          IconButton(
            icon: const Icon(Icons.share),
            onPressed: () {
            },
          ),
        ],
      ),
      body: Column(
        children: [
          Expanded(
            child: Image.file(
              File(imagePath),
              fit: BoxFit
                  .contain,
            ),
          ),
          Padding(
            padding: const EdgeInsets.all(16.0),
            child: Row(
              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
              children: [
                ElevatedButton(
                  onPressed: () {
                    Navigator.pop(context);
                  },
                  child: const Text('Take Another Photo'),
                ),
                ElevatedButton(
                  onPressed: () {
                  },
                  child: const Text('Save Photo'),
                ),
              ],
            ),
          ),
        ],
      ),
    );
  }
}
```

Output

Deskripsi Program

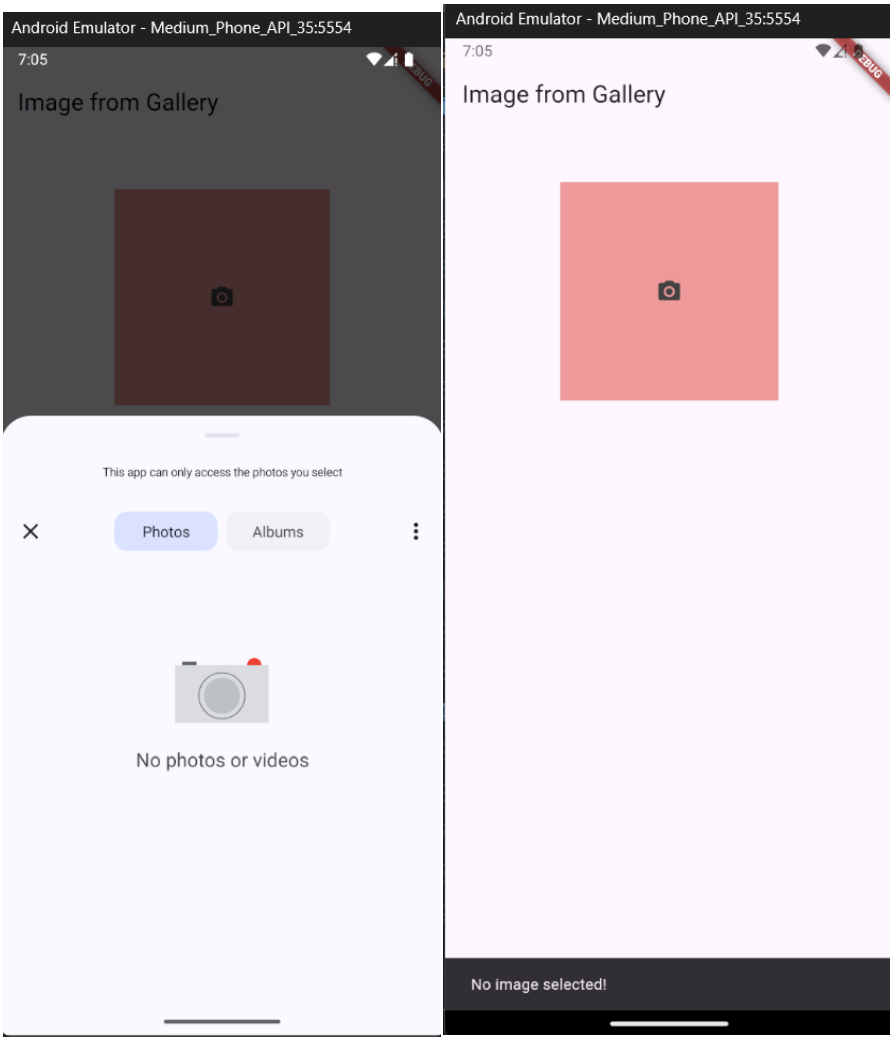
Kode di atas adalah implementasi dari fitur Camera API pada Flutter untuk menangkap, menampilkan, dan menyimpan foto. Kode terdiri dari tiga bagian utama: MyApp, MyCameraScreen, dan DisplayScreen. Pada bagian MyApp, aplikasi memulai dengan tampilan ImagePickerScreen, yang memungkinkan pengguna memilih gambar dari galeri. Pada MyCameraScreen, terdapat inisialisasi kamera menggunakan CameraController, di mana kamera pertama dari daftar perangkat diatur untuk menangkap gambar dengan resolusi tinggi. Saat tombol kamera ditekan, kamera mengambil gambar dan menavigasi ke DisplayScreen. Di DisplayScreen, gambar yang diambil ditampilkan dan menyediakan opsi untuk kembali mengambil foto atau menyimpan gambar. Kode ini menggunakan FutureBuilder untuk memastikan kamera telah diinisialisasi sebelum digunakan, dan CameraPreview untuk menampilkan pratinjau kamera.

- **Media API**

  Code:

```dart
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 {
                var source = type == ImageSourceType.camera
                    ? ImageSource.camera
                    : ImageSource.gallery;

                XFile? image = await imagePicker.pickImage(
                  source: source,
                  imageQuality: 50,
                  preferredCameraDevice: CameraDevice.front,
                );

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

Output



Deskripsi Program

Kode di atas adalah contoh penggunaan Media API untuk memilih gambar baik dari kamera atau galeri pada aplikasi Flutter. ImagePickerScreen adalah halaman yang memungkinkan pengguna memilih gambar dengan menggunakan kamera atau galeri berdasarkan pilihan yang diberikan pada parameter type. Dalam ImagePickerScreenState, saat pengguna mengetuk area gambar (gesture detector), aplikasi akan membuka kamera atau galeri sesuai dengan sumber yang dipilih, lalu mengambil gambar menggunakan ImagePicker. Gambar yang diambil akan ditampilkan dalam kontainer dengan ukuran 200x200 piksel. Jika pengguna tidak memilih gambar, akan muncul pesan pemberitahuan melalui SnackBar. Kode ini memanfaatkan Media API yang ada di Flutter untuk menangani pengambilan gambar dari perangkat pengguna.

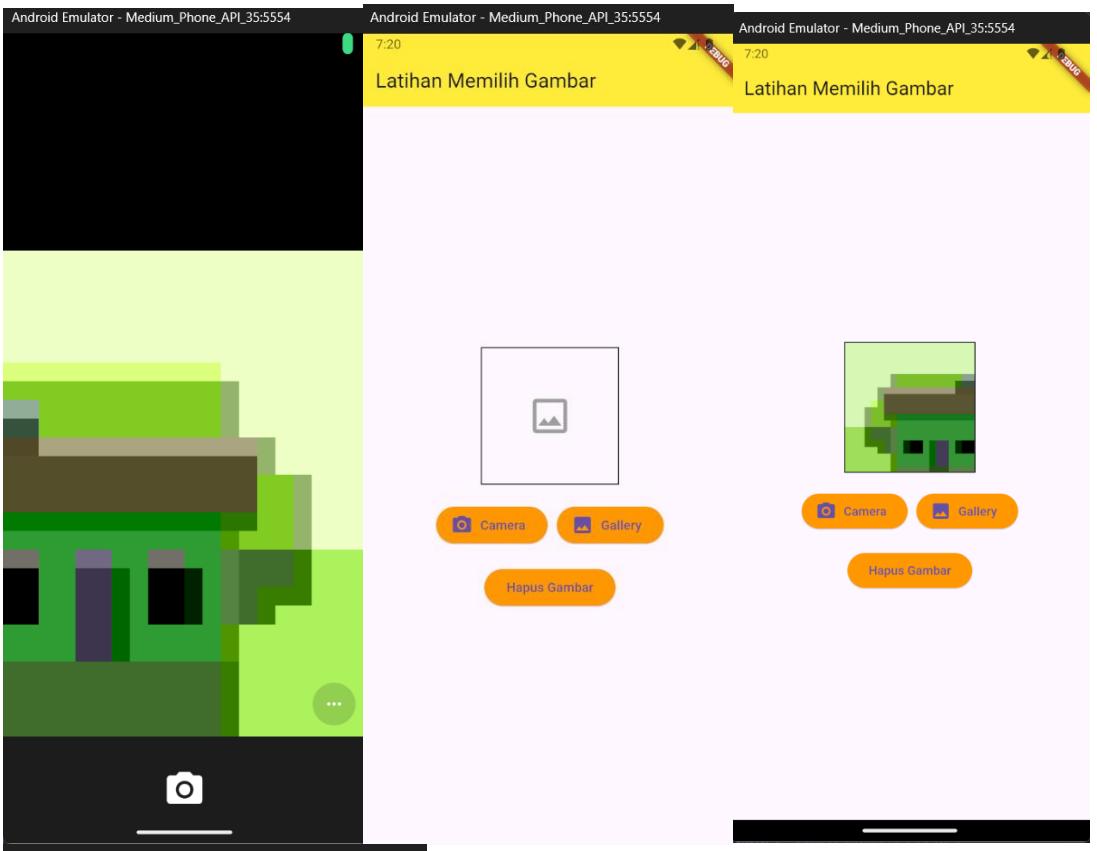
B. Unguided

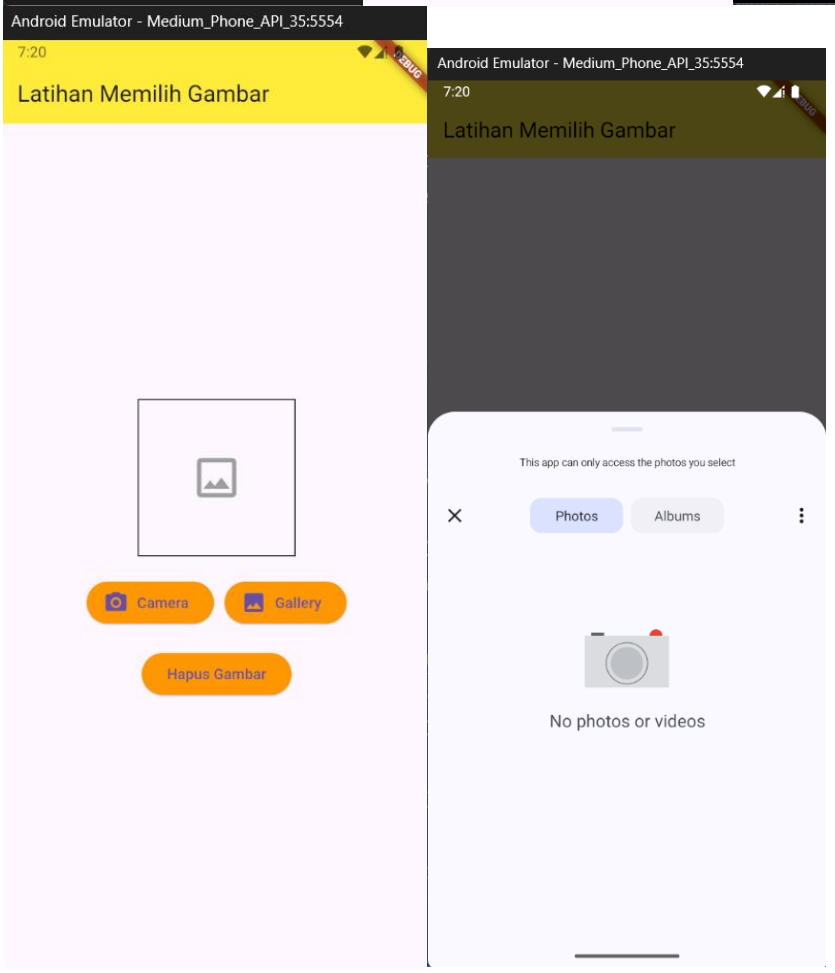
1. Soal 1

Code

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

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Latihan Memilih Gambar'),
          backgroundColor: Colors.yellow,
        ),
        body: const ImagePickerPage(),
      ),
    );
  }
}

class ImagePickerPage extends StatefulWidget {
  const ImagePickerPage({Key? key}) : super(key: key);

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

class _ImagePickerPageState extends State<ImagePickerPage> {
  File? _image;

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

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

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

  @override
  Widget build(BuildContext context) {
    return Center(
      child: Padding(
        padding: const EdgeInsets.all(20.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Container(
              width: 150,
              height: 150,
              decoration: BoxDecoration(
                border: Border.all(color: Colors.black, width: 1),
              ),
              child: _image == null
                  ? const Icon(
                      Icons.image_outlined,
                      size: 50,
                      color: Colors.grey,
                    )
                  : Image.file(
                      _image!,
                      fit: BoxFit.cover,
                    ),
            ),
            const SizedBox(height: 20),
            Row(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                ElevatedButton.icon(
                  onPressed: _pickImageFromCamera,
                  icon: const Icon(Icons.camera_alt),
                  label: const Text('Camera'),
                  style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.orange,
                  ),
                ),
                const SizedBox(width: 10),
                ElevatedButton.icon(
                  onPressed: _pickImageFromGallery,
                  icon: const Icon(Icons.image),
                  label: const Text('Gallery'),
                  style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.orange,
                  ),
                ),
              ],
            ),
            const SizedBox(height: 20),
            ElevatedButton(
              onPressed: _removeImage,
              child: const Text('Hapus Gambar'),
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.orange,
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

Output


Android Emulator - Medium_Phone_API_35:5554
Android Emulator - Medium_Phone_API_35:5554
7:20
Latihan Memilih Gambar
Camera
Gallery
Hapus Gambar
Android Emulator - Medium_Phone_API_35:5554
7:20
Latihan Memilih Gambar
Camera
Gallery
Hapus Gambar


Android Emulator - Medium_Phone_API_35:5554
7:20
Latihan Memilih Gambar
Camera
Gallery
Hapus Gambar
Android Emulator - Medium_Phone_API_35:5554
7:20
Latihan Memilih Gambar
This app can only access the photos you select
Photos
Albums
No photos or videos

Deskripsi Program

Kode di atas menunjukkan penerapan penggunaan Camera API dan Media API dalam aplikasi Flutter yang memungkinkan pengguna untuk memilih gambar dari kamera atau galeri. Aplikasi dimulai dengan widget MyApp yang menggunakan MaterialApp untuk menampilkan tampilan aplikasi. Pada halaman utama ImagePickerPage, terdapat tiga fitur utama: memilih gambar dari kamera, memilih gambar dari galeri, dan menghapus gambar yang telah dipilih.

Pada bagian _ImagePickerPageState, terdapat dua metode utama untuk mengambil gambar:

1. _pickImageFromGallery(): Menggunakan ImagePicker untuk membuka galeri perangkat dan memungkinkan pengguna memilih gambar dari galeri dengan cara memanggil pickImage() dan menyetel sumbernya ke ImageSource.gallery. Setelah gambar dipilih, gambar tersebut akan disimpan dalam variabel _image berupa objek File, lalu ditampilkan di UI.

2. _pickImageFromCamera(): Menggunakan ImagePicker dengan sumber ImageSource.camera untuk membuka kamera dan menangkap gambar. Gambar yang diambil disimpan dalam variabel _image dan ditampilkan di layar.

Fitur menghapus gambar diimplementasikan melalui metode _removeImage(), yang akan mengatur variabel _image menjadi null, sehingga gambar yang ditampilkan di UI akan hilang.

Di dalam widget build(), terdapat tampilan UI yang terdiri dari:

- Sebuah kotak dengan ukuran 150x150 piksel yang menampilkan gambar yang telah dipilih menggunakan Image.file() jika gambar sudah dipilih, atau menampilkan ikon kamera jika belum ada gambar yang dipilih.
- Dua tombol untuk memilih gambar, masing-masing dengan ikon untuk kamera Icons.camera_alt dan galeri Icons.image. Kedua tombol ini memanggil fungsi masing-masing _pickImageFromCamera dan _pickImageFromGallery ketika ditekan.
- Sebuah tombol Hapus Gambar yang akan memanggil metode _removeImage() untuk menghapus gambar yang ditampilkan.

Aplikasi ini mengintegrasikan penggunaan Camera API dan Media API dari Flutter, memberikan interaksi yang mudah untuk memilih gambar dari galeri atau kamera, serta memberikan fitur tambahan untuk menghapus gambar yang sudah dipilih.