LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK

MODUL IX
API PERANGKAT KERAS

Disusun Oleh :
Satria Ariq Adelard
Dompas/2211104033SE 06 2

Asisten Praktikum :
Muhammad Faza Zulian Gesit Al Barru
Aisyah Hasna Aulia

Dosen Pengampu :
Yudha Islami Sulistya

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2024

1. **GUIDED**
    a. **Camera API**

Source Code

- main.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,
      ),
    );
  }
}
```

- camera_screen.dart

```
import 'package:camera/camera.dart';
import 'package:flutter/material.dart';
import 'package:pertemuan9/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 {
```

```dart
  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(); // Perbaikan 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),
      ),
    );
  }
}
```

- display_screen.dart

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

class DisplayScreen extends StatelessWidget {
  final String imagePath; // Perbaikan nama variabel untuk konsistensi

  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(
            // Menambahkan tombol untuk berbagi gambar
            icon: const Icon(Icons.share),
            onPressed: () {
              // Implementasi fungsi berbagi bisa ditambahkan di sini
            },
          ),
        ],
      ),
      body: Column(
        children: [
          Expanded(
            child: Image.file(
              File(imagePath),
              fit: BoxFit
                  .contain, // Menambahkan BoxFit untuk tampilan yang lebih
baik
            ),
          ),
          Padding(
            padding: const EdgeInsets.all(16.0),
            child: Row(
              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
```

```
              children: [
                ElevatedButton(
                  onPressed: () {
                    Navigator.pop(context); // Kembali ke halaman kamera
                  },
                  child: const Text('Take Another Photo'),
                ),
                ElevatedButton(
                  onPressed: () {
                    // Implementasi fungsi simpan bisa ditambahkan di sini
                  },
                  child: const Text('Save Photo'),
                ),
              ],
            ),
          ),
        ],
      ),
    );
  }
}
```

b. Media API

Source Code

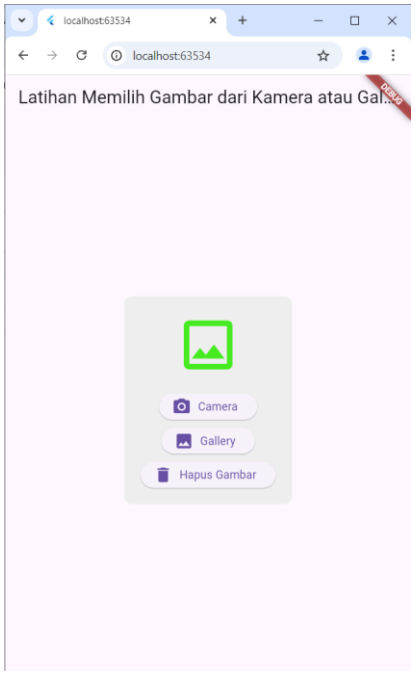
- image_picker.dart

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

class ImagePickerScreen extends StatefulWidget {
  final ImageSourceType type;

  const ImagePickerScreen(this.type, {super.key});

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

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

  ImagePickerScreenState(this.type);

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

```dart
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(type == ImageSourceType.camera
          ? "Image from Camera"
          : "Image from Gallery"),
    ),
    body: Column(
      children: <Widget>[
        const 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(
                  const 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



2. UNGUIDED

a. Soal 1

Source Code

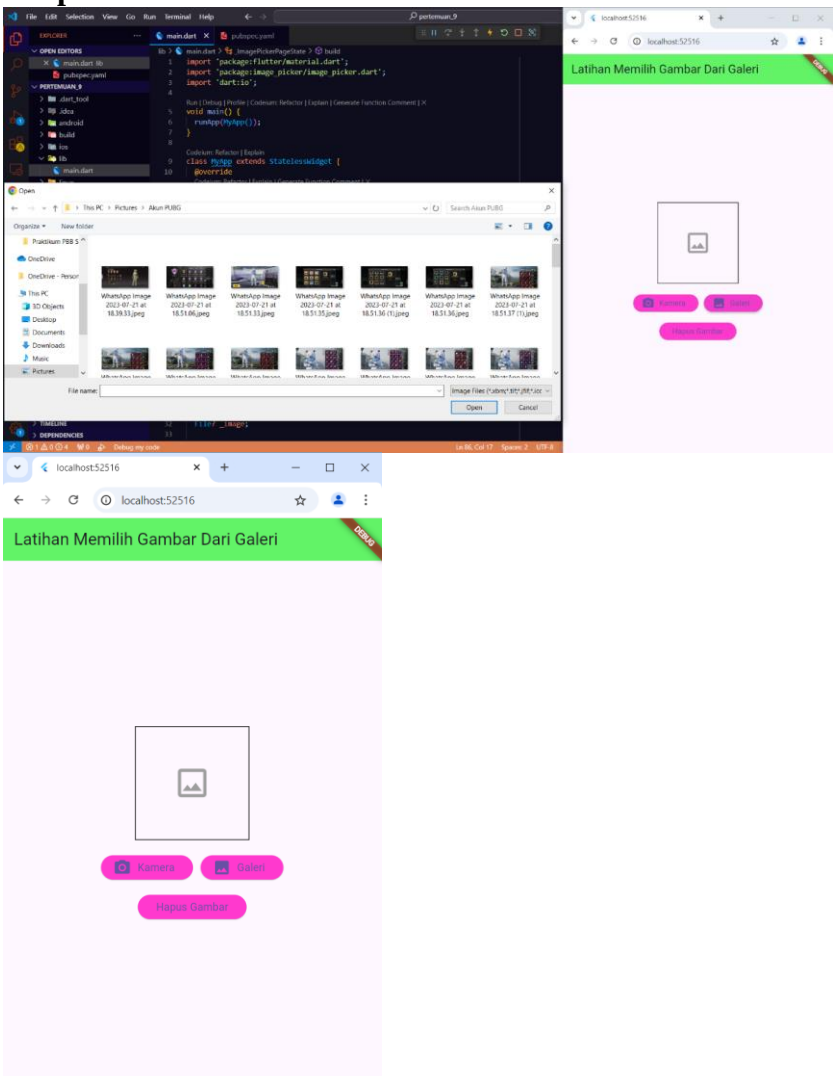
- main.dart

```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 Dari Galeri'),
          backgroundColor: const Color.fromARGB(255, 98, 243, 103),
        ),
        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('Kamera'),
                  style: ElevatedButton.styleFrom(
                    backgroundColor: const Color.fromARGB(255, 255, 57, 206),
                  ),
                ),
                const SizedBox(width: 10),
                ElevatedButton.icon(
                  onPressed: _pickImageFromGallery,
                  icon: const Icon(Icons.image),
                  label: const Text('Galeri'),
                  style: ElevatedButton.styleFrom(
```

```
                    backgroundColor: const Color.fromARGB(255, 255, 57, 206),
                  ),
                ),
              ],
            ),
            const SizedBox(height: 20),
            ElevatedButton(
              onPressed: _removeImage,
              child: const Text('Hapus Gambar'),
              style: ElevatedButton.styleFrom(
                backgroundColor: const Color.fromARGB(255, 255, 57, 206),
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

Output



Deskripsi

Kodingan diatas adalah implementasi aplikasi Flutter sederhana yang memungkinkan pengguna untuk memilih gambar dari galeri atau mengambil gambar menggunakan kamera, serta menghapus gambar yang telah dipilih. Aplikasi ini menggunakan package image_picker untuk menangani pemilihan gambar dari sumber yang berbeda. Gambar yang dipilih akan ditampilkan dalam sebuah Container, dan jika tidak ada gambar, ikon placeholder akan ditampilkan. Tombol-tombol yang tersedia diberi fungsi berbeda, yaitu memilih gambar dari kamera, memilih dari galeri, dan menghapus gambar, dengan masing-masing tombol memiliki desain dan warna khusus.