LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK

MODUL IX

API PERANGKAT KERAS

Disusun Oleh :

Muhammad Abdul Aziz / 2211104026

SE0601

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
File main.dart :

```dart
import 'package:flutter/material.dart';
import 'package:praktikum/image_picker_screen.dart';
import 'package:praktikum/myapi_page.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(ImageSourceType.camera),
      // MyApiPage(),
    );
  }
}
```

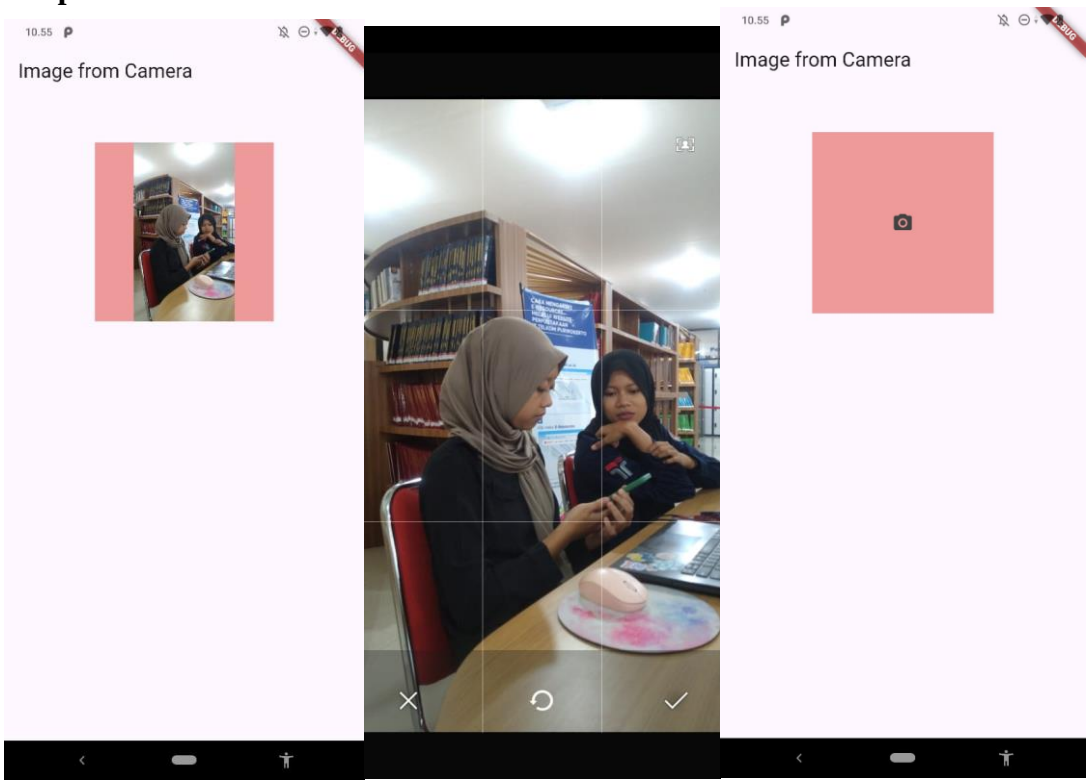
File image_picker_screen.dart :

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

enum ImageSourceType { camera, gallery }
```

File myapi_page.dart :

```dart
import 'dart:io';

import 'package:camera/camera.dart';
import 'package:flutter/material.dart';

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

  @override
  State<MyApiPage> createState() => _MyApiPageState();
}

class _MyApiPageState extends State<MyApiPage> {
  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() {
    _initializeCamera();
    super.initState();
  }

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('API Perangkat Keras'),
        centerTitle: true,
      ),
      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),
                ));
            await _controller.takePicture();
          } catch (e) {
            print(e);
          }
        },
        child: Icon(
          Icons.camera_alt_rounded,
          size: 30,
        ),
      ),
    );
  }
}

class DisplayPictureScreen extends StatelessWidget {
  final String imagePath;

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Display Foto'),
        centerTitle: true,
        backgroundColor: Colors.grey,
      ),
      body: Image.file(File(imagePath)),
    );
  }
}

```

Output :



Pnejelasan :

Aplikasi ini dikembangkan menggunakan Flutter dan menyediakan dua fitur utama yang berkaitan dengan kamera perangkat. Fitur pertama menggunakan plugin `image_picker` untuk mengambil gambar baik dari kamera maupun galeri. Pada fitur ini, pengguna dapat memilih sumber gambar, dan gambar yang diambil akan ditampilkan di layar aplikasi. Fitur ini diimplementasikan melalui widget `ImageFromGalleryEx`. Fitur kedua menggunakan plugin `camera` untuk memberikan kontrol langsung terhadap perangkat keras kamera. Melalui fitur ini, pengguna dapat melihat pratinjau langsung kamera pada perangkat dan menangkap gambar, yang kemudian ditampilkan melalui layar hasil menggunakan widget `DisplayPictureScreen`. Proses ini diimplementasikan dalam widget `MyApiPage`. Secara keseluruhan, aplikasi ini memanfaatkan kemampuan Flutter untuk berinteraksi dengan perangkat keras secara optimal melalui dua plugin utama, yaitu `image_picker` dan `camera`.

UNGUIDED
File main.dart :

```dart
import 'package:flutter/material.dart';
import 'package:unguided/image_picker_screen.dart';
import 'package:unguided/myapi_page.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(),
    );
  }
}
```

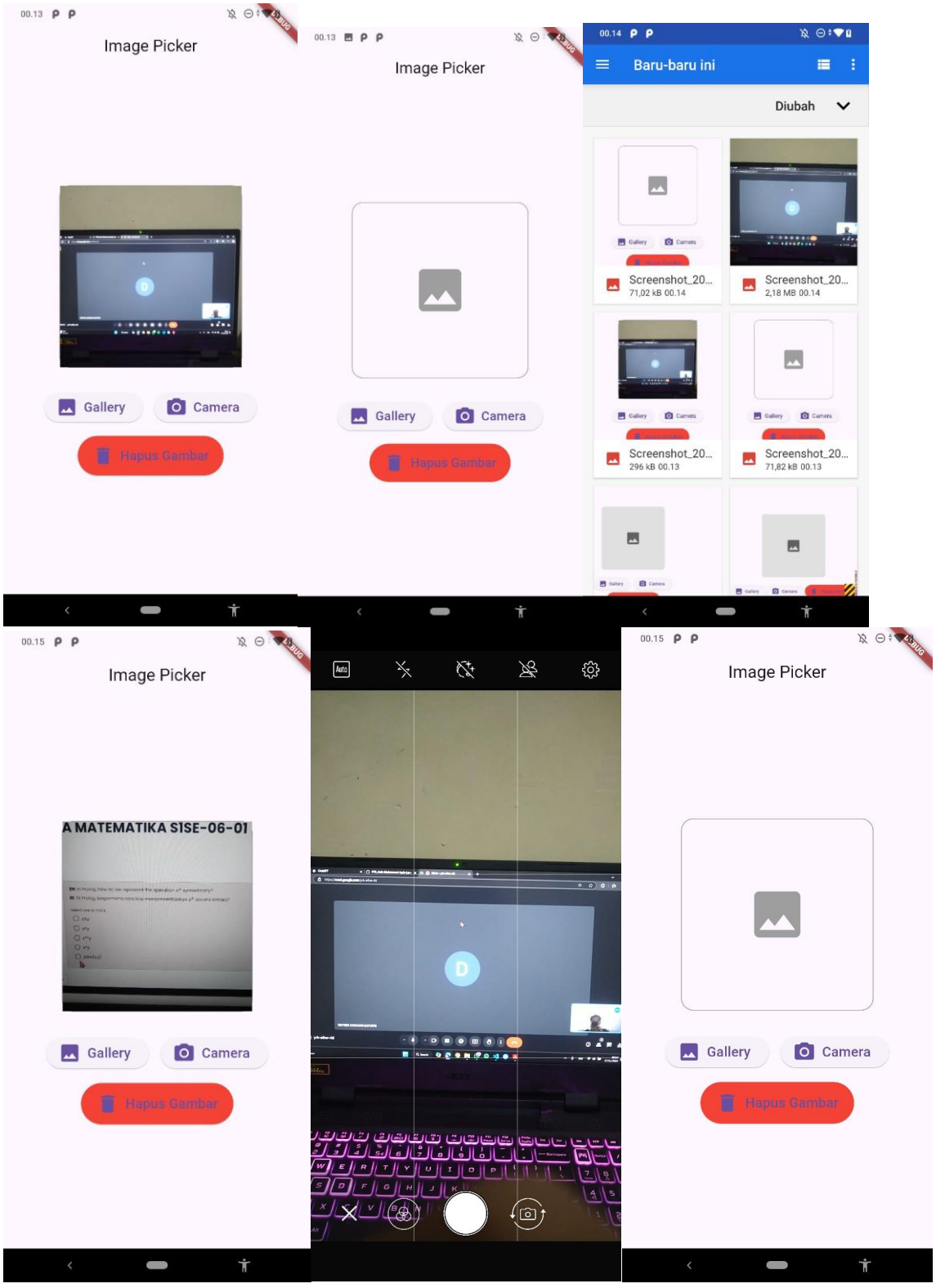
File image_picker_screen.dart :

```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) {
      print("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,
      ),
      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
                    ? Image.file(_image!, fit: BoxFit.cover)
                    : const Icon(
                        Icons.image,
                        size: 80,
                        color: Colors.grey,
                      ),
              ),

              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),
                    label: const Text(
                      "Gallery",
                      style: TextStyle(fontSize: 18),
                    ),
                  ),
                  ElevatedButton.icon(
                    onPressed: () => _pickImage(ImageSource.camera),
                    icon: const Icon(Icons.camera_alt, size: 30),
                    label: const Text(
                      "Camera",
                      style: TextStyle(fontSize: 18),
                    ),
                  ),
                  ElevatedButton.icon(
                    onPressed: _removeImage,
                    icon: const Icon(Icons.delete, size: 30),
                    label: const Text(
                      "Hapus Gambar",
                      style: TextStyle(fontSize: 18),
                    ),
                    style: ElevatedButton.styleFrom(
                      backgroundColor: Colors.red,
                      padding: const EdgeInsets.symmetric(
                        vertical: 12,
                        horizontal: 20,
                      ),
                    ),
                  ),
                ],
              ),
            ],
          ),
        ),
      );
    }
}
```

File myapi_page.dart :

```dart
import 'dart:io';

import 'package:camera/camera.dart';
import 'package:flutter/material.dart';

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

  @override
  State<MyApiPage> createState() => _MyApiPageState();
}

class _MyApiPageState extends State<MyApiPage> {
  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() {
    _initializeCamera();
    super.initState();
  }

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('API Perangkat Keras'),
        centerTitle: true,
      ),
      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),
                ));
            await _controller.takePicture();
          } catch (e) {
            print(e);
          }
        },
        child: Icon(
          Icons.camera_alt_rounded,
          size: 30,
        ),
      ),
    );
  }
}

class DisplayPictureScreen extends StatelessWidget {
  final String imagePath;

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Display Foto'),
        centerTitle: true,
        backgroundColor: Colors.grey,
      ),
      body: Image.file(File(imagePath)),
    );
  }
}

```

Output :



Penjelasan :

Aplikasi Flutter ini memiliki dua fitur utama terkait pengambilan gambar. Fitur pertama, menggunakan plugin `image_picker`, memungkinkan pengguna memilih gambar dari galeri atau kamera dan menampilkan hasilnya di layar. Fitur ini dilengkapi tombol hapus untuk menghilangkan gambar yang dipilih. Fitur kedua, menggunakan plugin `camera`, memberikan kontrol penuh terhadap kamera perangkat, termasuk pratinjau langsung dengan `CameraPreview` dan menangkap gambar yang hasilnya ditampilkan di halaman lain. Kode diorganisasi secara modular untuk memastikan kemudahan pengembangan dan pengalaman pengguna yang baik.