LAPORAN PRAKTIKUM
Modul 9
“API Perangkat Keras”

Disusun Oleh:
Muhammad Ralfi - 2211104054
Kelas SE-06-2

Dosen :
Yudha Islami Sulistya, S.Kom., M.Cs.

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO

2024

1. **Tujuan**
    a. Mahasiswa mampu memahami konsep layout pada Flutter
    b. Mahasiswa dapat mengimplementasikan desain user interface pada Flutter
2. **Landasan Teori**

Kamera API

Banyak aplikasi yang mengharuskan penggunaan kamera perangkat untuk mengambil foto dan video. Flutter menyediakan camera plugin untuk tujuan ini. cameraPlugin ini menyediakan alat untuk mendapatkan daftar kamera yang tersedia, menampilkan pratinjau yang berasal dari kamera tertentu, dan mengambil foto atau video. Untuk penggunaan nya perlu adanya instalasi plugin terlebih dahulu, berikut adalah cara untuk menginstall plugin Camera pada flutter:

a. Installasi Pada Android
    - Tambahkan paket camera yang ada pada Pub Dev kedalam file pubspec.yml
    - Jalankan perintah *'flutter pub get'*
    - Izinkan akses kamera pada AndroidManifest.xml

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

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

```
minSdkVersion 21
```

b. Installasi Pada IOS

Plugin ini dapat berjalan di versi iOS 10.0 atau lebih tinggi. Jika dijalankan di versi di bawah 10.0, pastikan bahwa ada program untuk cek versi dari iOS sebelum menggunakan fitur yang ada pada kamera. Plugin untuk cek versi iOS yaitu device_info_plus, tambahkan 2 baris pada ios/Runner/info/plist:

- Satu baris dengan Privacy - Camera Usage Description.
- Dan satu baris dengan Privacy - Microphone Usage Description

Atau dengan format teks penambahan key sebagai berikut:

```
<key>NSCameraUsageDescription</key>
<string>Can I use the camera please?</string>
<key>NSMicrophoneUsageDescription</key>
<string>Can I use the mic please?</string>
```

Media API

Media API adalah kumpulan alat dan pustaka yang berfungsi untuk mengelola serta berinteraksi dengan berbagai jenis media, termasuk gambar, video, dan audio. Meskipun Flutter tidak menyediakan API media bawaan yang mencakup semua kebutuhan terkait media, Anda dapat memanfaatkan paket tambahan untuk mengakses fitur-fitur media yang sering digunakan dalam aplikasi.

3. Guided

a. Implementasi Camera Plugin

1. Konfigurasi paket plugin sesuai dengan contoh sebelumnya pada landasan teori.
2. File Main.dart

```
import 'package:flutter/material.dart';
import 'package:prak9/cam.dart';

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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor:
Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const camera_screen(),
    );
  }
}
```

3. File camerascreen.dart

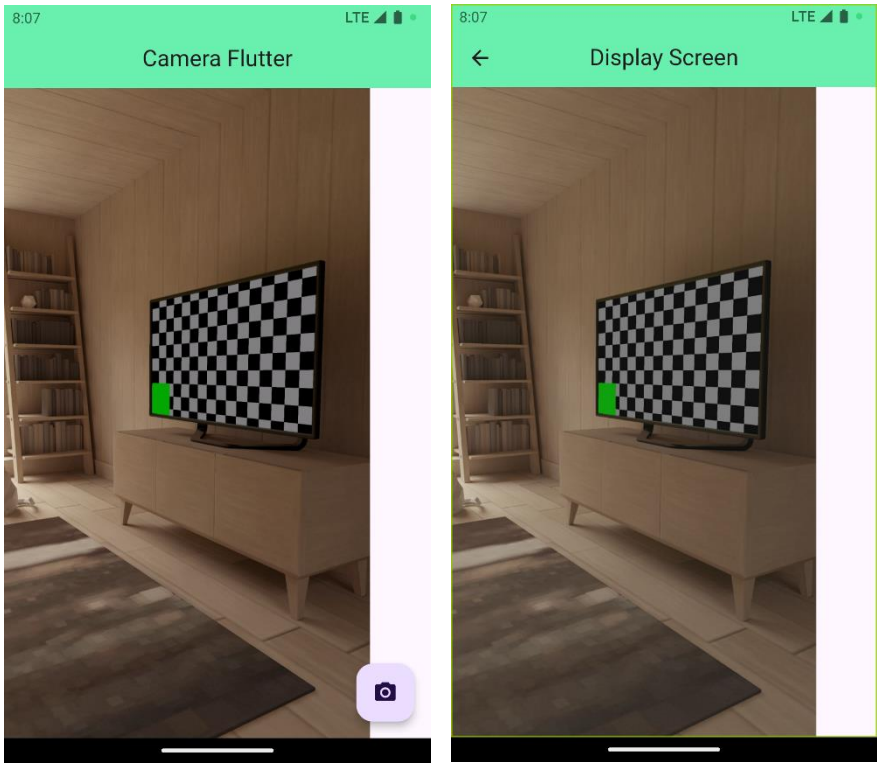
```
import 'package:camera/camera.dart';
import 'package:flutter/material.dart';
import 'package:prak9/display_screen.dart';

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

  @override
  State<camera_screen> createState() => _camera_screenState();
}

class _camera_screenState extends State<camera_screen> {
  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('Camera Flutter'),
        centerTitle: true,
        backgroundColor: Colors.greenAccent,
      ),
      body: FutureBuilder(
          future: _initializeControllerFuture,
          builder: (context, snapshot) {
            if (snapshot.connectionState == ConnectionState.done) {
              return CameraPreview(_controller);
            } else {
              return 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: Icon(Icons.camera_alt),
      ),
    );
  }
}
```

4. File screendisplay.dart

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

class DisplayScreen extends StatelessWidget {
  // const DisplayScreen({super.key});
  final String imagePath;

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

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

Output:



b. Implementasi Media API Menggunakan Image Picker

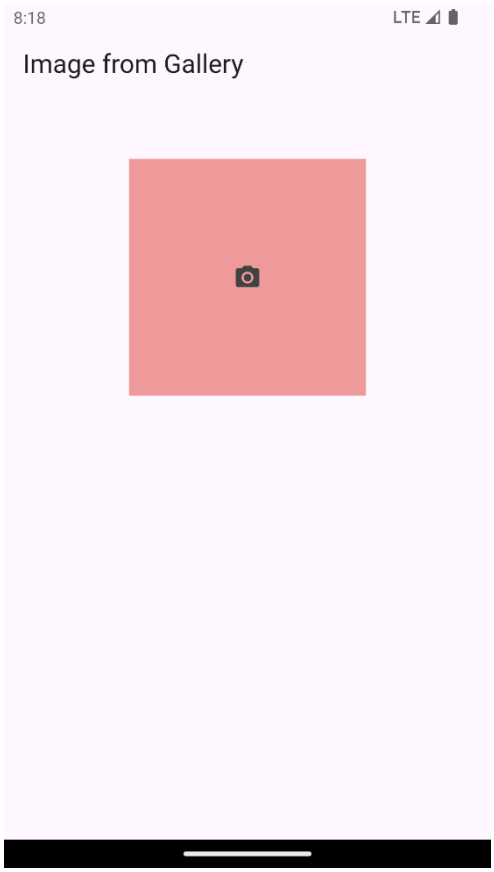
Untuk dapat menggunakan image picker perlu instalasi paket image_picker dari pub dev kedalam file pubspec.yml terlebih dahulu.

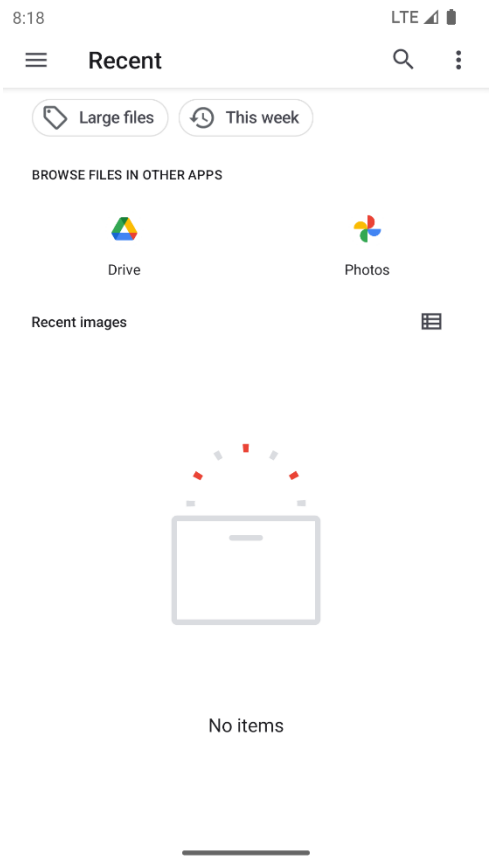
1. File imagepicker.dart

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


class ImageScreen extends StatefulWidget {
  final ImageSourceType type;

  ImageScreen(this.type);

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

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

  ImageScreenState(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);
```

```
                  });
                }
              },
              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,
                      )
                    : Container(
                        decoration: BoxDecoration(
                          color: Colors.red[200],
                        ),
                        width: 200,
                        height: 200,
                        child: Icon(
                          Icons.camera_alt,
                          color: Colors.grey[800],
                        ),
                      ),
              ),
            ),
          ),
        ],
      ),
    );
  }
}
enum ImageSourceType { camera, gallery }
```

Output:





4. **Unguided**

(Soal) Modifikasi project pemilihan gambar yang telah dikerjakan pada Tugas Pendahuluan Modul 09 agar fungsionalitas tombol dapat berfungsi untuk mengunggah gambar dengan ketentuan sebagai berikut:

a. Ketika tombol Gallery ditekan, aplikasi akan mengambil gambar dari galeri, dan setelah gambar dipilih, gambar tersebut akan ditampilkan di dalam container.
b. Ketika tombol Camera ditekan, aplikasi akan mengambil gambar menggunakan kamera, dan setelah pengambilan gambar selesai, gambar tersebut akan ditampilkan di dalam container.
c. Ketika tombol Hapus Gambar ditekan, gambar yang ada pada container akan dihapus.

Code:

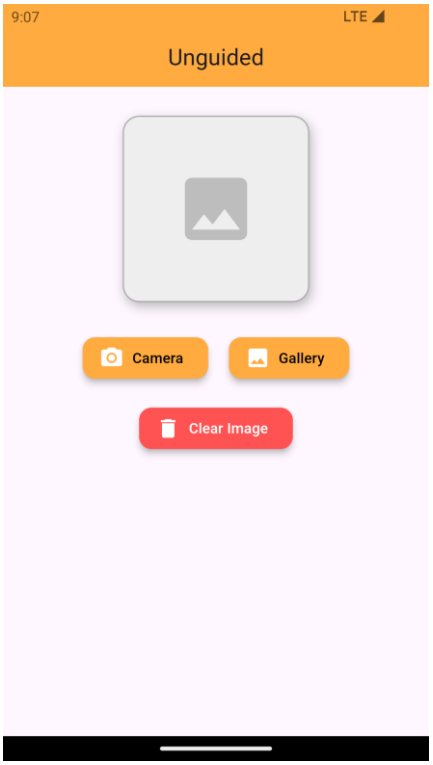
```
import 'package:flutter/material.dart';
import 'package:image_picker/image_picker.dart';
import 'dart:io';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: ImagePickerScreen(),
    );
  }
}

class ImagePickerScreen extends StatefulWidget {
  @override
  _ImagePickerScreenState createState() => _ImagePickerScreenState();
}

class _ImagePickerScreenState extends State<ImagePickerScreen> {
  File? _image;

  final ImagePicker _picker = ImagePicker();

  Future<void> _pickImageFromGallery() async {
          final    pickedFile    =    await    _picker.pickImage(source:
ImageSource.gallery);

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

  Future<void> _pickImageFromCamera() async {
    final pickedFile = await _picker.pickImage(source: ImageSource.camera);

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

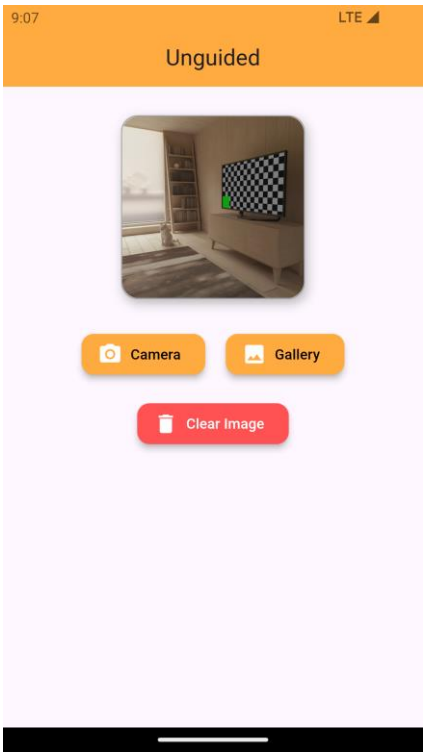
```
  void _clearImage() {
    setState(() {
      _image = null;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Unguided'),
        centerTitle: true,
        backgroundColor: Colors.orangeAccent,
      ),
      body: Center(
        child: Padding(
           padding: const EdgeInsets.symmetric(horizontal: 16.0, vertical:
8.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.start,
            children: [
              SizedBox(height: 20),
              Container(
                height: 180,
                width: 180,
                decoration: BoxDecoration(
                  border: Border.all(color: Colors.grey),
                  borderRadius: BorderRadius.circular(16),
                  color: Colors.grey[200],
                  boxShadow: [
                    BoxShadow(
                      color: Colors.black26,
                      blurRadius: 8,
                      offset: Offset(2, 4),
                    ),
                  ],
                ),
                child: _image != null
                    ? ClipRRect(
                        borderRadius: BorderRadius.circular(16),
                        child: Image.file(_image!, fit: BoxFit.cover),
                      )
                    : Center(
                        child: Icon(
                          Icons.image,
                          color: Colors.grey[400],
                          size: 80,
                        ),
                      ),
              ),
```

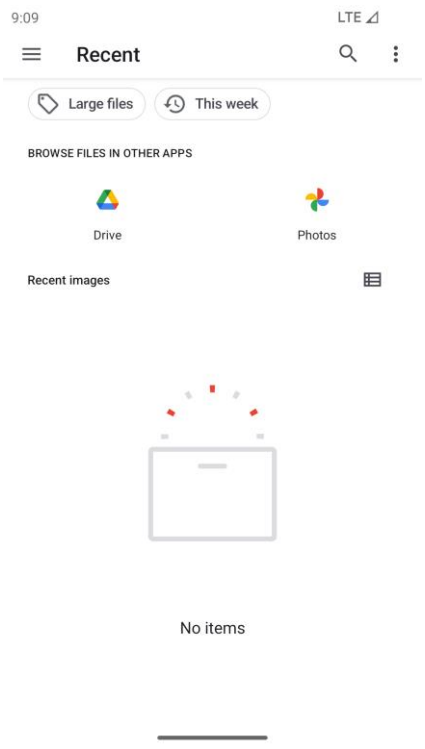
```
              SizedBox(height: 30),
              Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  ElevatedButton.icon(
                    onPressed: _pickImageFromCamera,
                    icon: Icon(Icons.camera_alt, color: Colors.white),
                           label:  Text("Camera",  style:  TextStyle(color:
Colors.black),),
                    style: ElevatedButton.styleFrom(
                      backgroundColor: Colors.orangeAccent,
                      shape: RoundedRectangleBorder(
                        borderRadius: BorderRadius.circular(12),
                      ),
                      elevation: 5,
                    ),
                  ),
                  SizedBox(width: 20),
                  ElevatedButton.icon(
                    onPressed: _pickImageFromGallery,
                    icon: Icon(Icons.photo, color: Colors.white),
                          label:  Text("Gallery",  style:  TextStyle(color:
Colors.black)),
                    style: ElevatedButton.styleFrom(
                      backgroundColor: Colors.orangeAccent,
                      shape: RoundedRectangleBorder(
                        borderRadius: BorderRadius.circular(12),
                      ),
                      elevation: 5,
                    ),
                  ),
                ],
              ),
              SizedBox(height: 20),
              ElevatedButton.icon(
                onPressed: _clearImage,
                icon: Icon(Icons.delete, color: Colors.white),
                     label:  Text("Clear  Image",  style:  TextStyle(color:
Colors.white)),
                style: ElevatedButton.styleFrom(
                  backgroundColor: Colors.redAccent,
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(12),
                  ),
                  elevation: 5,
                ),
              ),
            ],
          ),
        ),
```

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

Output:


9:07
LTE
Unguided
Camera
Gallery
Clear Image


9:07
LTE
Unguided
Camera
Gallery
Clear Image


9:09
LTE
Recent
Large files
This week
BROWSE FILES IN OTHER APPS
Drive
Photos
Recent images
No items

5. **Kesimpulan**

   Dengan menggunakan plugin paket camera dan media dalam pengembangan aplikasi berbasis flutter akan sangat memudahkan pengembang, contoh penerapannya seperti pada guided dan unguided dimana kita bisa menjalankan fitur kamera dari perangkat dan mengunggah gambar dari folder.