

TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK

MODUL IX
API PERANGKAT KERAS



Disusun Oleh:
Salman alfa Rizzi
Kelas:
SE-06-02

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 PURWOKERTO2024

API PERANGKAT KERAS

Tujuan Praktikum

Mahasiswa mampu memahami konsep layout pada Flutter

Mahasiswa dapat mengimplementasikan desain user interface pada Flutter

1. Camera API


Camera API berfungsi untuk memungkinkan *developer* (pengembang) untuk mengakses dan mengontrol kamera perangkat. Flutter menyediakan paket camera yang memudahkan implementasi fitur kamera untuk mengambil foto, merekam video, dan mengakses umpan kamera secara langsung. Paket ini sangat berguna untuk membuat aplikasi yang membutuhkan pengambilan gambar atau video, seperti aplikasi media sosial atau e-commerce.

Pada bahasan kali ini, kita akan menggunakan packages atau plugin Camera supaya aplikasi yang dibuat dapat mengakses kamera yang ada pada device.

Cara instalasi:

- Tambahkan paket *camera* yang ada pada Pub Dev di *pubspec.yaml*
- Lalu 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" />
```

 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:

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

Android

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

```
minSdkVersion 21
```

Hal tersebut penting karena MediaRecorder tidak bekerja dengan baik di emulator sesuai dengan [dokumentasi](#).

d. Implementasi kamera pada halaman Flutter

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

class CameraScreen extends StatefulWidget {
  @override
  _CameraScreenState createState() => _CameraScreenState();
}

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

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

```

        // Initialize camera on startup
        _initializeCamera();
    }

    Future<void> _initializeCamera() async {
        // Ambil daftar kamera yang tersedia di perangkat
        final cameras = await availableCameras();
        final firstCamera = cameras.first;

        // Buat kontroler kamera dan mulai kamera
        _controller = CameraController(
            firstCamera,
            ResolutionPreset.high,
        );

        initializeControllerFuture = controller.initialize();
    }

    @override
    void dispose() {
        // Bersihkan kontroler ketika widget dihapus
        _controller.dispose();
        super.dispose();
    }

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(title: Text('Camera Example')),
            body: FutureBuilder<void>(
                future: _initializeControllerFuture,
                builder: (context, snapshot) {
                    if (snapshot.connectionState ==
ConnectionState.done) {
                        return CameraPreview(_controller);
                    } else {
                        return
                            Center(child:
CircularProgressIndicator());
                    }
                }
            )
        );
    }

```

```

    }
  },
),
floatingActionButton: FloatingActionButton(
  onPressed: () async {
    try {
      // Pastikan kamera sudah diinisialisasi
      await _initializeControllerFuture;
      // Ambil gambar
      final image = await _controller.takePicture();
      // Tampilkan atau gunakan gambar
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) =>
DisplayPictureScreen(imagePath: image.path),
        ),
      );
    } catch (e) {
      print(e);
    }
  },
  child: Icon(Icons.camera_alt),
),
);
}
}

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 Picture')),

```

```
body: Image.file(File(imagePath)),  
    );  
}  
}
```

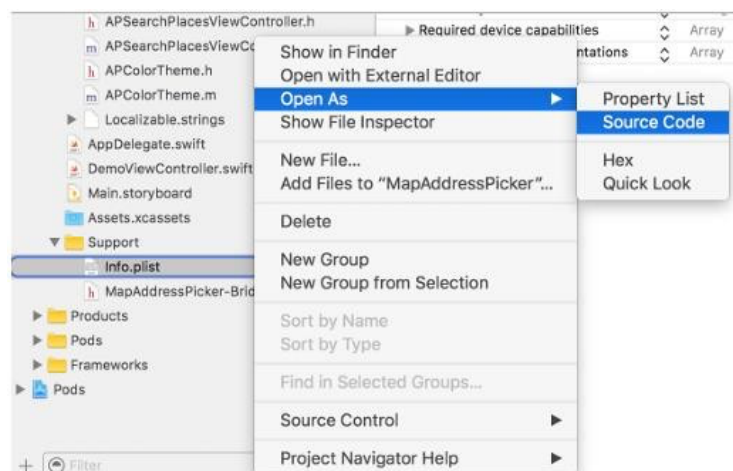
2. Media API

Media API adalah sekumpulan alat dan pustaka yang mendukung pengelolaan dan interaksi dengan berbagai jenis media, seperti gambar, video, dan audio. Flutter tidak memiliki API media bawaan untuk semua kebutuhan media, tetapi dapat menggunakan paket-paket tambahan untuk mengakses fitur media yang umum di aplikasi.

Pada bahasan kali ini, kita akan menggunakan packages atau plugin **Image Picker** supaya aplikasi dapat mengakses media galeri pada device. Pada platform iOS diperlukan konfigurasi tambahan, namun pada android tidak diperlukan konfigurasi tambahan. Berikut untuk konfigurasi tambahan:

A. iOS

Pada iOS diperlukan versi iOS 9.0.0 keatas, pastikan build version di set ke iOS 9.0.0 atau lebih tinggi. tambahkan konfigurasi pada file info.plist di Runner iOS. Kemudian ubah format tampilan **.plist** dengan tipe source code. Dapat dilihat pada gambar dibawah.



- Tambahkan potongan kode berikut pada **info.plist**

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

- Untuk penggunaan image picker, dapat dilihat pada potongan kode berikut :

```
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 }

```

Tugas Mandiri (Unguided)

1. (Soal) 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, screenshot output, dan deskripsi program.

Kreatifitas menjadi nilai tambah.

Source Code:

```
import 'package:flutter/material.dart';
import 'package:pertemuan_9/GUIDED/camera.screnn.dart';
import 'package:pertemuan_9/TP/api.dart';

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.green,
      ),
      home: const MyApi(), // Pastikan nama kelas di sini sesuai
    );
  }
}
```

Deskripsi Program:

Program ini adalah aplikasi Flutter yang memungkinkan pengguna mengambil dan memilih gambar menggunakan kamera perangkat atau galeri, serta menampilkan hasilnya di layar. Aplikasi ini menginisialisasi kamera dan menampilkan pratinjau menggunakan widget CameraPreview, serta memungkinkan pengguna mengambil gambar melalui tombol khusus. Selain itu, pengguna dapat memilih gambar dari galeri perangkat menggunakan library image_picker. Hasil gambar ditampilkan di layar, dan pengguna dapat menghapus gambar tersebut dengan tombol khusus untuk mengembalikan tampilan ke kondisi awal. Jika terjadi kegagalan saat mengambil atau memilih gambar, aplikasi akan menampilkan notifikasi kesalahan melalui ScaffoldMessenger untuk memberikan informasi kepada pengguna. Program ini ditulis dalam bahasa Dart dengan menggunakan berbagai library Flutter seperti camera dan image_picker.

ScreenShoot:

