|  |  |  |
| --- | --- | --- |
| Mata Kuliah | **:** | Pemograman Mobile |
| Program Studi | **:** | D4 – Sistem Informasi Bisnis |
| Semester | **:** | 5 |
| Kelas | **:** | SIB |
| NIM | **:** | 2341760056 |
| Nama | **:** | Revani Nanda Putri |
| Jobsheet Ke- | **:** | 7 (Aplikasi OCR Sederhana dengan Flutter) |
| Link Github | **:** | <https://github.com/revaniputeri/mobile-programming-labs/tree/main/jobsheet-7> |

**Laporan Jobsheet**

**Praktikum Ke-1 (Membangun Layout di Flutter)**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 1 | Membuat project Flutter baru |
| 2 | Buka file pubspec.yaml, lalu tambahkan dependensi berikut di bawah bagian dependencies:  dependencies:    flutter:      sdk: flutter    google\_mlkit\_text\_recognition: ^0.10.0    camera: ^0.10.5+5    path\_provider: ^2.1.2    path: ^1.8.3 |
| 3 | Simpan file, lalu jalankan: |
| 4 | Buka file: android/app/src/main/AndroidManifest.xml  Tambahkan baris berikut di dalam tag <manifest>, sebelum <application>:  <uses-permission android:name ="android.permission.CAMERA" /> |
| 5 | Di dalam folder lib/, buat struktur berikut: |
| 6 | Kode program:  lib/main.dart  import 'package:flutter/material.dart';  import 'screens/splash\_screen.dart';  void main() {    runApp(const *MyApp*());  }  class *MyApp* extends *StatelessWidget* {    const *MyApp*({super.key});    @override  *Widget* build(*BuildContext* context) {      return *MaterialApp*(        title: 'OCR Sederhana',        theme: *ThemeData*(primarySwatch: *Colors*.blue),        home: const *SplashScreen*(),        debugShowCheckedModeBanner: false,      );    }  }  File: lib/screens/splash screen.dart  import 'dart:async';  import 'package:flutter/material.dart';  import 'home\_screen.dart';  class *SplashScreen* extends *StatefulWidget* {    const *SplashScreen*({super.key});    @override  *State*<*SplashScreen*> createState() => *\_SplashScreenState*();  }  class *\_SplashScreenState* extends *State*<*SplashScreen*> {    @override    void initState() {      super.initState();  *Timer*(const *Duration*(seconds: 2), () {  *Navigator*.pushReplacement(          context,  *MaterialPageRoute*(builder: (\_) => const *HomeScreen*()),        );      });    }    @override  *Widget* build(*BuildContext* context) {      return *Scaffold*(        backgroundColor: *Colors*.blue,        body: *Center*(          child: *Column*(            mainAxisAlignment: *MainAxisAlignment*.center,            children: const [  *CircularProgressIndicator*(color: *Colors*.white),  *SizedBox*(height: 20),  *Text*('OCR Scanner',                style: *TextStyle*(color: *Colors*.white, fontSize: 24)),            ],          ),        ),      );    }  }  File: lib/screens/home\_screen.dart  import 'package:flutter/material.dart';  import 'scan\_screen.dart';  class *HomeScreen* extends *StatelessWidget* {    const *HomeScreen*({super.key});    @override  *Widget* build(*BuildContext* context) {      return *Scaffold*(        appBar: *AppBar*(title: const *Text*('Menu Utama')),        body: *Center*(          child: *ElevatedButton*(            onPressed: () {  *Navigator*.push(                context,  *MaterialPageRoute*(builder: (\_) => const *ScanScreen*()),              );            },            child: const *Text*('Mulai Scan Teks'),          ),        ),      );    }  }  File: lib/screens/scan\_screen.dart  import 'dart:io';  import 'package:flutter/material.dart';  import 'package:camera/camera.dart';  import 'package:google\_mlkit\_text\_recognition/google\_mlkit\_text\_recognition.dart';  import 'package:path/path.dart' as path;  import 'package:path\_provider/path\_provider.dart';  import 'result\_screen.dart';  late *List*<*CameraDescription*> cameras;  class *ScanScreen* extends *StatefulWidget* {    const *ScanScreen*({super.key});    @override  *State*<*ScanScreen*> createState() => *\_ScanScreenState*();  }  class *\_ScanScreenState* extends *State*<*ScanScreen*> {    late *CameraController* \_controller;    late *Future*<void> \_initializeControllerFuture;    @override    void initState() {      super.initState();      \_initCamera();    }    void \_initCamera() async {      cameras = await availableCameras();      \_controller = *CameraController*(cameras[0], *ResolutionPreset*.medium);      \_initializeControllerFuture = \_controller.initialize();      if (mounted) {        setState(() {});      }    }    @override    void dispose() {      \_controller.dispose();      super.dispose();    }  *Future*<*String*> \_ocrFromFile(*File* imageFile) async {      final inputImage = *InputImage*.fromFile(imageFile);      final textRecognizer = *TextRecognizer*(script: *TextRecognitionScript*.latin);      final *RecognizedText* recognizedText = await textRecognizer.processImage(inputImage);      textRecognizer.close();      return recognizedText.text;    }  *Future*<void> \_takePicture() async {      try {        await \_initializeControllerFuture;        if (!mounted) return;  *ScaffoldMessenger*.of(context).showSnackBar(          const *SnackBar*(content: *Text*('Memproses OCR, mohon tunggu...'), duration: *Duration*(seconds: 2)));        final *XFile* image = await \_controller.takePicture();        final ocrText = await \_ocrFromFile(*File*(image.path));        if (!mounted) return;  *Navigator*.push(          context,  *MaterialPageRoute*(builder: (\_) => *ResultScreen*(ocrText: ocrText)),        );      } catch (e) {        if (!mounted) return;  *ScaffoldMessenger*.of(context).showSnackBar(*SnackBar*(content: *Text*('Error saat mengambil/memproses foto: $e')));      }    }    @override  *Widget* build(*BuildContext* context) {      if (!\_controller.value.isInitialized) {        return const *Scaffold*(body: *Center*(child: *CircularProgressIndicator*()));      }      return *Scaffold*(        appBar: *AppBar*(title: const *Text*('Kamera OCR')),        body: *Column*(          children: [  *Expanded*(              child: *AspectRatio*(                aspectRatio: \_controller.value.aspectRatio,                child: *CameraPreview*(\_controller),              ),            ),  *Padding*(              padding: const *EdgeInsets*.all(16.0),              child: *ElevatedButton*.icon(                onPressed: \_takePicture,                icon: const *Icon*(*Icons*.camera),                label: const *Text*('Ambil Foto & Scan'),              ),            ),          ],        ),      );    }  }  File: lib/screens/result\_screen.dart  import 'package:flutter/material.dart';  class *ResultScreen* extends *StatelessWidget* {    final *String* ocrText;    const *ResultScreen*({super.key, required this.ocrText});    @override  *Widget* build(*BuildContext* context) {      return *Scaffold*(        appBar: *AppBar*(title: const *Text*('Hasil OCR')),        body: *Padding*(          padding: const *EdgeInsets*.all(16.0),          child: *SingleChildScrollView*(            child: *SelectableText*(              ocrText.isEmpty                  ? 'Tidak ada teks ditemukan.'                  : ocrText.replaceAll('\n', ' '),              style: const *TextStyle*(fontSize: 18),            ),          ),        ),      );    }  } |

**Tugas Praktikum**

|  |  |
| --- | --- |
| **Langkah** | **Jawaban/Deskripsi** |
| 1 | Jalankan aplikasi di emulator atau HP. |
| 2 | Lakukan scan terhadap teks cetak (misal: buku, koran, atau layar HP). |
| 3 | Amati hasil OCR yang muncul. |
| 4 | Jawab pertanyaan berikut:   1. Apakah semua teks terbaca dengan akurat? Mengapa?   Jawab:  Tidak semua teks selalu terbaca dengan akurat. Contoh di atas hasilnya terbaca “halo” dengan benar karena pencahayaan cukup, hurufnya jelas, dan kontras tinggi antara teks dan background. Tapi kalau posisi miring, cahaya terlalu gelap/terang, atau font-nya aneh, hasil OCR bisa jadi salah baca.   1. Apa kegunaan fitur OCR dalam kehidupan sehari-hari?   Jawab:  untuk mengubah gambar atau dokumen cetak menjadi teks digital yang bisa diedit, disalin, atau dicari. Misalnya buat memindai nota, kartu identitas, buku, atau formulir.   1. Sebutkan 2 contoh aplikasi nyata yang menggunakan OCR!   Jawab:   1. Google Lens bisa menyalin teks dari foto langsung ke clipboard 2. Microsoft OneNote / Adobe Scan buat memindai dokumen dan mengubahnya jadi teks yang bisa diedit. |