

JOBSHEET – APLIKASI OCR SEDERHANA DENGAN FLUTTER

1. IDENTITAS PRAKTIKAN

Komponen	Isi
Nama	<u>Vita Eka Saraswati</u>
Kelas / NIM	<u>29 / 2341760082</u>
Tanggal	<u>16 Oktober 2025</u>
Guru / Dosen	<u>Ade Ismail, S.Kom., M.TI</u>

2. TUJUAN PRAKTIKUM

Setelah menyelesaikan jobsheet ini, siswa/mahasiswa mampu:

1. Membuat aplikasi Flutter multi-halaman.
2. Menggunakan plugin kamera untuk mengambil gambar.
3. Mengintegrasikan **OCR (Optical Character Recognition)** menggunakan library `google_mlkit_text_recognition`.
4. Menampilkan hasil OCR di halaman terpisah.
5. Menerapkan navigasi dasar antar layar menggunakan Navigator.

3. ALAT DAN BAHAN

- Laptop/komputer dengan Flutter SDK terinstal
- VS Code atau Android Studio
- Emulator Android atau perangkat Android fisik
- Koneksi internet (untuk instalasi dependensi)

4. LANGKAH KERJA

4.1. Langkah 1: Buat Proyek Baru

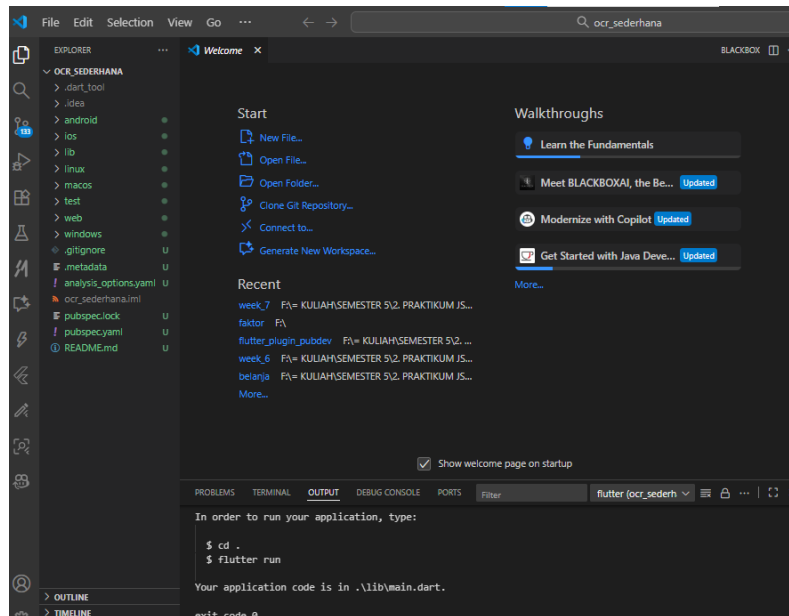
Buka terminal, lalu jalankan:

```

1 flutter create ocr_sederhana
2 cd ocr_sederhana

```

Listing 1: Membuat proyek Flutter



4.2. Langkah 2: Tambahkan Plugin

Buka file pubspec.yaml, lalu tambahkan dependensi berikut di bawah bagian dependencies:

```

1 dependencies:
2   flutter:
3     sdk : flutter
4   google_mlkit_text_recognition: ^0.10.0
5   camera: ^0.10.5+5
6   path_provider: ^2.1.2
7   path: ^1.8.3

```

```

30 dependencies:
31   flutter:
32     sdk: flutter
33     google_mlkit_text_recognition: ^0.15.0
34     google_mlkit_commons: ^0.11.0
35     camera: ^0.10.6
36     path_provider: ^2.1.2
37     path: ^1.8.3
38     cupertino_icons: ^1.0.8
39
40
41 dev_dependencies:

```

Listing 2: pubspec.yaml - dependencies

Simpan file, lalu jalankan:

```
1 flutter pub get
```

```
PS F:\KULIAH\SEMESTER 5\2. PRAKTIKUM JS\PEMROGRAMAN MOBILE\github\pem_mobile\week_7\ocr_sederhana> flutter pub get
Resolving dependencies... (1.5s)
Downloading packages...
camera 0.10.6 (0.11.2 available)
characters 1.4.0 (1.4.1 available)
flutter_lints 5.0.0 (6.0.0 available)
google_mlkit_commons 0.5.0 (0.11.0 available)
google_mlkit_text_recognition 0.10.0 (0.15.0 available)
lints 5.1.1 (6.0.0 available)
material_color_utilities 0.11.1 (0.13.0 available)
meta 1.16.0 (1.17.0 available)
test_api 0.7.6 (0.7.7 available)
Got dependencies!
9 packages have newer versions incompatible with dependency constraints.
Try 'flutter pub outdated' for more information.
```

4.3. Langkah 3: Tambahkan Izin Kamera (Android)

Buka file: android/app/src/main/AndroidManifest.xml

Tambahkan baris berikut di dalam tag <manifest>, sebelum <application>:

```
1 <uses-permission android:name="android.permission.CAMERA" />
```

4.4. Langkah 4: Buat Struktur Folder

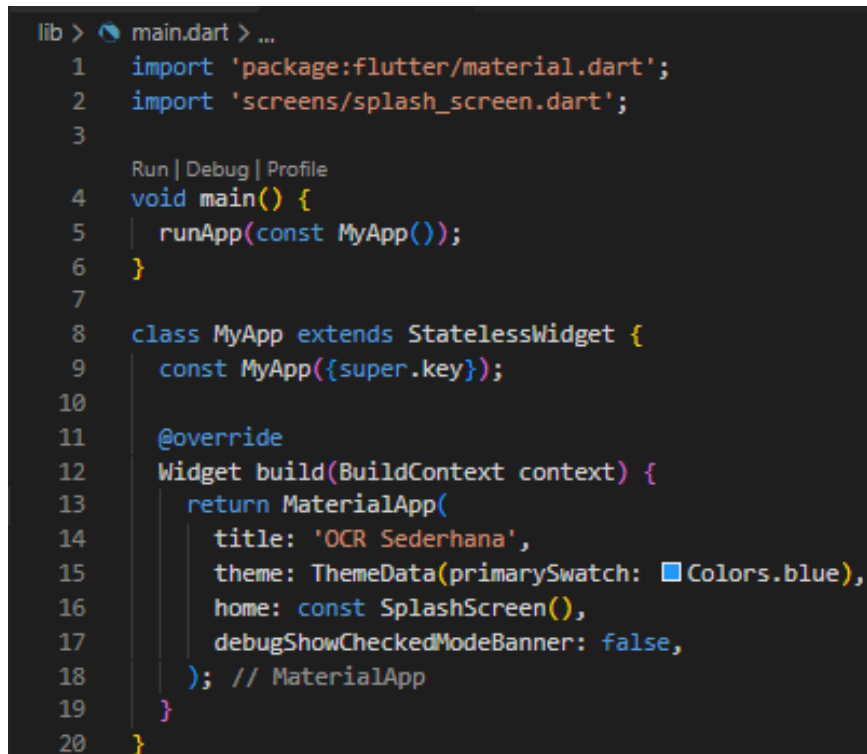
Di dalam folder lib/, buat struktur berikut:

```
1 lib /
2     main.dart
3     screens/
4         splash_screen.dart
5         home_screen.dart
6         scan_screen.dart
7         result_screen.dart
```

5. KODE PROGRAM

5.1. File: lib/main.dart

```
1 import 'package:flutter/material.dart';
2 import 'screens/splash_screen.dart';
3
4 void main () {
5   runApp(const MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   const MyApp({super.key});
10
11   @override
12   Widget build(BuildContext context) {
13     return MaterialApp(
14       title: 'OCR Sederhana',
15       theme: ThemeData(primarySwatch: Colors.blue),
16       home : const SplashScreen(),
17       debugShowCheckedModeBanner: false,
18     );
19   }
20 }
```



```
lib > main.dart > ...
1  import 'package:flutter/material.dart';
2  import 'screens/splash_screen.dart';
3
4  Run | Debug | Profile
5  void main() {
6    runApp(const MyApp());
7  }
8
9  class MyApp extends StatelessWidget {
10    const MyApp({super.key});
11
12    @override
13    Widget build(BuildContext context) {
14      return MaterialApp(
15        title: 'OCR Sederhana',
16        theme: ThemeData(primarySwatch: Colors.blue),
17        home: const SplashScreen(),
18        debugShowCheckedModeBanner: false,
19      ); // MaterialApp
20    }
21  }
```

Listing 3: main.dart

5.2. File: lib/screens/splash_screen.dart

```
1 import 'dart:async';
2 import 'package:flutter/material.dart';
3 import 'home_screen.dart';
4
5 class SplashScreen extends StatefulWidget {
6   const SplashScreen({super.key});
7
8   @override
9   State<SplashScreen> createState() => _SplashScreenState();
10 }
11
12 class _SplashScreenState extends State<SplashScreen> {
13   @override
14   void initState() {
15     super.initState();
16     Timer(const Duration(seconds: 2), () {
17       Navigator.pushReplacement(
18         context,
19         MaterialPageRoute(builder: (_) => const HomeScreen()),
20       );
21     });
22   }
23
24   @override
25   Widget build(BuildContext context) {
26     return Scaffold(
27       backgroundColor: Colors.blue,
28       body: Center(
29         child: Column(
30           mainAxisAlignment: MainAxisAlignment.center,
31           children: const [
32             CircularProgressIndicator(color: Colors.white),
33             SizedBox(height: 20),
34             Text('OCR Scanner',
35               style: TextStyle(color: Colors.white, fontSize:
36                 24)),
37             ],
38         ),
39     );
40   }
41 }
```

```

lib > screens > splash_screen.dart > _SplashScreenState
1  import 'dart:async';
2  import 'package:flutter/material.dart';
3  import 'home_screen.dart';
4
5  class SplashScreen extends StatefulWidget {
6    const SplashScreen({super.key});
7
8    @override
9    State<SplashScreen> createState() => _SplashScreenState();
10 }
11
12 class _SplashScreenState extends State<SplashScreen> {
13   @override
14   void initState() {
15     super.initState();
16     Timer(const Duration(seconds: 2), () {
17       Navigator.pushReplacement(
18         context,
19         MaterialPageRoute(builder: (_) => const HomeScreen()),
20       );
21     }); // Timer
22   }
23
24   @override
25   Widget build(BuildContext context) {
26     return Scaffold(
27       backgroundColor: Colors.blue,
28       body: Center(
29         child: Column(
30           mainAxisAlignment: MainAxisAlignment.center,
31           children: const [
32             CircularProgressIndicator(color: Colors.white),
33             SizedBox(height: 20),
34             Text(
35               'OCR Scanner',
36               style: TextStyle(color: Colors.white, fontSize: 24),
37             ), // Text
38           ],
39         ), // Column
40       ), // Center
41     ); // Scaffold
42   }
43 }

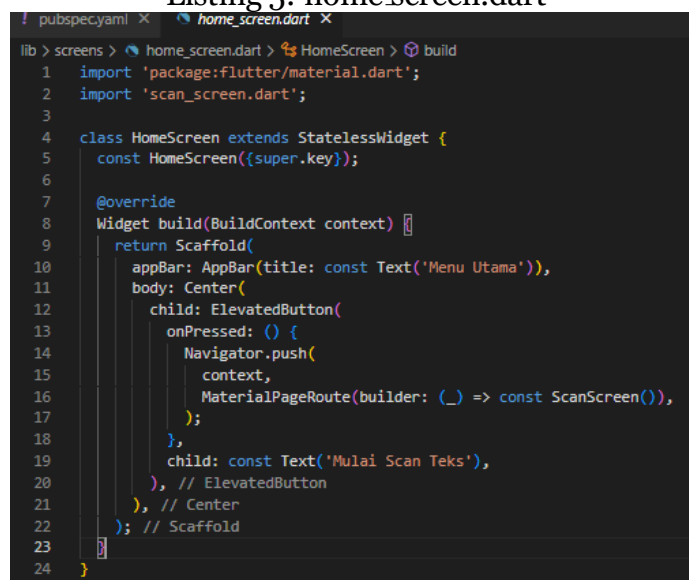
```

Listing 4: splash_screen.dart

5.3. File: lib/screens/home_screen.dart

```
1 import 'package:flutter/material.dart';
2 import 'scan_screen.dart';
3
4 class HomeScreen extends StatelessWidget {
5   const HomeScreen({super.key});
6
7   @override
8   Widget build(BuildContext context) {
9
10    return Scaffold(
11      appBar: AppBar(title: const Text('Menu Utama')),
12      body: Center(
13        child: ElevatedButton(
14          onPressed: () {
15            Navigator.push(
16              context,
17              MaterialPageRoute(builder: (_) => const ScanScreen
18                ),
19            );
20          },
21          child: const Text('Mulai Scan Teks'),
22        ),
23      ),
24    );
25  }
```

Listing 5: home_screen.dart



```
pubspec.yaml x home_screen.dart x
lib > screens > home_screen.dart > HomeScreen > build
1 import 'package:flutter/material.dart';
2 import 'scan_screen.dart';
3
4 class HomeScreen extends StatelessWidget {
5   const HomeScreen({super.key});
6
7   @override
8   Widget build(BuildContext context) {
9     return Scaffold(
10       appBar: AppBar(title: const Text('Menu Utama')),
11       body: Center(
12         child: ElevatedButton(
13           onPressed: () {
14             Navigator.push(
15               context,
16               MaterialPageRoute(builder: (_) => const ScanScreen()),
17             );
18           },
19           child: const Text('Mulai Scan Teks'),
20         ), // ElevatedButton
21       ), // Center
22     ); // Scaffold
23   }
24 }
```

5.4 File: lib/screens/scan_screen.dart

```
lib > screens > scan_screen.dart > _ScanScreenState
1  import 'dart:io';
2  import 'package:flutter/material.dart';
3  import 'package:camera/camera.dart';
4  import 'package:google_mlkit_text_recognition/google_mlkit_text_recognition.dart';
5  import 'package:path_provider/path_provider.dart';
6  import 'result_screen.dart';
7
8  late List<CameraDescription> cameras;
9
10 class ScanScreen extends StatefulWidget {
11   const ScanScreen({super.key});
12
13   @override
14   State<ScanScreen> createState() => _ScanScreenState();
15 }
16
17 class _ScanScreenState extends State<ScanScreen> {
18   CameraController? _controller;
19   late Future<void> _initializeControllerFuture;
20
21   @override
22   void initState() {
23     super.initState();
24     _initCamera();
25   }
26
27   void _initCamera() async {
28     try {
29       cameras = await availableCameras();
30       _controller = CameraController(cameras.first, ResolutionPreset.medium);
31
32       _initializeControllerFuture = _controller!.initialize();
33       await _initializeControllerFuture;
34
35       if (mounted) {
36         setState(() {});
37       }
38     } catch (e) {
39       debugPrint('Error initializing camera: $e');
40     }
41   }
42
43   @override
44   void dispose() {
45     _controller?.dispose();
46     super.dispose();
47   }
48
49   Future<String> _ocrFromFile(File imageFile) async {
50     final inputImage = InputImage.fromFile(imageFile);
51     final textRecognizer = TextRecognizer(script: TextRecognitionScript.latin);
52     final RecognizedText recognizedText = await textRecognizer.processImage(
53       inputImage,
54     );
55     textRecognizer.close();
56     return recognizedText.text;
57   }
58
59   Future<void> _takePicture() async {
60     if (_controller == null) return;
61
62     try {
63       await _initializeControllerFuture;
64
65       if (!mounted) return;
66
67       ScaffoldMessenger.of(context).showSnackBar(
68         const SnackBar(
69           content: Text('Memproses OCR, mohon tunggu...'),
70           duration: Duration(seconds: 2),
71         ), // SnackBar
72       );
73
74       final XFile image = await _controller!.takePicture();
75       final ocrText = await _ocrFromFile(File(image.path));
76     }
```



```

77     if (!mounted) return;
78     Navigator.push(
79       context,
80       MaterialPageRoute(builder: (_) => ResultScreen(ocrText: ocrText)),
81     );
82   } catch (e) {
83     if (!mounted) return;
84     ScaffoldMessenger.of(
85       context,
86     ).showSnackBar(SnackBar(content: Text('Error: $e')));
87   }
88 }
89
90 @override
91 Widget build(BuildContext context) {
92   if (_controller == null || !_controller!.value.isInitialized) {
93     return const Scaffold(body: Center(child: CircularProgressIndicator()));
94   }
95
96   return Scaffold(
97     appBar: AppBar(
98       title: const Text('Kamera OCR'),
99       centerTitle: true,
100       backgroundColor: Colors.deepPurple,
101     ), // AppBar
102     body: Column(
103       children: [
104         Expanded(
105           child: AspectRatio(
106             aspectRatio: _controller!.value.aspectRatio,
107             child: CameraPreview(_controller!),
108           ), // AspectRatio
109         ), // Expanded
110         Padding(
111           padding: const EdgeInsets.all(16.0),
112           child: ElevatedButton.icon(
113             style: ElevatedButton.styleFrom(
114               backgroundColor: Colors.deepPurple,
115               padding: const EdgeInsets.symmetric(
116                 horizontal: 24,
117                 vertical: 12,
118               ), // EdgeInsets.symmetric
119               shape: RoundedRectangleBorder(
120                 borderRadius: BorderRadius.circular(12),
121               ), // RoundedRectangleBorder
122             ),
123             onPressed: _takePicture,
124             icon: const Icon(Icons.camera_alt, color: Colors.white),
125             label: const Text(
126               'Ambil Foto & Scan',
127               style: TextStyle(color: Colors.white),
128             ), // Text
129           ), // ElevatedButton.icon
130         ), // Padding
131       ],
132     ), // Column
133   ); // Scaffold
134 }
135

```

Listing 6: scan screen.dart

5.5 File: lib/screens/result screen.dart

```
lib > screens > result_screen.dart > ResultScreen > build
1  import 'package:flutter/material.dart';
2
3  class ResultScreen extends StatelessWidget {
4    final String ocrText;
5
6    const ResultScreen({super.key, required this.ocrText});
7
8    @override
9    Widget build(BuildContext context) {
10     return Scaffold(
11       appBar: AppBar(title: const Text('Hasil OCR')),
12       body: Padding(
13         padding: const EdgeInsets.all(16.0),
14         child: SingleChildScrollView(
15           child: SelectableText(
16             ocrText.isEmpty
17               ? 'Tidak ada teks ditemukan.'
18               : ocrText.replaceAll('\n', ' '),
19             style: const TextStyle(fontSize: 18),
20           ), // SelectableText
21         ), // SingleChildScrollView
22       ), // Padding
23     ); // Scaffold
24   }
25 }
```

Listing 7: result.screen.dart

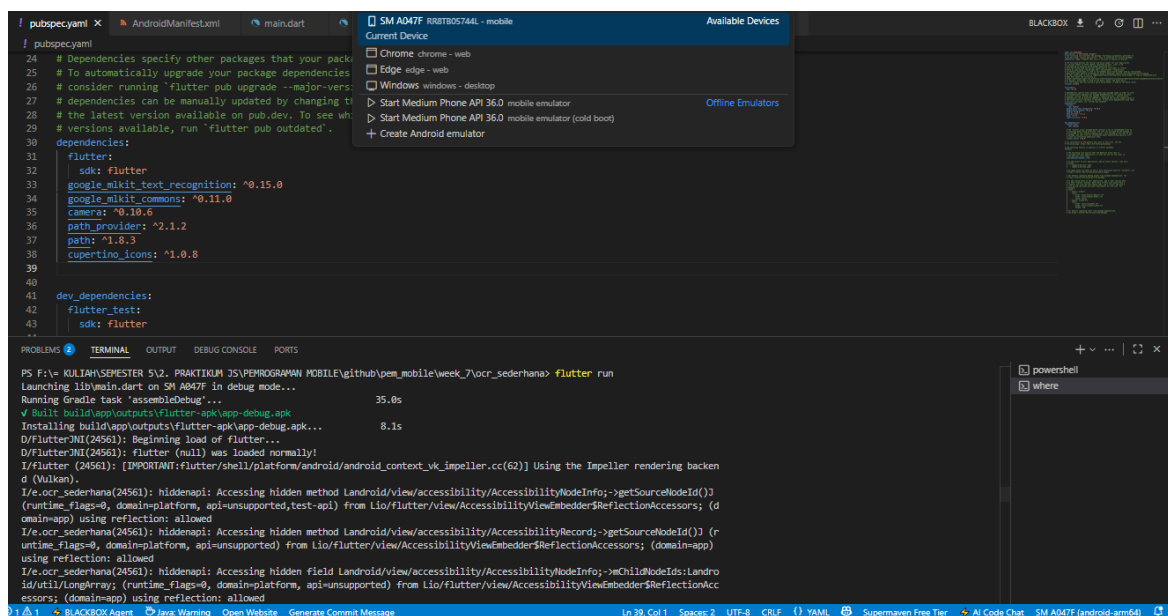
6 TUGAS PRAKTIKUM

1. Jalankan aplikasi di emulator atau HP.

Pada praktikum kali ini saya menggunakan device handphone android dan USB untuk menjalankan kode program.

**Catatan :*

- Sebelum menjalankan kode program pastikan komputer/PC dan device yang digunakan telah mengaktifkan developer mode (mode pengembang).
- Setelah developer mode aktif, jalankan perintah flutter run.

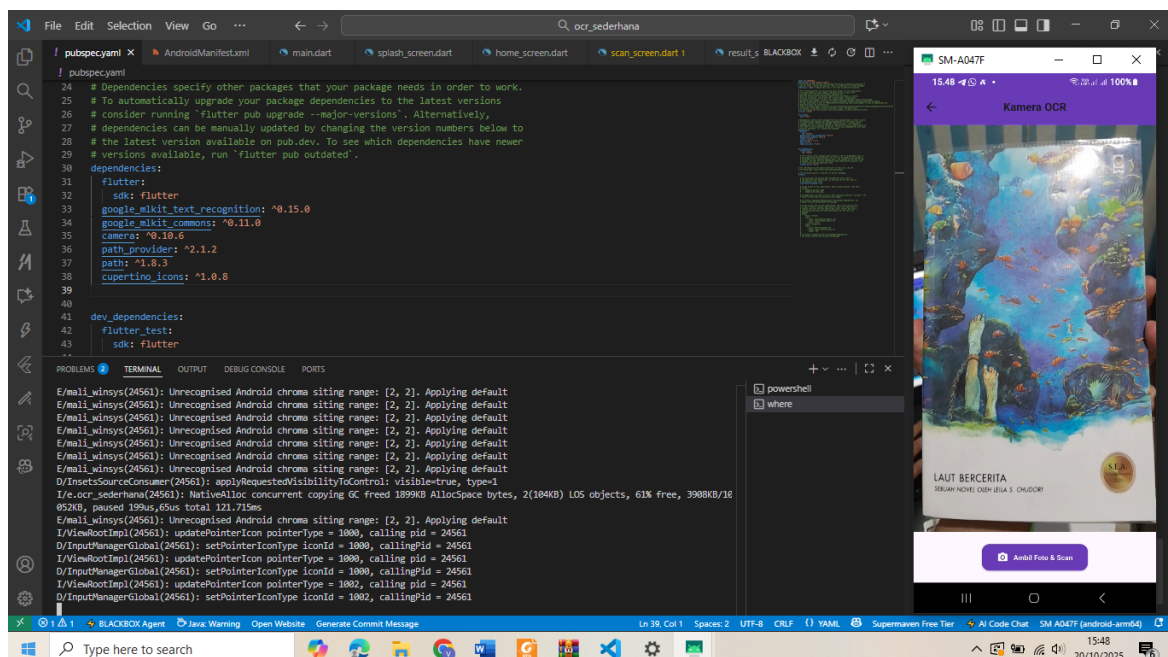


```
pubspec.yaml
! pubspec.yaml
# Dependencies specify other packages that your package
# To automatically upgrade your package dependencies
# consider running 'flutter pub upgrade --major-versions'
# dependencies can be manually updated by changing the
# the latest version available on pub.dev. To see which
# versions available, run 'flutter pub outdated'.
dependencies:
  flutter:
    sdk: flutter
  google_mlkit_text_recognition: ^0.15.0
  google_mlkit_commons: ^0.11.0
  camera: ^0.10.6
  path_provider: ^2.1.2
  path: ^1.8.3
  cupertino_icons: ^1.0.8
dev_dependencies:
  flutter_test:
    sdk: flutter
  flutter_lints: ^2.0.1

PS F:\KULIAH\SEMESTER 5\2. PRAKTIKUM 35\PEMEROGAWAN MOBILE\github\pem_mobile\week_7\ocr_sederhana> flutter run
Launching lib/main.dart on SM-A047F in debug mode...
Running Gradle task 'assembleDebug'...
✓ Built build/app/outputs/flutter-app-debug.apk.
Installing build/app/outputs/flutter-app-debug.apk...
D:\FlutterJNI(24561): Beginning load of flutter...
D:\FlutterJNI(24561): flutter (null) was loaded normally!
I/flutter (24561): [D:\FlutterJNI(24561)\platform\android\android_context_vk_impeller.cc(62)] Using the Impeller rendering backend (Vulkan).
I/e.ocr_sederhana(24561): hiddenapi: Accessing hidden method Landroid/view/accessibility/AccessibilityNodeInfo->getSourceNodeId() (runtime_flags=0, domain=platform, api=unsupported, test-api) from Lio/flutter/view/AccessibilityViewEmbedder$ReflectionAccessors; (d main=app) using reflection: allowed
I/e.ocr_sederhana(24561): hiddenapi: Accessing hidden method Landroid/view/accessibility/AccessibilityRecord->getSourceNodeId() (runtime_flags=0, domain=platform, api=unsupported) from Lio/flutter/view/AccessibilityViewEmbedder$ReflectionAccessors; (domain=app) using reflection: allowed
I/e.ocr_sederhana(24561): hiddenapi: Accessing hidden field Landroid/view/accessibility/AccessibilityNodeInfo->getChildNodeIds() (runtime_flags=0, domain=platform, api=unsupported) from Lio/flutter/view/AccessibilityViewEmbedder$ReflectionAccessors; (domain=app) using reflection: allowed
Ln 39, Col 1 Spaces: 2 UTF-8 CRLF YAMLSupermarket Free TierAI Code ChatSM-A047F (android-arm64)
```

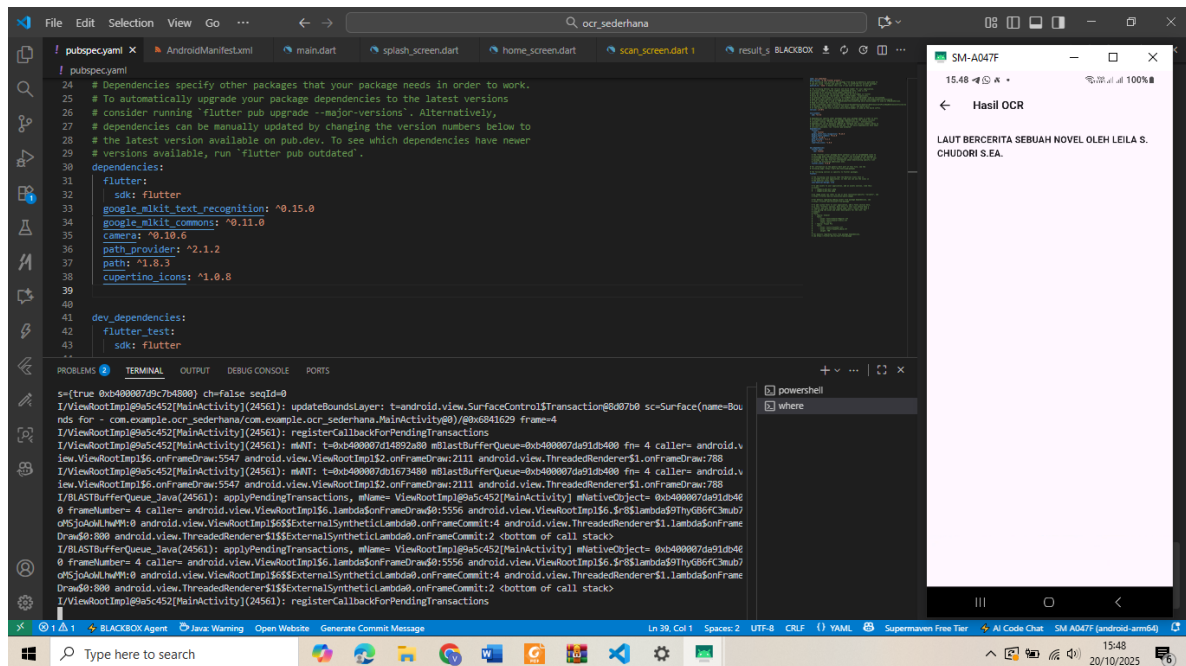
2. Lakukan scan terhadap teks cetak (misal: buku, koran, atau layar HP).

Jawab:



3. Amati hasil OCR yang muncul.

Jawab:



4. Jawab pertanyaan berikut:

- a. Apakah semua teks terbaca dengan akurat? Mengapa?

Jawab:

Pada percobaan di atas teks akurat. Tetapi perlu terkadang OCR juga menghasilkan teks yang tidak akurat karena dipengaruhi oleh kualitas gambar, pencahayaan, ukuran dan jenis font, serta noise atau bayangan. Sehingga seringkali teks yang buram, terlalu kecil, atau miring sering sulit terbaca dengan benar.

- b. Apa kegunaan fitur OCR dalam kehidupan sehari-hari?

Jawab

OCR berguna untuk **mengubah teks dari gambar atau dokumen fisik menjadi teks digital**. Contohnya memindai dokumen, membaca struk belanja atau faktur secara otomatis, membantu penyandang disabilitas membaca teks melalui aplikasi pembaca layar

- c. Sebutkan 2 contoh aplikasi nyata yang menggunakan OCR!

Jawab:

Contoh aplikasi nyata yang menggunakan OCR:

- o **Google Lens** membaca teks dari foto, menerjemahkan, atau menyalin teks.
- o **M-banking** yang membaca Dokumen misal nya untuk pembukaan Rekening, nasabah hanya perlu Mengambil gambar KTP maka kolom data diri terisi otomatis
- o **ETLE** - Sistem ETLE yang diterapkan oleh Korlantas Polri secara nasional di seluruh Indonesia yang menggunakan sistem OCR untuk membaca nomor plat kendaraan

7 CATATAN PENTING

- Pastikan kamera perangkat dalam kondisi baik dan pencahayaan cukup.
- Plugin google mlkit text recognition bekerja **offline** dan mendukung bahasa Latin (termasuk Indonesia).
- Jika muncul error saat pertama kali buka kamera, pastikan izin kamera sudah diizinkan di pengaturan HP.

8 PENILAIAN

Aspek	Skor (1–5)
Kelengkapan kode	
Aplikasi berjalan lancar	
Jawaban tugas	
Ketepatan waktu	
Total	

$$\text{Nilai Akhir} = \text{Total Skor} \times 5$$

Selamat mengerjakan!