

PEMROGRAMAN MOBILE
JOBSHEET 7 – APLIKASI OCR
SEDERHANA DENGAN FLUTTER



Oleh :

Muhammad Rohman Al Kautsar

2341760055

SIB 3C – 26

POLITEKNIK NEGERI MALANG
PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI

Link Github : https://github.com/rohmanalka/ocr_sederhana

Praktikum : Membuat ocr sederhana

1. Buat project baru dengan nama ocr_sederhana
2. Tambahkan dependensi berikut ke pubspec.yaml

```
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
```

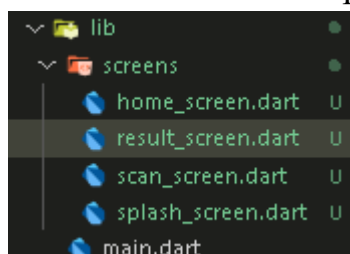
3. lalu jalankan flutter pub get

```
PS C:\JSFlutter\ocr_sederhana> flutter pub get  
Resolving dependencies...  
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)
```

4. Tambahkan Izin Kamera (Android), Tambahkan di file
C:\JSFlutter\ocr_sederhana\android\app\src\main\AndroidManifest.xml
sebelum application

```
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android">  
2 <uses-permission android:name="android.permission.CAMERA" />  
3 <application  
4     android:label="ocr_sederhana"  
5     android:name="${applicationName}"  
6     android:icon="@mipmap/ic_launcher">
```

5. Buat struktur folder seperti berikut dan isikan dengan kode



6. Isikan kode screens

```

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: 24)),
36           ],
37         ),
38       ),
39     );
40   }
41 }

```

```

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/path.dart' as path;
6 import 'package:path_provider/path_provider.dart';
7 import 'result_screen.dart';
8
9 late List<CameraDescription> cameras;
10
11 class ScanScreen extends StatefulWidget {
12   const ScanScreen({super.key});
13
14   @override
15   State<ScanScreen> createState() => _ScanScreenState();
16 }
17
18 class _ScanScreenState extends State<ScanScreen> {
19   CameraController? _controller;
20   late Future<void> _initializeControllerFuture;
21
22   @override
23   void initState() {
24     super.initState();
25     _initCamera();
26   }
27
28   void _initCamera() async {
29     cameras = await availableCameras();
30     _controller = CameraController(cameras[0], ResolutionPreset.medium);
31     _initializeControllerFuture = _controller!.initialize();
32     if (mounted) {
33       setState(() {});
34     }
35   }
36
37   @override
38   void dispose() {
39     _controller?.dispose();
40     super.dispose();
41   }
42
43   Future<String> _ocrFromFile(File imageFile) async {
44     final inputImage = InputImage.fromFile(imageFile);
45     final textRecognizer = TextRecognizer(script: TextRecognitionScript.latin);
46     final RecognizedText recognizedText = await textRecognizer.processImage(inputImage);
47     textRecognizer.close();
48     return recognizedText.text;
49   }
50
51   Future<void> _takePicture() async {
52     try {
53       await _initializeControllerFuture;
54
55       if (!mounted) return;
56       ScaffoldMessenger.of(context).showSnackBar(
57         const SnackBar(content: Text('Memproses OCR, mohon tunggu...'), duration: Duration(seconds: 2)),
58       );
59
60       // tambahkan ! karena _controller nullable
61       final XFile image = await _controller!.takePicture();
62
63       final ocrText = await _ocrFromFile(File(image.path));
64
65       if (!mounted) return;
66       Navigator.push(
67         context,
68         MaterialPageRoute(builder: (_) => ResultScreen(ocrText: ocrText)),
69       );
70     } catch (e) {
71       if (!mounted) return;
72       ScaffoldMessenger.of(context).showSnackBar(
73         SnackBar(content: Text('Error saat mengambil/memproses foto: $e')),
74       );
75     }
76   }
77
78   @override
79   Widget build(BuildContext context) {
80     if (_controller == null || !_controller!.value.isInitialized) {
81       return const Scaffold(body: Center(child: CircularProgressIndicator()));
82     }
83
84     return Scaffold(
85       appBar: AppBar(title: const Text('Kamera OCR')),
86       body: Column(
87         children: [
88           Expanded(
89             child: AspectRatio(
90               aspectRatio: _controller!.value.aspectRatio,
91               child: CameraPreview(_controller!),
92             ),
93           ),
94           Padding(
95             padding: const EdgeInsets.all(16.0),
96             child: ElevatedButton.icon(
97               onPressed: _takePicture,
98               icon: const Icon(Icons.camera),
99               label: const Text('Ambil Foto & Scan'),
100             ),
101         ],
102       ),
103     );
104   }
105 }
106
107 }

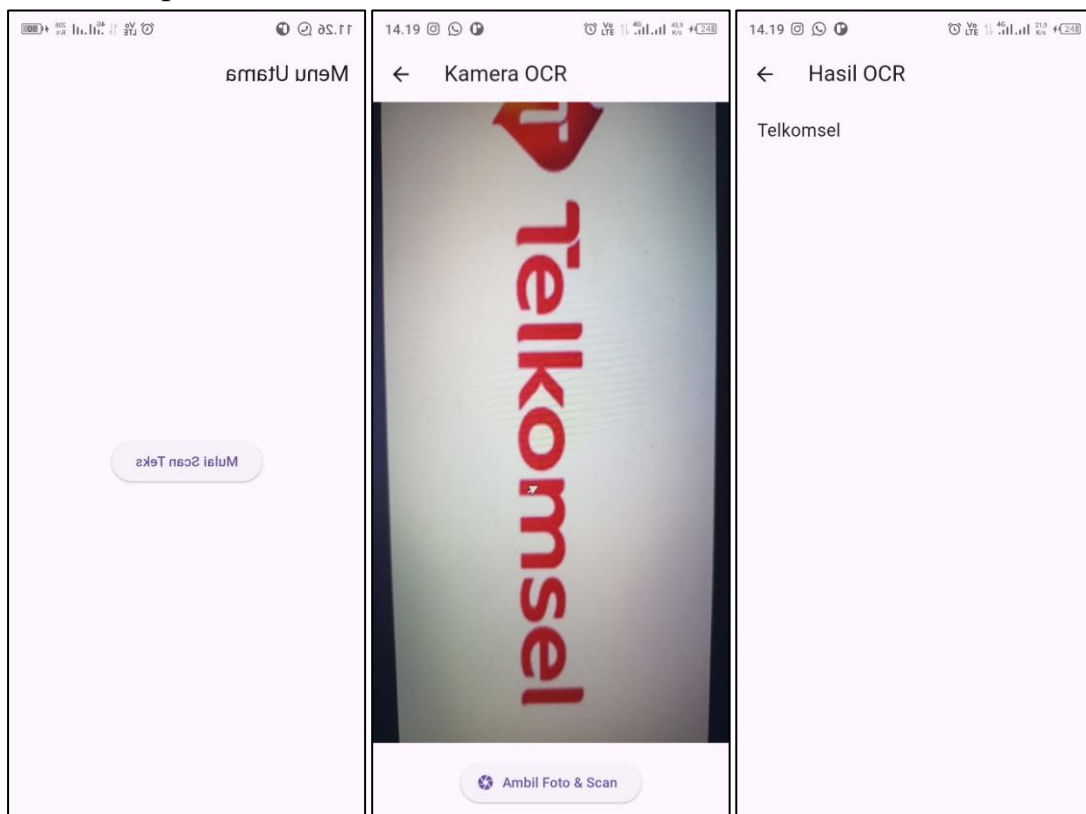
```

```

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 ? 'Tidak ada teks ditemukan.' : ocrText.replaceAll('\n', ' '),
17             style: const TextStyle(fontSize: 18),
18           ),
19         ),
20       ),
21     );
22   }
23 }

```

7. Jalankan aplikasi



Tugas Praktikum: Analisis OCR

a. Apakah semua teks terbaca dengan akurat? Mengapa?

Tidak, tidak semua teks dapat terbaca dengan akurat 100% oleh sistem OCR. Akurasi pengenalan bervariasi dan dipengaruhi oleh beberapa faktor utama, yaitu:

1. Kualitas Gambar Sumber:

- **Resolusi Rendah:** Gambar yang buram, pecah (*pixelated*), atau dipindai dengan DPI rendah akan menurunkan akurasi OCR secara signifikan.
- **Kontras dan Pencahayaan:** Gambar yang terlalu terang (*overexposed*) atau terlalu gelap (*underexposed*) menyebabkan OCR kesulitan membedakan antara karakter dan latar belakang.

2. Jenis dan Kualitas Teks:

- **Tulisan Tangan:** Meskipun teknologi ICR (*Intelligent Character Recognition*) canggih mampu mengatasinya, akurasi tetap jauh lebih rendah dibandingkan teks cetak standar.
- **Font Dekoratif/Tidak Umum:** Teks dengan jenis huruf yang terlalu dekoratif, kecil, atau terlalu padat sulit dikenali dengan tepat.

3. Noise dan Gangguan Visual:

- Keberadaan "noise" seperti titik-titik acak, noda pada dokumen, latar belakang yang ramai (*busy background*), atau bayangan dapat menyebabkan sistem OCR salah menginterpretasikan karakter (misalnya, mengenali noda sebagai tanda baca).

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

Fitur OCR memiliki kegunaan yang luas dalam mempermudah digitalisasi dan pengelolaan data, antara lain:

1. **Konversi Dokumen Fisik ke Digital:** OCR berfungsi untuk mengubah surat, buku, kuitansi, faktur, atau dokumen kertas yang dipindai (*scan*) atau difoto menjadi **teks digital yang dapat disunting** (*editable text*), misalnya di Microsoft Word atau Google Docs. Ini sangat mempermudah pengarsipan.
2. **Pencarian Teks dalam Gambar:** OCR memungkinkan pengguna untuk melakukan pencarian kata atau frasa tertentu di dalam dokumen yang awalnya berbasis gambar (misalnya, di dalam file PDF yang dipindai (*searchable PDF*)).
3. **Ekstraksi Data Cepat (*Data Extraction*):** Digunakan untuk mengambil informasi spesifik secara otomatis seperti nomor telepon, alamat, harga, atau nomor seri dari kartu nama, papan informasi, atau struk belanja hanya dengan memotretnya.

4. **Aksesibilitas dan Terjemahan:** OCR memungkinkan perangkat lunak membaca teks yang ada di dunia nyata (misalnya, rambu jalan atau label produk) untuk diterjemahkan atau dibacakan bagi penyandang disabilitas visual.

c. Sebutkan 2 contoh aplikasi nyata yang menggunakan OCR!

Dua contoh aplikasi populer yang memanfaatkan teknologi OCR adalah:

1. **Google Lens / Google Foto (Google Search)**

- **Kegunaan:** Memungkinkan pengguna untuk mengambil foto teks dari dunia nyata (misalnya, plang nama atau menu) dan langsung **menyalin teks tersebut** ke ponsel, **mencari informasi** terkait teks di Google, atau **menerjemahkannya** secara *real-time*.

2. **CamScanner (atau aplikasi pemindai dokumen sejenis)**

- **Kegunaan:** Digunakan secara luas untuk memindai dokumen kertas menggunakan kamera ponsel. Aplikasi ini dapat mengubah gambar hasil pindaian menjadi file **PDF yang dapat dicari (*searchable PDF*)** atau langsung mengekstrak teks di dalamnya agar dapat disalin dan diedit.