JOBSHEET – APLIKASI OCR
SEDERHANA DENGAN FLUTTER

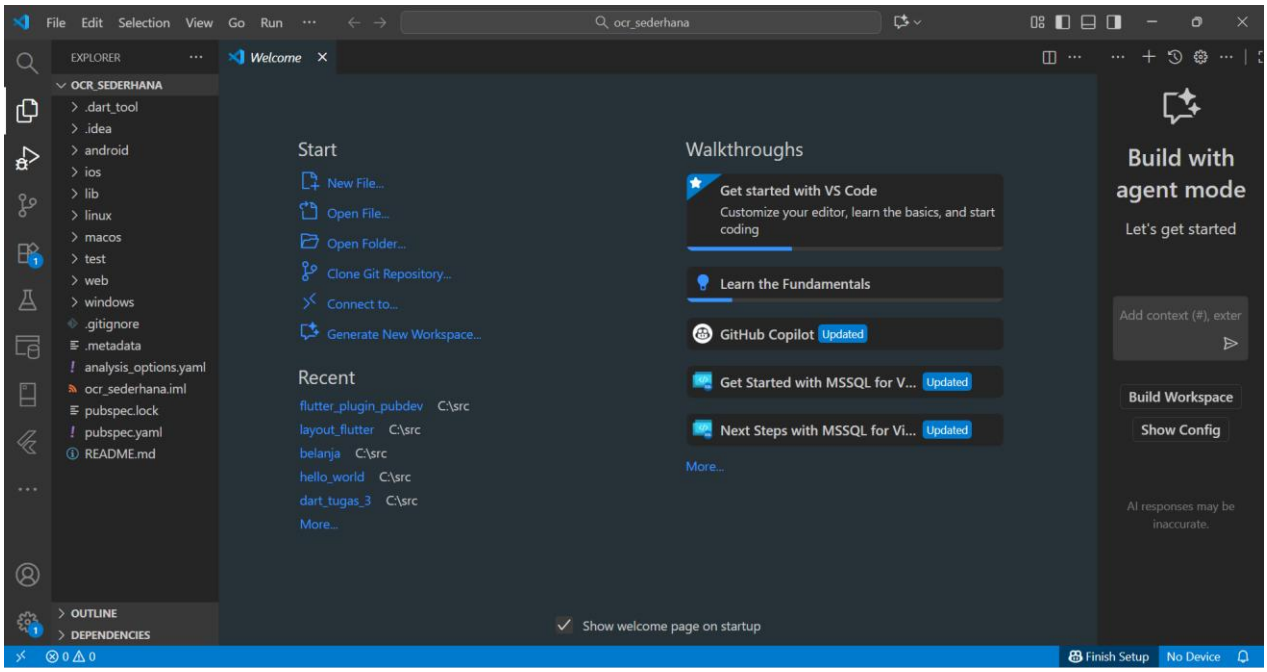
1. IDENTITAS PRAKTIKAN

Komponen	Isi
Nama	Deanissa Sherly Sabilla
Kelas / NIM	SIB 3C/2341760187
Tanggal	21 Oktober 2025

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

2. LANGKAH KERJA

2.1. Langkah 1: Buat Proyek Baru



2.2. Langkah 2: Tambahkan Plugin

Buka file pubspec.yaml, lalu tambahkan dependensi:

```
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
```

file, lalu jalankan:

```
flutter pub get
```

```
PS C:\src\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)
  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Try `flutter pub outdated` for more information.
PS C:\src\ocr_sederhana>
```

2.3. Langkah 3: Tambahkan Izin Kamera (Android)

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

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

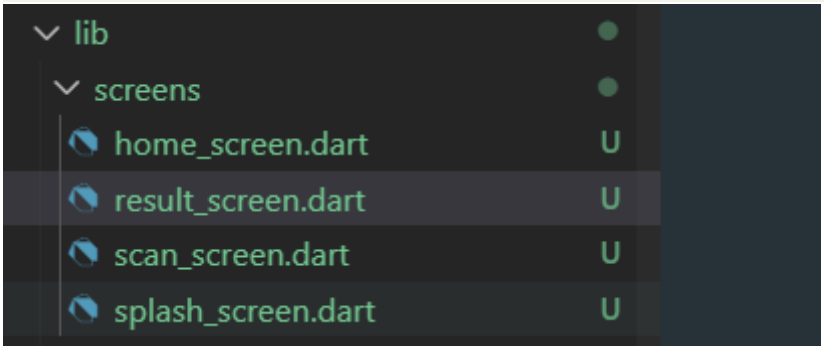
Tambahkan baris berikut di dalam tag <manifest>, sebelum <application>:

```
android > app > src > main > AndroidManifest.xml
1   <manifest xmlns:android="http://schemas.android.com/apk/res/android">
2   <!--menambahkan baris izin kamera-->
3   <uses-permission android:name="android.permission.CAMERA" />
4       <application
```

2.4. Langkah 4: Buat Struktur Folder

Di dalam folder lib/, buat struktur berikut:

```
lib/
          main.dart
          screens/
               splash_screen.dart
               home_screen.dart
               scan_screen.dart
               result_screen.dart
```

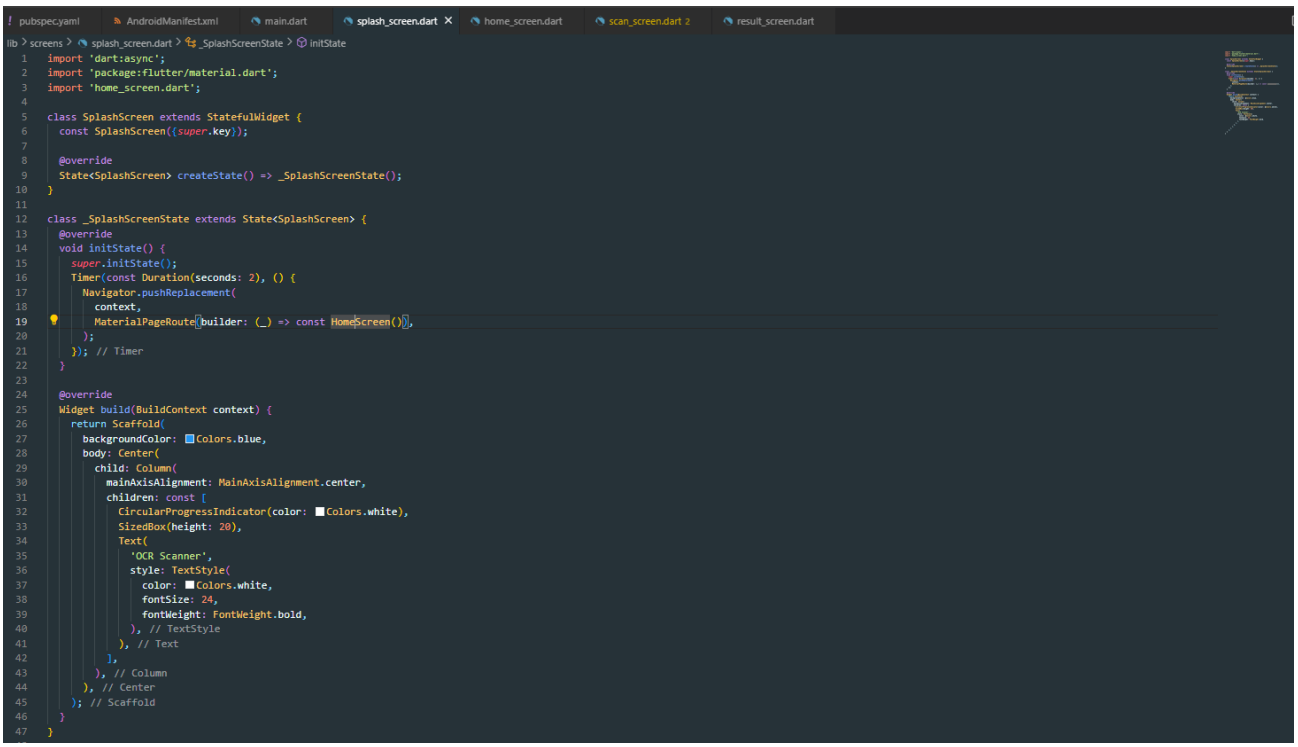


3. KODE PROGRAM

3.1. File: lib/main.dart

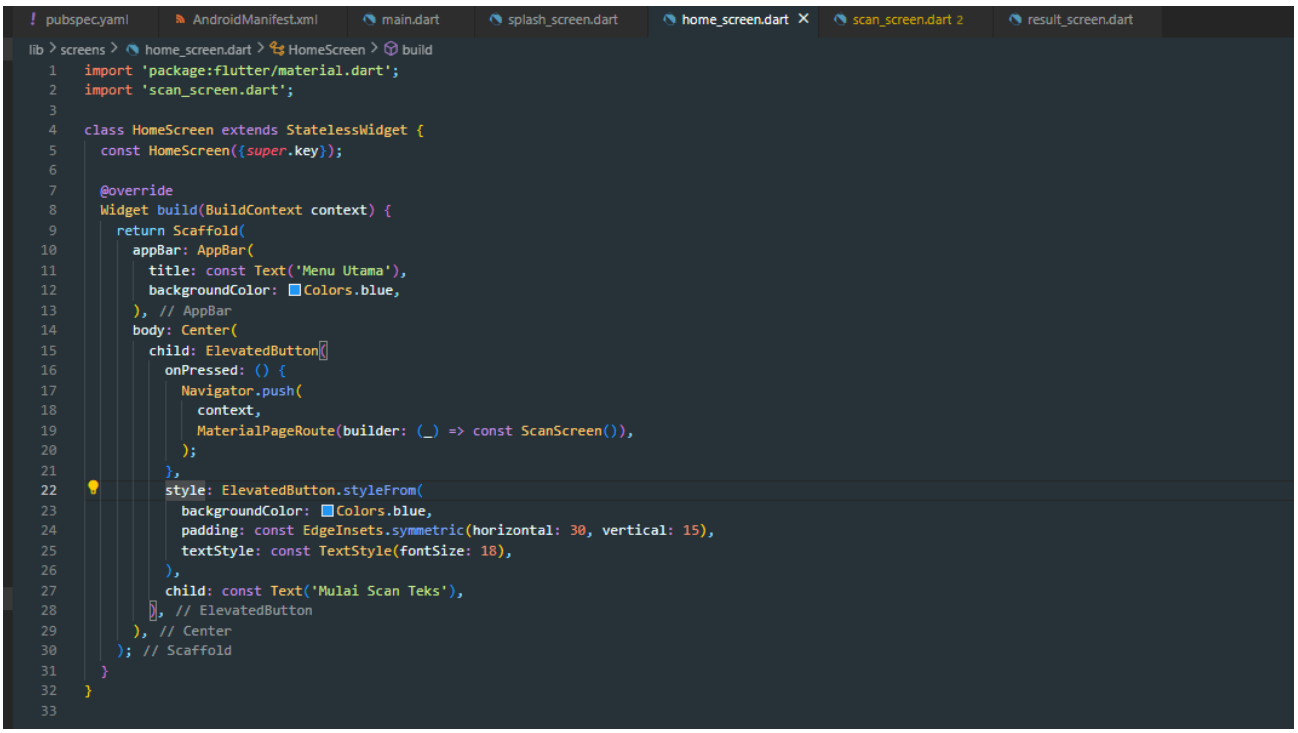
```dart
import 'package:flutter/material.dart';
import 'screens/splash_screen.dart';

Run | Debug | Profile
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,
 ); // MaterialApp
 }
 }

```

3.2. File: lib/screens/splash screen.dart



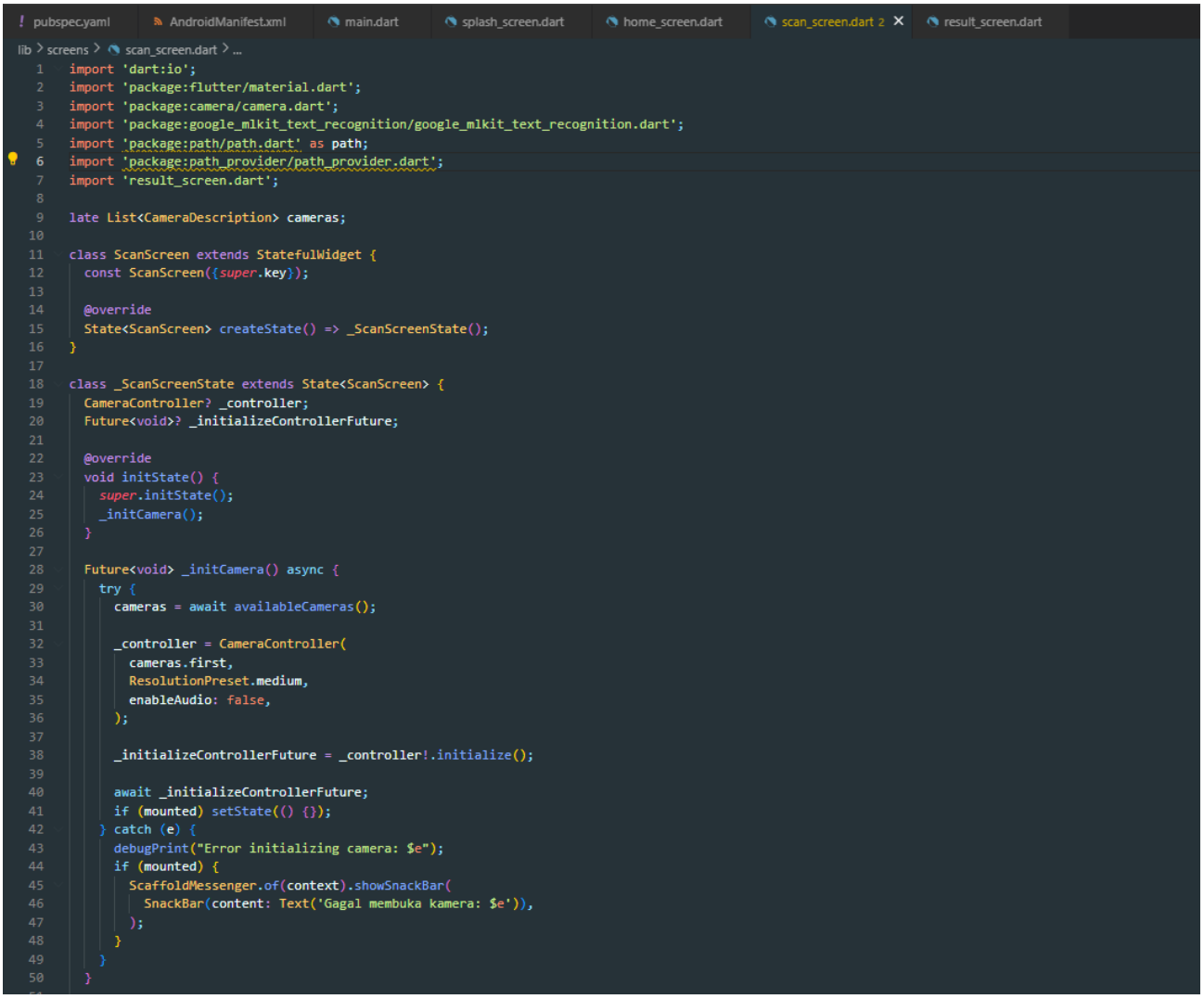
```dart
import 'dart:async';
import 'package:flutter/material.dart';
import 'home_screen.dart';

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()),
      );
    }); // Timer
  }

  @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,
                fontWeight: FontWeight.bold,
              ), // TextStyle
            ), // Text
          ],
        ), // Column
      ), // Center
    ); // Scaffold
  }
}
```

3.3. File: lib/screens/home screen.dart



```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'),
        backgroundColor: Colors.blue,
      ), // AppBar
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            Navigator.push(
              context,
              MaterialPageRoute(builder: (_) => const ScanScreen()),
            );
          },
          style: ElevatedButton.styleFrom(
            backgroundColor: Colors.blue,
            padding: const EdgeInsets.symmetric(horizontal: 30, vertical: 15),
            textStyle: const TextStyle(fontSize: 18),
          ),
          child: const Text('Mulai Scan Teks'),
        ), // ElevatedButton
      ), // Center
    ); // Scaffold
  }
}
```

3.4. File: lib/screens/scan screen.dart



```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> {
  CameraController? _controller;
  Future<void>? _initializeControllerFuture;

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

  Future<void> _initCamera() async {
    try {
      cameras = await availableCameras();

      _controller = CameraController(
        cameras.first,
        ResolutionPreset.medium,
        enableAudio: false,
      );

      _initializeControllerFuture = _controller!.initialize();

      await _initializeControllerFuture;
      if (mounted) setState(() {});
    } catch (e) {
      debugPrint("Error initializing camera: $e");
      if (mounted) {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text('Gagal membuka kamera: $e')),
        );
      }
    }
  }
```

```dart
  @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 {
    if (_controller == null || !_controller!.value.isInitialized) {
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text('Kamera belum siap')),
      );
      return;
    }

    try {
      await _initializeControllerFuture;

      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) {
      debugPrint('Error saat mengambil foto: $e');
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Terjadi kesalahan: $e')),
      );
    }
  }

  @override
  Widget build(BuildContext context) {
    if (_controller == null) {
      return const Scaffold(
        body: Center(child: CircularProgressIndicator()),
      ); // Scaffold
    }
```

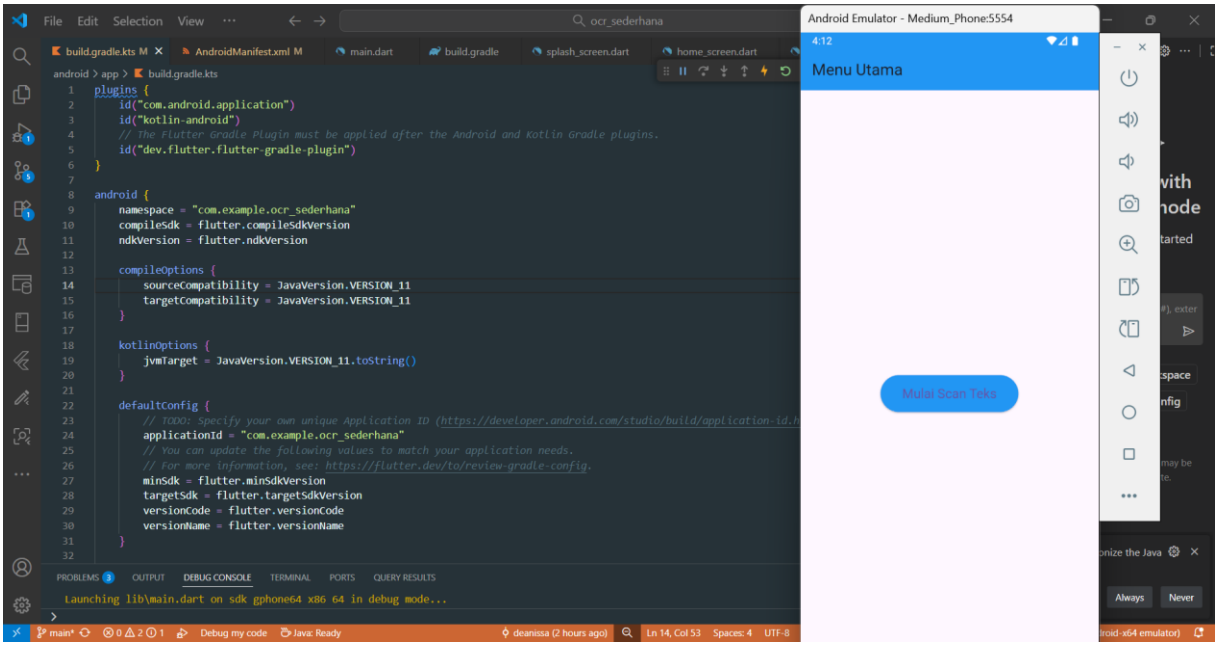
```dart

    return Scaffold(
      appBar: AppBar(title: const Text('Kamera OCR')),
      body: FutureBuilder<void>(
        future: _initializeControllerFuture,
        builder: (context, snapshot) {
          if (snapshot.connectionState == ConnectionState.done) {
            return Column(
              children: [
                Expanded(
                  child: AspectRatio(
                    aspectRatio: _controller!.value.aspectRatio,
                    child: CameraPreview(_controller!),
                  ), // AspectRatio
                ), // Expanded
                Padding(
                  padding: const EdgeInsets.all(16.0),
                  child: ElevatedButton.icon(
                    onPressed: _takePicture,
                    icon: const Icon(Icons.camera),
                    label: const Text('Ambil Foto & Scan'),
                  ), // ElevatedButton.icon
                ), // Padding
              ],
            ); // Column
          } else {
            return const Center(child: CircularProgressIndicator());
          }
        },
      ), // FutureBuilder
    ); // Scaffold
  }
}
```

3.5. File: lib/screens/result screen.dart

```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'),
        backgroundColor: Colors.blue,
      ), // AppBar
      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,
              height: 1.5,
            ), // TextStyle
          ), // SelectableText
        ), // SingleChildScrollView
      ), // Padding
    ); // Scaffold
  }
}
```

4. TUGAS PRAKTIKUM

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:

   a. Apakah semua teks terbaca dengan akurat? Mengapa?
      Jawab:
      Tidak selalu semua teks terbaca dengan akurat. Akurasi hasil **OCR (Optical Character Recognition)** bergantung pada beberapa faktor, seperti:
      - **Kualitas gambar** (resolusi, pencahayaan, dan ketajaman).
      - **Jenis dan ukuran font** pada teks.
      - **Tingkat kebersihan latar belakang** (jika gambar buram, miring, atau terdapat bayangan).
      - **Bahasa dan karakter khusus** yang didukung oleh mesin OCR.

   b. Apa kegunaan fitur OCR dalam kehidupan sehari-hari?
      Jawab:
      Fitur OCR berguna untuk **mengubah teks dari gambar atau dokumen cetak menjadi teks digital** yang bisa diedit, dicari, dan disimpan.

   c. Sebutkan 2 contoh aplikasi nyata yang menggunakan OCR!
      Jawab:
      Google Lens dan Microsoft Office Lens / Adobe Scan