

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE**

JOBSHEET 5

APLIKASI PERTAMA DAN WIDGET DASAR FLUTTER



Oleh:

NIMAS SEPTIANDINI

2341760087

SIB-3C

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG


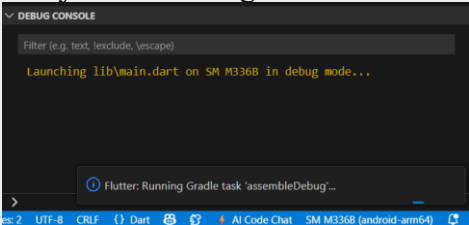

2025/2026

Repositori Github: <https://github.com/septyandini921/flutter-fundamental-part1.git>

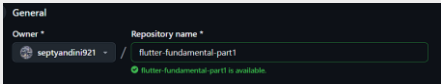
Praktikum 1: Membuat Project Flutter Baru


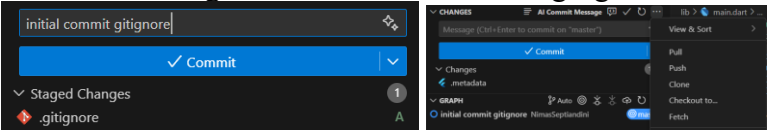
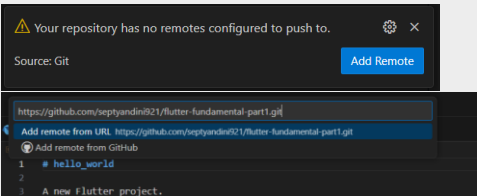
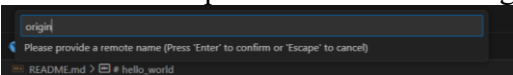
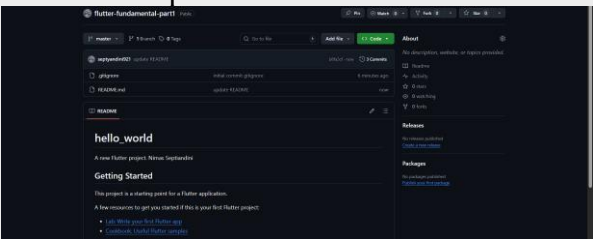
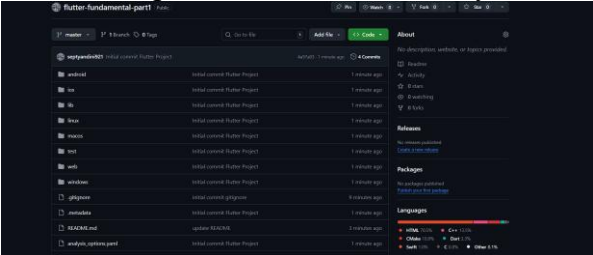
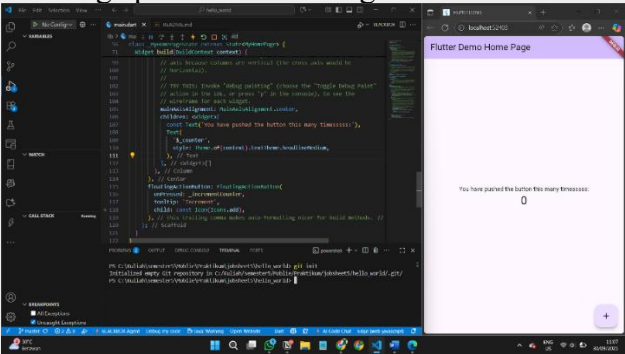
Langkah	Praktikum
1	Buka VS Code, lalu tekan tombol Ctrl + Shift + P maka akan tampil <i>Command Palette</i> , lalu ketik Flutter . Pilih New Application Project .
2	Kemudian buat folder sesuai style laporan praktikum yang Anda pilih. Disarankan pada folder dokumen atau desktop atau alamat folder lain yang tidak terlalu dalam atau panjang. Lalu pilih Select a folder to create the project in .
3	Buat nama project flutter hello_world seperti berikut, lalu tekan Enter . Tunggu hingga proses pembuatan project baru selesai.
4	Jika sudah selesai proses pembuatan project baru, pastikan tampilan seperti berikut. Pesan akan tampil berupa "Your Flutter Project is ready!" artinya Anda telah berhasil membuat project Flutter baru.

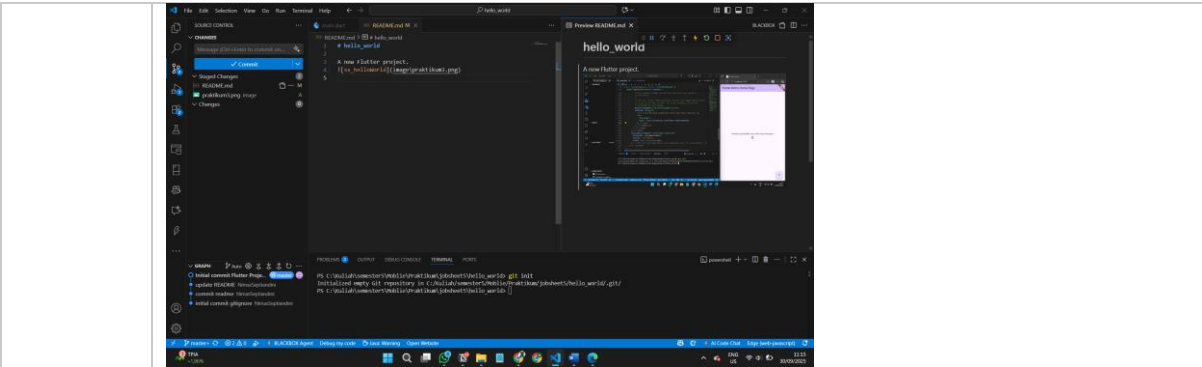
Praktikum 2: menjalankan aplikasi ke perangkat fisik (device Android atau iOS)

Langkah	Praktikum
1	<p>Mengaktifkan mode debug pada device android</p> 
2	Melakukan >flutter: select device
3	<p>Menjalankan debug</p> 
4	<p>Aplikasi dapat dijalankan di perangkat fisik android</p> 

Praktikum 3: Membuat Repository GitHub dan Laporan Praktikum

Langkah	Praktikum
1	<p>Membuat repository baru dengan nama "flutter-fundamental-part1"</p> 
2	<p>Melakukan inisialisai git init pada terminal</p> <pre>PS C:\Kuliah\semester5\Moblie\Praktikum\jobsheet5\hello_world> git init Initialized empty Git repository in C:\Kuliah\semester5\Moblie\Praktikum\jobsheet5\hello_world\.git\ PS C:\Kuliah\semester5\Moblie\Praktikum\jobsheet5\hello_world></pre>
3	Menambahkan .gitignore pada staging area untuk initial commit

	
4	<p>Memberi message dan melakukan commit gitignore kemudian melakukan push</p> 
5	<p>Add remote</p> 
6	<p>Memberi nama pada remote name dengan nama "origin"</p> 
7	<p>Commit and push README.md</p> 
8	<p>Melakukan push folder dan file lainnya</p> 
9	<p>Running aplikasi flutter dengan web browser</p> 
10	<p>Menambahkan dokumentasi gambar pada README</p>

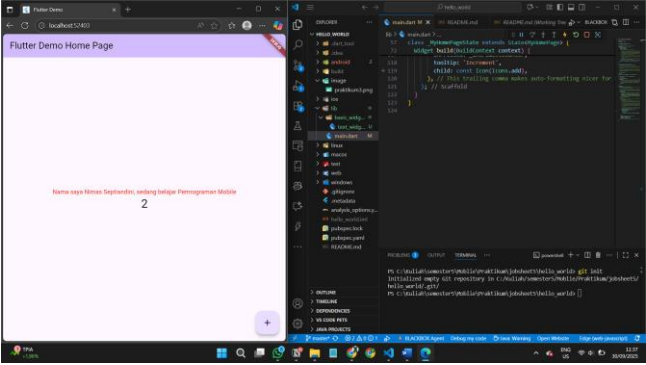
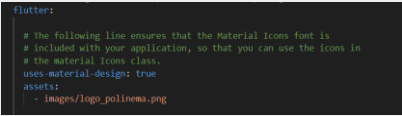



11

Hasil akhir


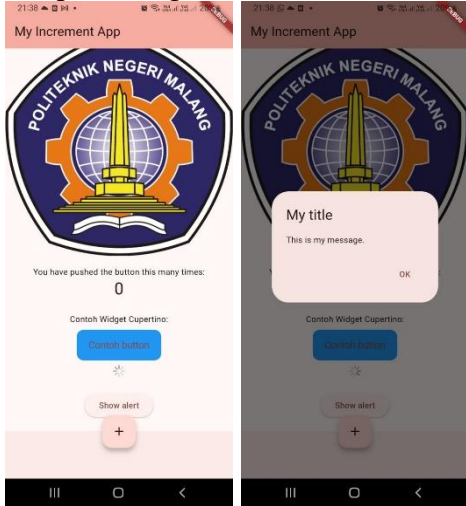

A screenshot of a GitHub repository for a project named 'hello_world'. The README file is displayed, containing instructions for running the project. The repository is owned by 'pudiprati' and is a Flutter project. The README includes a 'Flutter Demo Home Page' image and a list of dependencies. The right sidebar shows 'Based on your tech stack' recommendations for tools like 'KLSA Generic generator', 'Inkify using Docker image', and 'MSBuild based projects'.

Praktikum 4: Menerapkan Widget Dasar

Langkah	Praktikum
1	Membuat folder baru basic_widget dalam folder lib, kemudian membuat file baru dengan nama text_widget.dart
2	<p>Import file text_widget.dart ke main.dart</p> 
3	<p>Membuat file image_widget.dart dalam folder basic_widgets dan menyesuaikan pubspec.yaml lalu mengimport file image_widget dalam main.dart</p>  

Praktikum 5: Menerapkan Widget Material Design dan iOS Cupertino

Langkah	Praktikum
1	<p>Buat file di basic_widgets > loading_cupertino.dart. Import stateless widget dari material dan cupertino.</p>  <pre> lib > basic_widget > loading_cupertino.dart > loading_cupertino.dart 1 import 'package:flutter/material.dart'; 2 import 'package:flutter/cupertino.dart'; 3 4 class LoadingCupertino extends StatelessWidget { 5 const LoadingCupertino({super.key}); 6 7 @override 8 Widget build(BuildContext context) { 9 return MaterialApp(10 home: Container(11 margin: const EdgeInsets.only(top: 30), 12 color: Colors.white, 13 child: Column(14 children: <widget> 15 CupertinoButton(16 child: const Text("Contoh button"), 17 onPressed: () {}, 18), // CupertinoActivityIndicator(), 19], // <widget> 20), // Column 21), // Container 22); // MaterialApp 23 } 24 } 25 </pre>
2	<p>Buat file di basic_widgets > fab_widget.dart. Import stateless widget dari material.</p>  <pre> lib > basic_widget > fab_widget.dart > fab_widget.dart 1 import 'package:flutter/material.dart'; 2 3 class FabWidget extends StatelessWidget { 4 const FabWidget({super.key}); 5 6 @override 7 Widget build(BuildContext context) { 8 return MaterialApp(9 home: Scaffold(10 floatingActionButton: FloatingActionButton(11 onPressed: () { 12 // aksi ketika tombol ditekan 13 debugPrint("FAB ditekan!"); 14 }, 15 child: const Icon(Icons.thumbUp), 16 backgroundColor: Colors.pink, 17), // FloatingActionButton 18), // Scaffold 19); // MaterialApp 20 } 21 } 22 </pre>
3	<p>Scaffold widget digunakan untuk mengatur tata letak sesuai dengan material design. Ubah isi kode main.dart</p>  <pre> lib > main.dart > main.dart 1 import 'package:flutter/material.dart'; 2 import 'package:flutter/cupertino.dart'; 3 import 'basic_widget/loading_widget.dart'; 4 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: 'Flutter Demo', 16 theme: ThemeData(17 // This is the default theme, you can change it to whatever you want. 18 colorScheme: ColorScheme.fromSeed(seedColor: Colors.red), 19 useMaterial3: true, 20), // ThemeData 21 // This is the main content of the app, you can change it to whatever you want. 22 home: const MyIncrementApp(), 23); // MaterialApp 24 } 25 } 26 </pre>

	
4	<p>Dialog widget pada flutter memiliki dua jenis dialog yaitu AlertDialog dan SimpleDialog. Ubah isi kode main.dart</p> 
5	<p>Menambahkan input textField</p> 
6	<p>Menambahkan input date time picker</p>

