

LAPORAN PRAKTIKUM

JOBSHEET FLUTTER 1



Oleh:

Afgan Galih Fauz Amjad Amadinah

2341760004

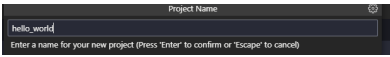
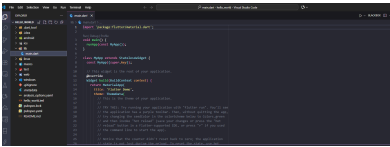
SIB 3C

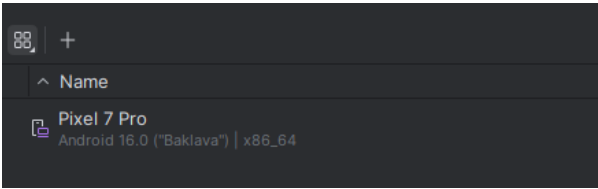
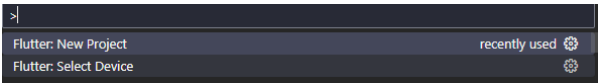
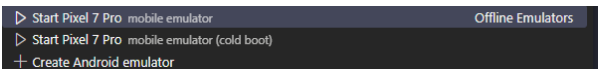
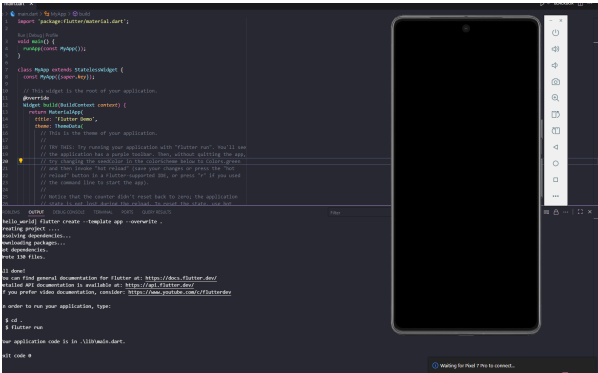
01

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

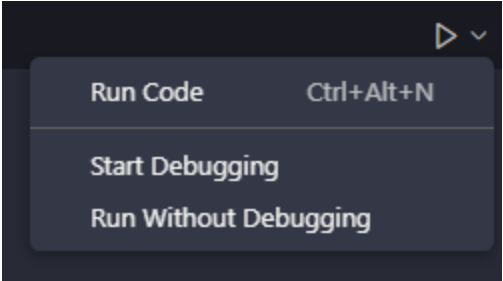
POLITEKNIK NEGERI MALANG

TAHUN AJARAN 2025/2026

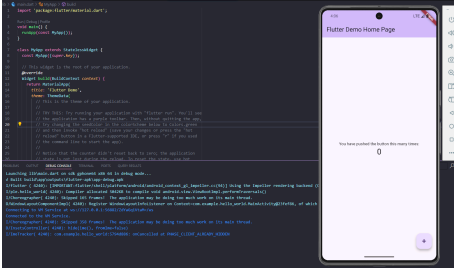
Praktikum 1: Membuat Project Flutter Baru	
Langkah	Dokumentasi
Buka VSCode, tekan Ctrl + Shift + P dan membuat projek flutter baru	
<i>Your Flutter Project is Ready!</i>	

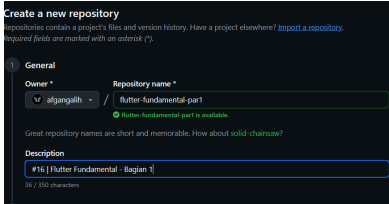
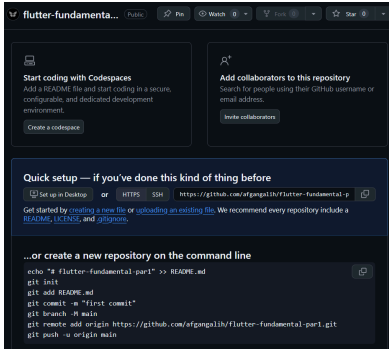
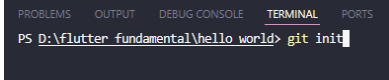
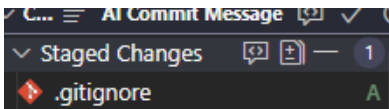
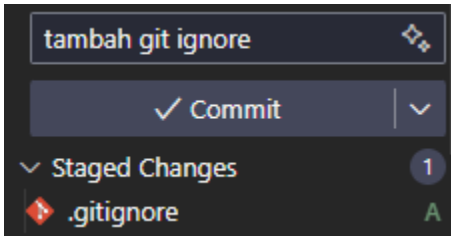
Praktikum 2: Menghubungkan Perangkat Android atau Emulator	
Langkah	Gambar
Pergi ke Android Studio untuk membuat Virtual Device	
Ctrl + Shift + P, kemudian pilih Select Devices untuk memilih Virtual Device yg sudah dibuat sebelumnya	
Dalam hal ini, saya memilih Google Pixel 7 Pro	
Menunggu Virtual Device tersambung	

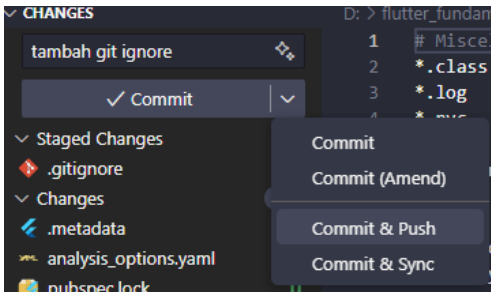
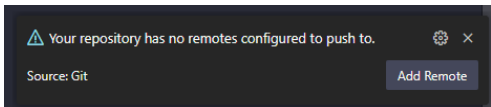
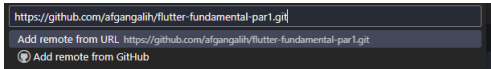
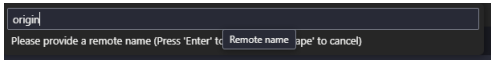
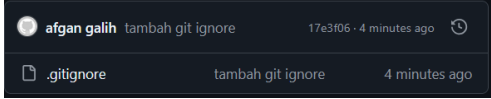
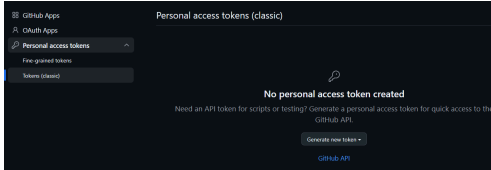
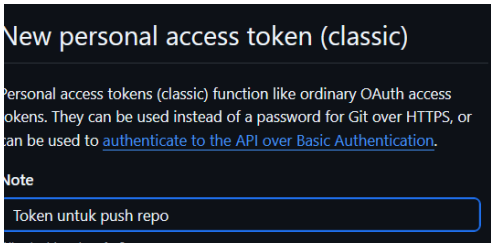
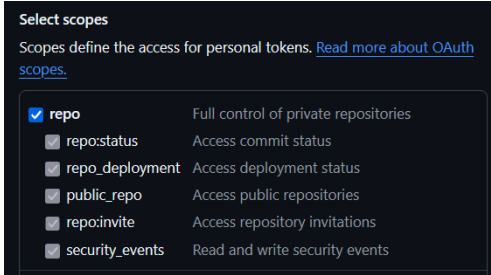
Setelah Virtual Device Muncul, pilih **Start Debugging**



Projek flutter pada Emulator sudah muncul



Praktikum Ke - 3: Membuat Repository Github dan Laporan Praktikum	
Langkah	Dokumentasi
Login ke github, buat repository baru dengan nama flutter-fundamental-part1	
Create Repository	
Kembali ke VS code, project flutter hello_world, buka terminal pada menu Terminal > New Terminal . Lalu ketik perintah berikut untuk inisialisasi git pada project	
Pilih menu Source Control di bagian kiri, lalu lakukan stages (+) pada file .gitignore untuk mengunggah file pertama ke repository GitHub.	
Beri pesan commit " tambah gitignore " lalu klik Commit (✓)	

<p>Lakukan push dengan klik bagian menu titik tiga > Push</p>	
<p>Di pojok kanan bawah akan tampil seperti gambar berikut. Klik "Add Remote"</p>	
<p>Salin tautan repository Anda dari browser ke bagian ini, lalu klik Add remote</p>	
<p>Setelah berhasil, tulis remote name dengan "origin"</p>	
	
<p>Profile Developer Setting Token Classic</p>	
<p>Membuat note</p>	
<p>Centang repo agar bisa melakukan push dan pull repository</p>	

Salin token

Tokens you have generated that can be used to access the [GitHub API](#).



Make sure to copy your personal access token now. You won't be able to see it again!



ghp_sVn6I5FePp5DwHCE0jy01aUZzYgNY0CJjPP



Delete

Staging & Push README.md

```
PS D:\Flutter_fundamental\hello_world> git add README.md
>> git commit -m "Menambahkan file README.md"
>>
[main 968e115] Menambahkan file README.md
 1 file changed, 16 insertions(+)
 create mode 100644 README.md
PS D:\Flutter_fundamental\hello_world> git push origin main
>>
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 597 bytes | 597.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/afgangelih/flutter-fundamental-par1.git
 17e3f06..968e115  main -> main
PS D:\Flutter_fundamental\hello_world>
```



afgan galih Menambahkan file READM... 968e115 · 1 minute ago



.gitignore

tambah git ignore

24 minutes ago



README.md

Menambahkan file R...

1 minute ago

Lakukan push juga untuk semua file lainnya dengan pilih **Stage All Changes**. Beri pesan commit "project hello_world". Maka akan tampil di repository GitHub Anda seperti berikut.

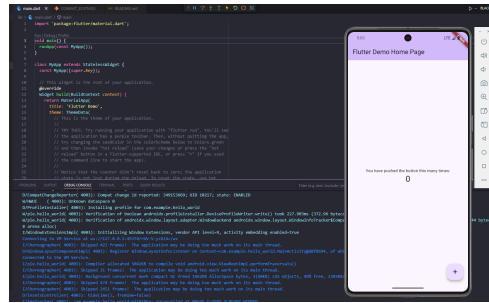
flutter-fundamenta... Public Pin Watch 0

main Go to file + <> Code

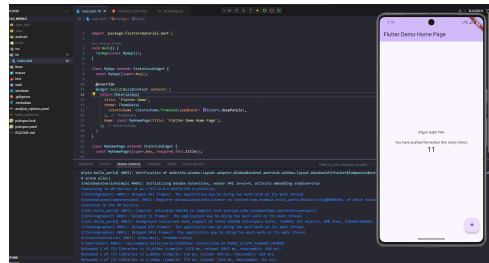
afgan galih project hello world 4388c4a · now

android	project hello world	now
ios	project hello world	now
lib	project hello world	now
linux	project hello world	now
macos	project hello world	now
test	project hello world	now
web	project hello world	now
windows	project hello world	now
.gitignore	tambah git ignore	26 minutes ago
.metadata	project hello world	now
README.md	Menambahkan file R...	3 minutes ago
analysis_options.yaml	project hello world	now
pubspec.lock	project hello world	now

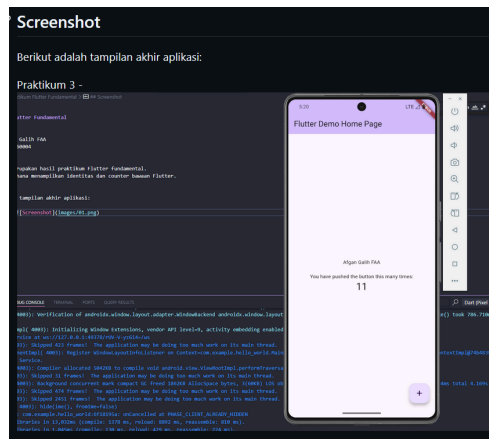
Running project **hello_world**

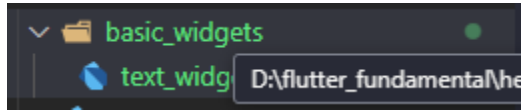

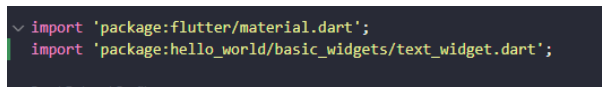
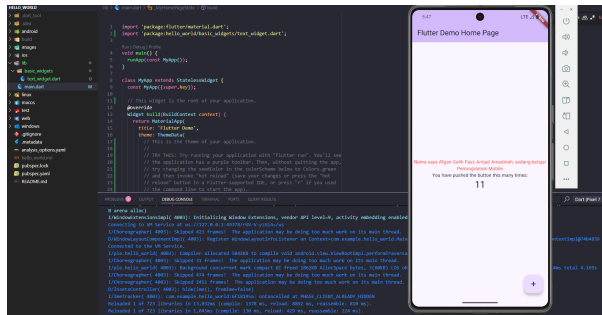
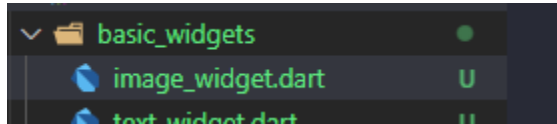
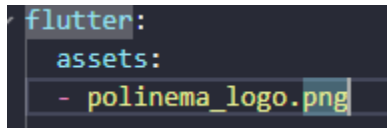
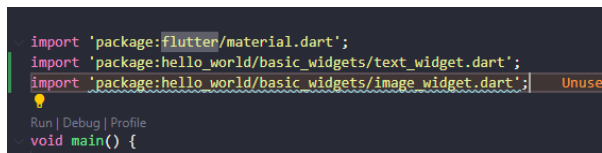


Final

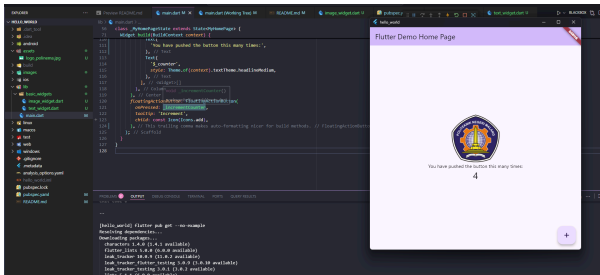


Preview github README.md



Praktikum 4: Menerapkan Widget Dasar	
Langkah	Dokumentasi
Langkah 1	
Buat folder baru basic_widgets di dalam folder lib . Kemudian buat file baru di dalam basic_widgets dengan nama text_widget.dart	
Kode program text_widget.dart	
Import file text_widget.dart	
Setelah save and hot reload	
Langkah 2	
Buat file image_widget.dart	
Penyesuaian asset pada file pubspec.yaml	
Import file image_widget.dart	

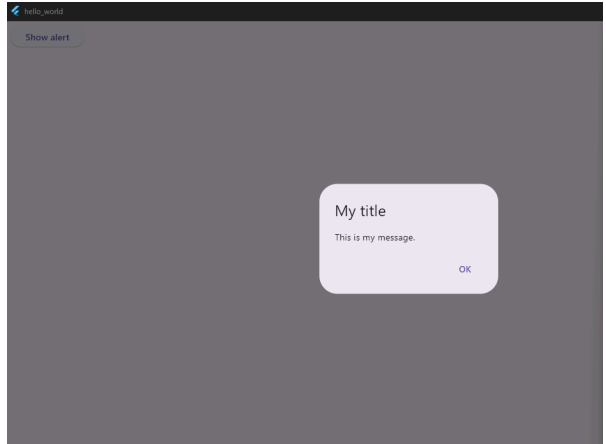

Menampilkan Widget Image



Praktikum 5: Menerapkan Widget Material Design dan iOS Cupertino

Langkah	Dokumentasi
Langkah 1	
Buat file loading_cupertino.dart	 <pre>9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 return MaterialApp(home: Container(margin: const EdgeInsets.only(top: 30), color: Colors.white, child: Column(children: <Widget>[CupertinoButton(child: const Text("Contoh button"), onPressed: () {},), // CupertinoButton const CupertinoActivityIndicator(),], // <Widget>[]), // Column), // Container); // MaterialApp }</pre>
Langkah 2	
Buat file fab_widget.dart	 <pre>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 appBar: AppBar(title: const Text("FAB Example")), 11 floatingActionButton: Builder(12 builder: (context) => FloatingActionButton(13 onPressed: () { 14 ScaffoldMessenger.of(context).showSnackBar(15 const SnackBar(content: Text("Tombol FAB ditekan!")); 16); 17 }, 18 child: const Icon(Icons.thumb_up), // The 'child' argument 19 backgroundColor: Colors.pink, 20), // FloatingActionButton 21), // Builder 22 body: const Center(23 child: Text("Halo ini contoh Floating Action Button"), 24), 25), 26); 27 }</pre>
Langkah 3	
Ubah isi kode main.dart Dialog widget pada flutter memiliki dua jenis dialog yaitu AlertDialog dan SimpleDialog.	 <pre>import 'package:flutter/material.dart'; Run Debug Profile void main() { runApp(const MyApp()); } class MyApp extends StatelessWidget { const MyApp({Key? key}) : super(key: key); // Parameter 'key' could be null. // This widget is the root of your application. @override Widget build(BuildContext context) { return MaterialApp(title: 'Flutter Demo', theme: ThemeData(primarySwatch: Colors.red,), // ThemeData home: const MyHomePage(title: 'My Increment App'),); // MaterialApp } } class MyHomePage extends StatefulWidget { const MyHomePage({Key? key, required this.title}) : super(key: key); }</pre>
Langkah 4	

Praktikum 5: Menerapkan Widget Material Design dan iOS Cupertino

Langkah	Dokumentasi
Langkah 1	
Dialog widget pada flutter memiliki	
Langkah 5	
lutter menyediakan widget yang dapat menerima input dari pengguna aplikasi yaitu antara lain Checkbox, Date and Time Pickers, Radio Button, Slider, Switch, TextField. Contoh penggunaan TextField widget adalah sebagai berikut:	<pre>class MyApp extends StatelessWidget { const MyApp({Key? key}) : super(key: key); @override Widget build(BuildContext context) { return MaterialApp(home: Scaffold(appBar: AppBar(title: const Text("Contoh TextField")), body: const TextField(obscureText: false, decoration: InputDecoration(border: OutlineInputBorder(), labelText: 'Nama',),),),); } }</pre> 
Langkah 6	

Praktikum 5: Menerapkan Widget Material Design dan iOS Cupertino

Langkah	Dokumentasi
Langkah 1	
Implementasi date & pickers	<pre>import 'dart:async'; import 'package:flutter/material.dart'; Run Debug Profile void main() => runApp(const MyApp()); class MyApp extends StatelessWidget { const MyApp({Key? key}) : super(key: key); // Parameter 'key' could be @override Widget build(BuildContext context) { return const MaterialApp(title: 'Contoh Date Picker', home: MyHomePage(title: 'Contoh Date Picker'),); // MaterialApp } } class MyHomePage extends StatefulWidget { const MyHomePage({Key? key, required this.title}) : super(key: key); final String title; @override</pre> 