

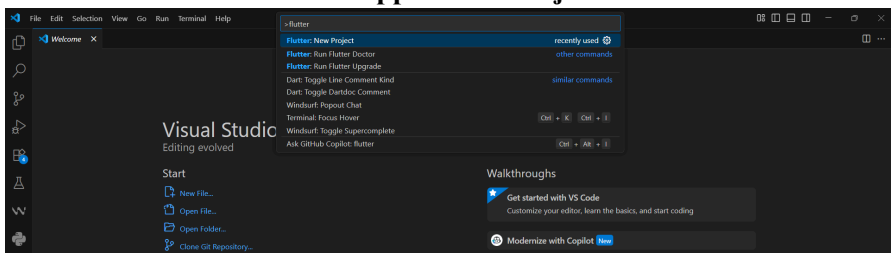
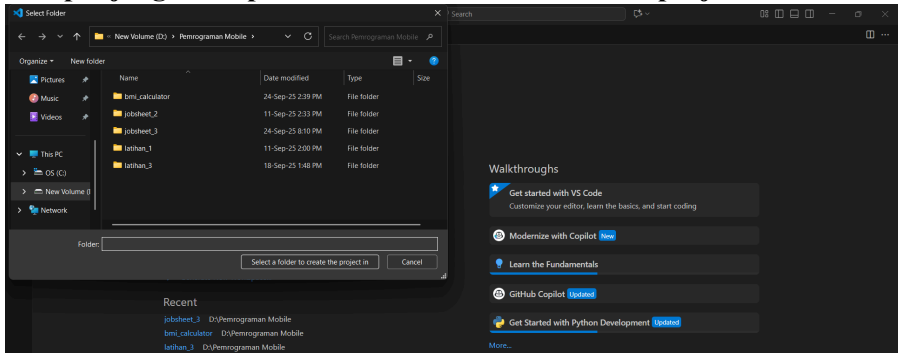


Mata Kuliah : Pemrograman Mobile  
Program Studi : D4 – Sistem Informasi Bisnis  
Semester : 5

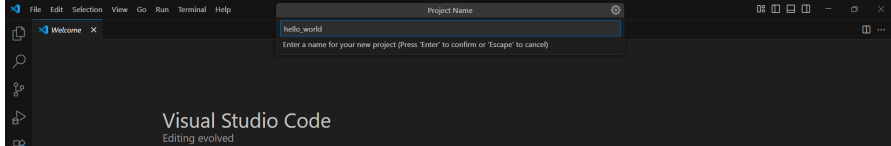
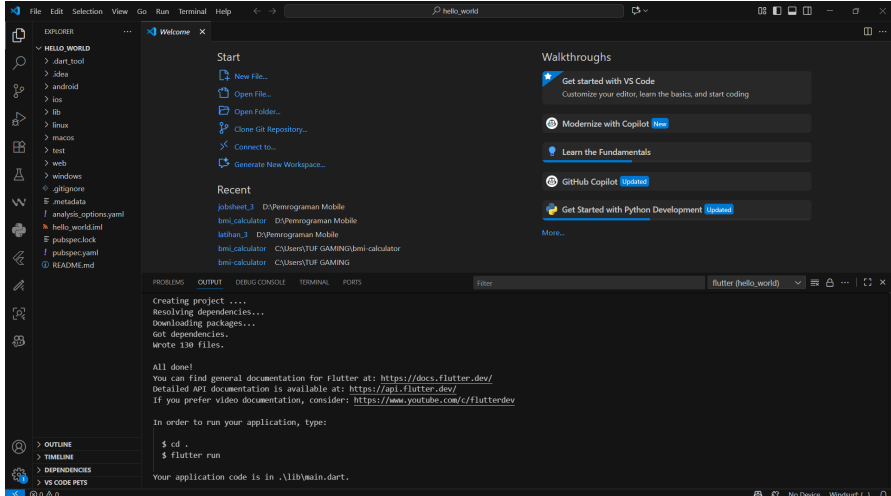
Kelas : SIB - 3D  
NIM : 2341760112  
Nama : Titania Aurellia Putri Dwiendra  
Jobsheet Ke- : 4  
Link Github : <https://github.com/titaniaaurelliaa/flutter-fundamental-part1>

## Laporan Jobsheet

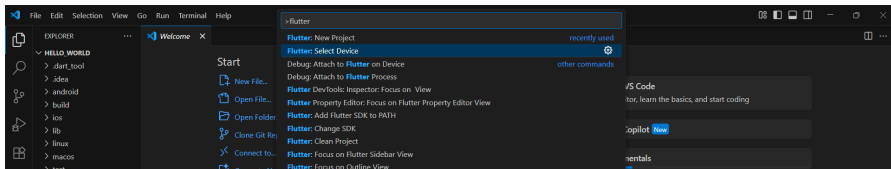
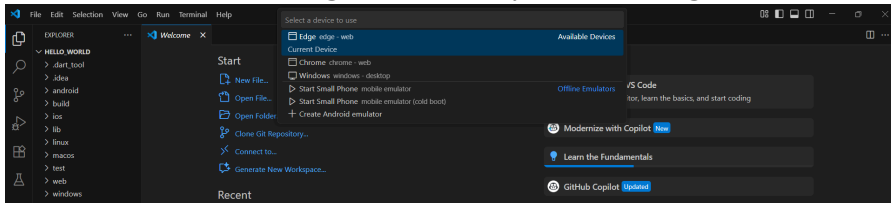
### Praktikum Ke-1 Membuat Project Flutter Baru

Langkah	Jawaban/Deskripsi
1	<b>Buka VS Code, lalu tekan tombol Ctrl + Shift + P maka akan tampil Command Palette, lalu ketik Flutter. Pilih New Application Project.</b> 
2	<b>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.</b> 
3	<b>Buat nama project flutter hello_world seperti berikut, lalu tekan Enter. Tunggu hingga proses pembuatan project baru selesai.</b>



	
4	<p><b>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.</b></p> 

## Praktikum Ke-2 Menghubungkan Perangkat Android atau Emulator

Langkah	Jawaban/Deskripsi
1	<p><b>Tekan tombol search di VS Code, lalu muncul Command Palette, lalu ketik "&gt;Flutter: Select Device"</b></p> 
2	<p><b>Pilih emulator sesuai keinginan, disini saya memilih edge</b></p> 
3	<p><b>Setelah memilih dan mengklik Edge, klik run untuk menjalankannya</b></p>

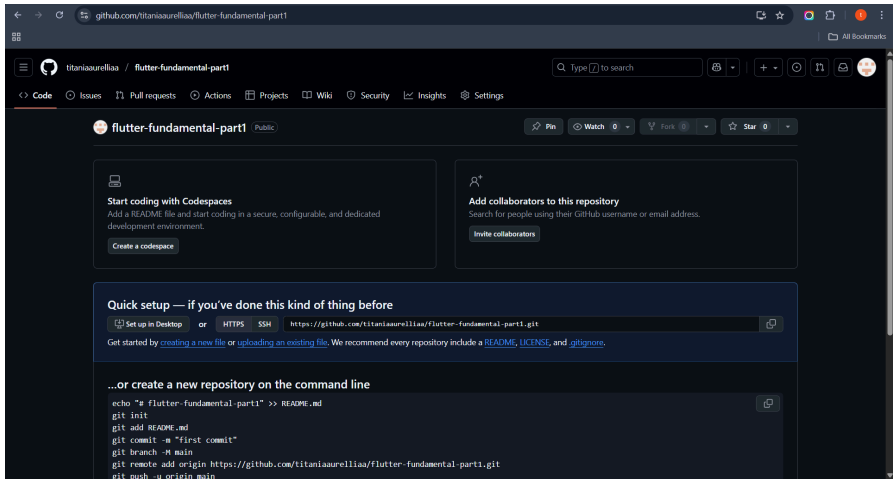
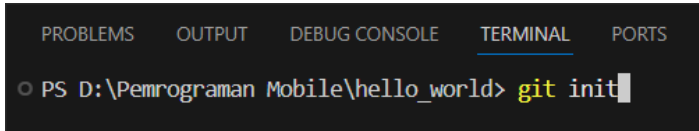
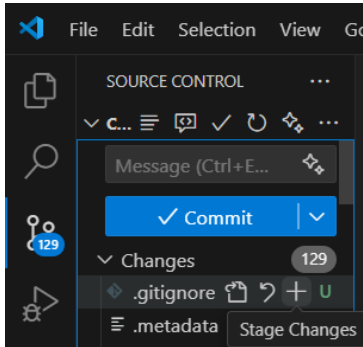


4	<p>Setelah itu akan otomatis terbuka di Edge, dan akan muncul tampilan seperti ini</p>

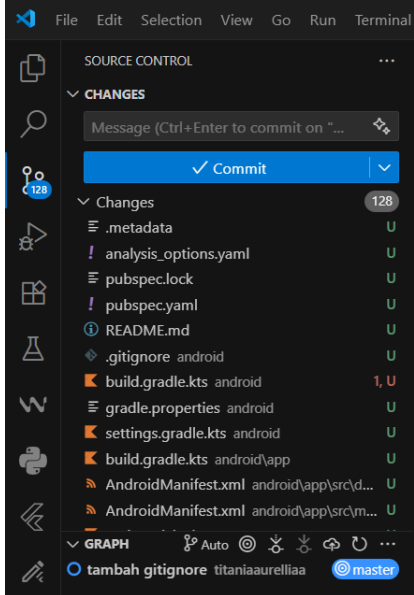
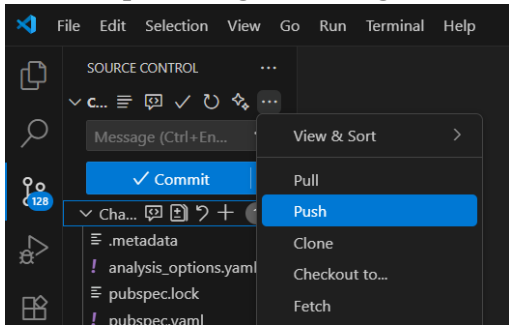
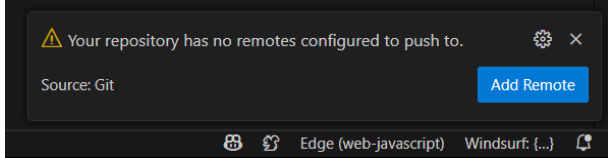
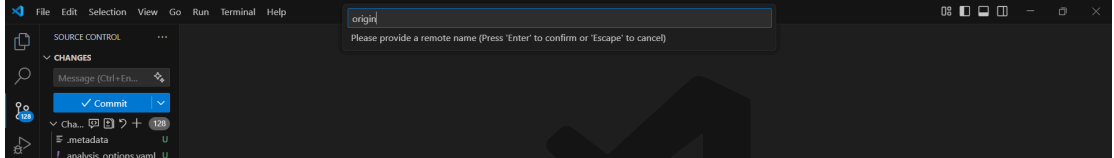
### Praktikum Ke-3 Membuat Repository GitHub dan Laporan Praktikum

Langkah	Jawaban/Deskripsi
1	<p>Login ke akun GitHub Anda, lalu buat repository baru dengan nama "flutter-fundamental-part1"</p>
2	<p>Lalu klik tombol "Create repository" lalu akan tampil seperti gambar berikut.</p>



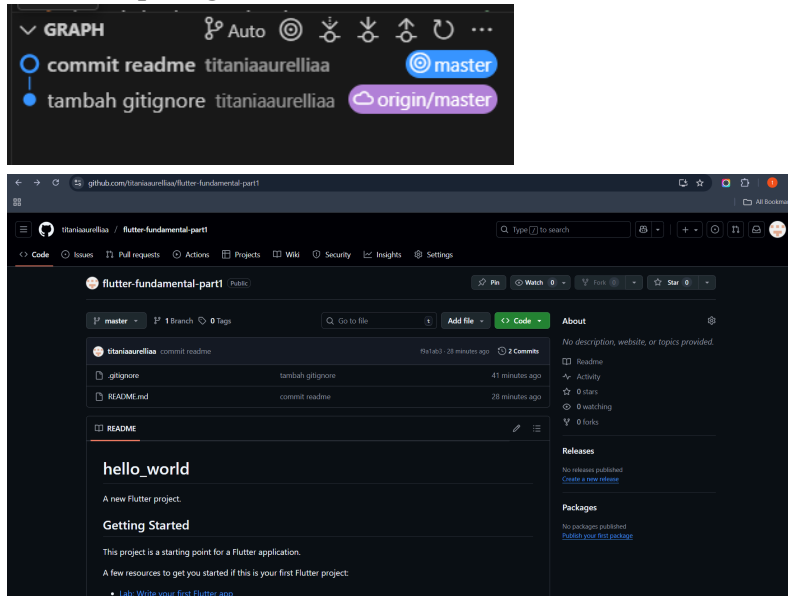
	
3	<p><b>Kembali ke VS code, project flutter hello_world, buka terminal pada menu Terminal &gt; New Terminal. Lalu ketik perintah berikut untuk inisialisasi git pada project Anda.</b></p> 
4	<p><b>Pilih menu Source Control di bagian kiri, lalu lakukan stages (+) pada file .gitignore untuk mengunggah file pertama ke repository GitHub.</b></p> 
5	<p><b>Beri pesan commit "tambah gitignore" lalu klik Commit (✓)</b></p>



	
6	<b>Lakukan push dengan klik bagian menu titik tiga &gt; Push</b> 
7	<b>Di pojok kanan bawah akan tampil seperti gambar berikut. Klik "Add Remote"</b> 
8	<b>Salin tautan repository Anda dari browser ke bagian ini, lalu klik Add remote</b>  <b>Setelah berhasil, tulis remote name dengan "origin"</b> 
9	<b>Lakukan hal yang sama pada file README.md mulai dari Langkah 4. Setelah berhasil melakukan push, masukkan username GitHub Anda dan password berupa token yang</b>

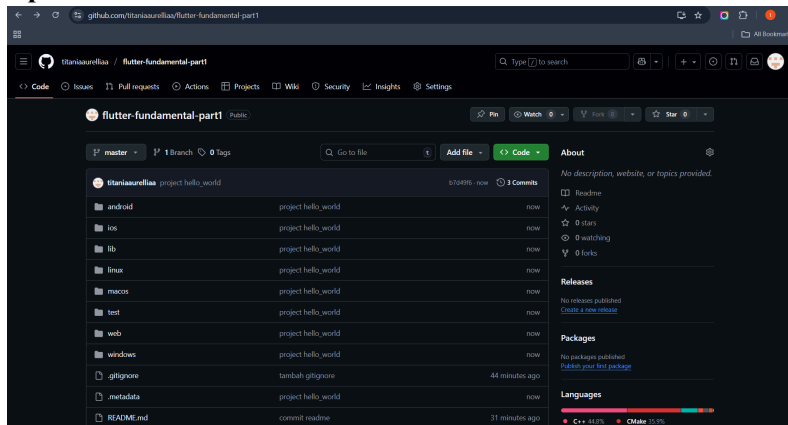


telah dibuat (pengganti password konvensional ketika Anda login di browser GitHub). Reload halaman repository GitHub Anda, maka akan tampil hasil push kedua file tersebut seperti gambar berikut.



10

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.



11

Kembali ke VS Code, ubah platform di pojok kanan bawah ke emulator atau device atau bisa juga menggunakan browser Chrome. Lalu coba running project hello\_world dengan tekan F5 atau Run > Start Debugging. Tunggu proses kompilasi hingga selesai, maka aplikasi flutter pertama Anda akan tampil seperti berikut.



12	<p>Silakan screenshot seperti pada Langkah 11, namun teks yang ditampilkan dalam aplikasi berupa nama lengkap Anda. Simpan file screenshot dengan nama 01.png pada folder images (buat folder baru jika belum ada) di project hello_world Anda. Lalu ubah isi README.md seperti berikut, sehingga tampil hasil screenshot pada file README.md. Kemudian push ke repository Anda.</p>

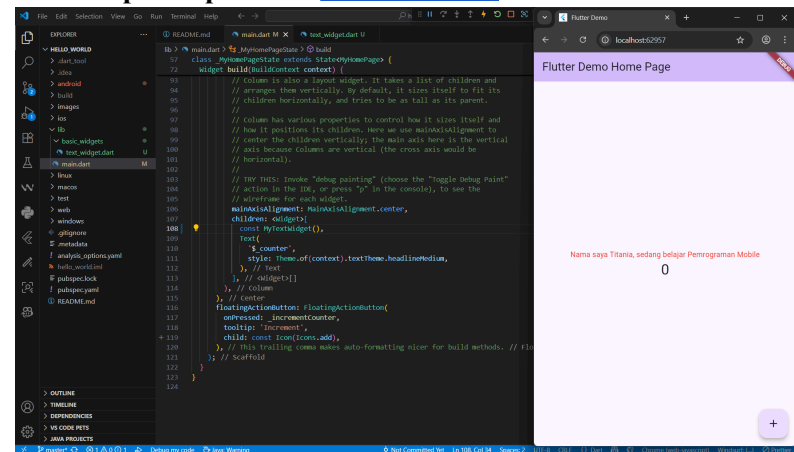
#### Praktikum Ke-4 Menerapkan Widget Dasar

Langkah	Jawaban/Deskripsi
1	Buat folder baru basic_widgets di dalam folder lib. Kemudian buat file baru di dalam basic_widgets dengan nama text_widget.dart. Ketik atau salin kode program berikut ke project hello_world Anda pada file text_widget.dart.



```
lib > basic_widgets > text_widget.dart > MyTextWidget
1 import 'package:flutter/material.dart';
2
3 Windsurf: Refactor | Explain
4 class MyTextWidget extends StatelessWidget {
5   const MyTextWidget({Key? key}) : super(key: key);
6
7   @override
8   Widget build(BuildContext context) {
9     return const Text(
10       "Nama saya Titania, sedang belajar Pemrograman Mobile",
11       style: TextStyle(color: Colors.red, fontSize: 14),
12       textAlign: TextAlign.center); // Text
13 }
```

Lakukan import file text\_widget.dart ke main.dart, lalu ganti bagian text widget dengan kode di atas. Maka hasilnya seperti gambar berikut. Screenshot hasil milik Anda, lalu dibuat laporan pada file [README.md](#).



2

Buat sebuah file image\_widget.dart di dalam folder basic\_widgets dengan isi kode berikut.

```
lib > basic_widgets > image_widget.dart > MyImageWidget
1 import 'package:flutter/material.dart';
2
3 Windsurf: Refactor | Explain
4 class MyImageWidget extends StatelessWidget {
5   const MyImageWidget({Key? key}) : super(key: key);
6
7   @override
8   Widget build(BuildContext context) {
9     return const Image(
10       image: AssetImage("logo_polinema.jpg")
11     ); // Image
12 }
```

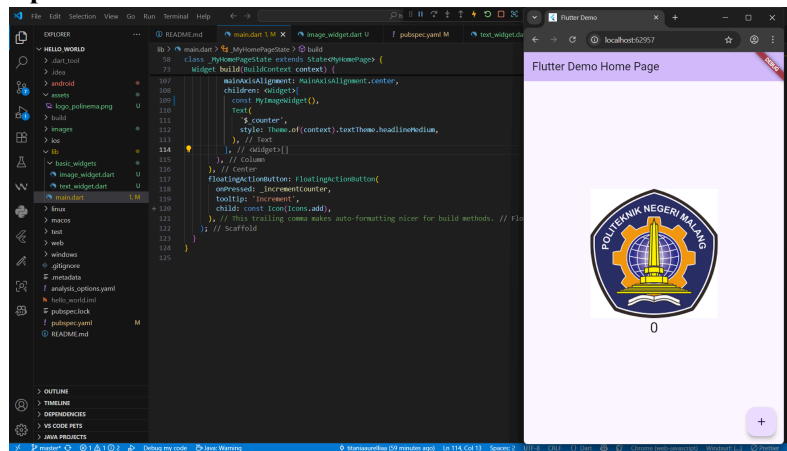




Lakukan penyesuaian asset pada file pubspec.yaml dan tambahkan file logo Anda di folder assets project hello\_world.

```
! pubspec.yaml
91 flutter:
92   assets:
93     - logo_polinema.png
```

Jangan lupa sesuaikan kode dan import di file main.dart kemudian akan tampil gambar seperti berikut.



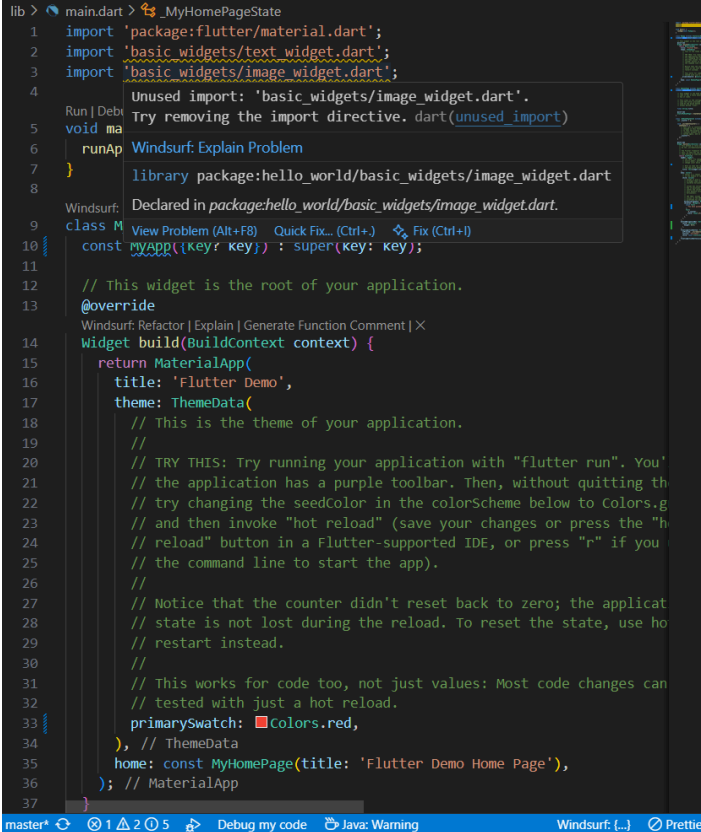
## Praktikum Ke-5 Menerapkan Widget Material Design dan iOS Cupertino

Langkah	Jawaban/Deskripsi
1	Buat file di basic_widgets > loading_cupertino.dart. Import stateless widget dari material dan cupertino. Lalu isi kode di dalam method Widget build adalah sebagai berikut.



	<pre>lib &gt; basic_widgets &gt; loading_cupertino.dart &gt; LoadingCupertino 1  import 'package:flutter/cupertino.dart'; 2  import 'package:flutter/material.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: &lt;Widget&gt;[ 15             CupertinoButton( 16               child: const Text("Contoh button"), 17               onPressed: () {}, 18             ), // CupertinoButton 19             const CupertinoActivityIndicator(), 20           ], // &lt;Widget&gt;[] 21         ), // Column 22       ), // Container 23     ); // MaterialApp 24   } 25 }</pre>
2	<p><b>Buat file di basic_widgets &gt; fab_widget.dart. Import stateless widget dari material. Lalu isi kode di dalam method Widget build adalah sebagai berikut.</b></p> <pre>lib &gt; basic_widgets &gt; fab_widget.dart &gt; FabWidgetWithLogic 1  import 'package:flutter/material.dart'; 2 3  class FabWidgetWithLogic extends StatelessWidget { 4    const FabWidgetWithLogic({super.key}); 5 6    @override 7    Widget build(BuildContext context) { 8      return MaterialApp( 9        home: Scaffold( 10         appBar: AppBar( 11           title: const Text('FAB Example'), 12         ), // AppBar 13         body: const Center( 14           child: Text('Tekan FAB di pojok kanan bawah'), 15         ), // Center 16         floatingActionButton: FloatingActionButton( 17           onPressed: () { 18             // Tampilkan snackbar ketika FAB ditekan 19             ScaffoldMessenger.of(context).showSnackBar( 20               const SnackBar( 21                 content: Text('Tombol Like ditekan!'), 22                 duration: Duration(seconds: 2), 23               ), // SnackBar 24             ); 25           }, 26           tooltip: 'Like', // Tooltip saat long press 27           child: const Icon(Icons.thumb_up), 28           backgroundColor: Colors.pink, 29         ), // FloatingActionButton 30       ), // Scaffold 31     ); // MaterialApp 32   } 33 }</pre>



3	<p><b>Scaffold widget digunakan untuk mengatur tata letak sesuai dengan material design. Ubah isi kode main.dart seperti berikut.</b></p> 
4	<p><b>Dialog widget pada flutter memiliki dua jenis dialog yaitu AlertDialog dan SimpleDialog. Ubah isi kode main.dart seperti berikut.</b></p>



	 <pre>lib &gt; main.dart &gt; showAlertDialog 1 import 'package:flutter/material.dart'; 2 import 'basic_widgets/text_widget.dart'; 3 import 'basic_widgets/image_widget.dart'; 4 5 void main() { 6   runApp(const MyApp()); 7 } 8 9 class MyApp extends StatelessWidget { 10   const MyApp({Key? key}) : super(key: key); 11 12   // This widget is the root of your application. 13   @override 14   Widget build(BuildContext context) { 15     return const MaterialApp( 16       home: Scaffold( 17         body: MyLayout(), 18       ), // Scaffold 19     ); // MaterialApp 20   } 21 } 22 23 class MyLayout extends StatelessWidget { 24   const MyLayout({Key? key}) : super(key: key); 25 26   @override 27   Widget build(BuildContext context) { 28     return Padding( 29       padding: const EdgeInsets.all(8.0), 30       child: ElevatedButton( 31         child: const Text('Show alert'), 32         onPressed: () { 33           showAlertDialog(context); 34         }, 35       ), // ElevatedButton 36     ); 37   } 38 }</pre>
5	<b>Flutter 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:</b>



	 <pre>lib &gt; main.dart &gt; MyApp 1 import 'dart:async'; 2 import 'package:flutter/material.dart'; 3 4 Run   Debug   Profile 5 void main() =&gt; runApp(const MyApp()); 6 7 Windsurf: Refactor   Explain 8 class MyApp extends StatelessWidget { 9   const MyApp({Key? key}) : super(key: key); 10 11   @override 12   Widget build(BuildContext context) { 13     return MaterialApp( 14       home: Scaffold( 15         appBar: AppBar(title: const Text("Contoh TextField")), 16         body: const TextField( 17           obscureText: false, 18           decoration: InputDecoration( 19             border: OutlineInputBorder(), 20             labelText: 'Nama', 21           ), // InputDecoration 22         ), // TextField 23       ), // Scaffold 24     ); // MaterialApp 25   } 26 27 Windsurf: Refactor   Explain 28 class MyHomePage extends StatefulWidget { 29   const MyHomePage({Key? key, required this.title}) : super(key: key); 30 31   final String title; 32 33   @override 34   State&lt;MyHomePage&gt; createState() =&gt; _MyHomePageState(); 35 } 36 37 Windsurf: Refactor   Explain 38 class _MyHomePageState extends State&lt;MyHomePage&gt; { 39   // ... 40 }</pre>
6	<b>Date and Time Pickers termasuk pada kategori input dan selection widget, berikut adalah contoh penggunaan Date and Time Pickers.</b>



```
lib > main.dart > _MyHomePageState
1 | import 'dart:async';
2 | import 'package:flutter/material.dart';
3 |
Run | Debug | Profile
4 | void main() => runApp(const MyApp());
5 |
Windsurf: Refactor | Explain
6 | class MyApp extends StatelessWidget {
7 |   const MyApp({Key? key}) : super(key: key);
8 |
9 |   @override
Windsurf: Refactor | Explain | Generate Function Comment | X
10 |   Widget build(BuildContext context) {
11 |     return const MaterialApp(
12 |       title: 'Contoh Date Picker',
13 |       home: MyHomePage(title: 'Contoh Date Picker'),
14 |     ); // MaterialApp
15 |   }
16 | }
17 |
Windsurf: Refactor | Explain
18 | class MyHomePage extends StatefulWidget {
19 |   const MyHomePage({Key? key, required this.title}) : super(key: key);
20 |
21 |   final String title;
22 |
23 |   @override
24 |   _MyHomePageState createState() => _MyHomePageState();
25 | }
26 |
Windsurf: Refactor | Explain
27 | class _MyHomePageState extends State<MyHomePage> {
28 |   // Variable/State untuk mengambil tanggal
29 |   DateTime selectedDate = DateTime.now();
30 |
31 |   // Initial SelectDate Flutter
Windsurf: Refactor | Explain | Generate Function Comment | X
32 |   Future<void> _selectDate(BuildContext context) async {
33 |     // Initial DateTime Final Picked
34 |     final DateTime? picked = await showDatePicker(
35 |
```