

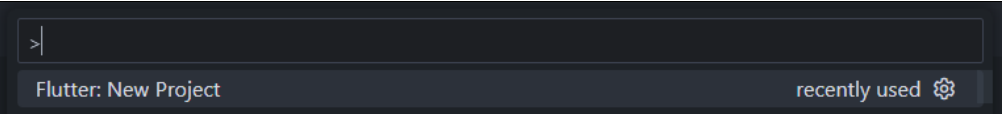
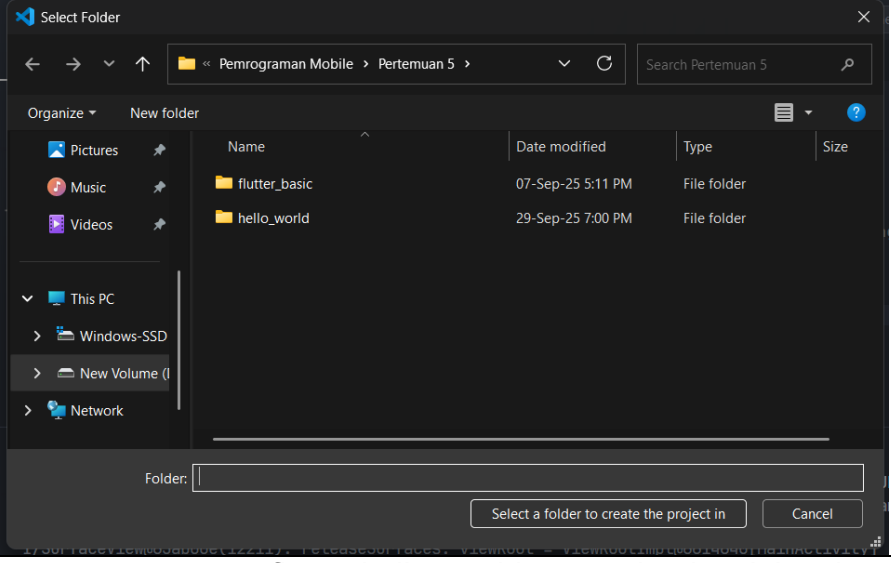
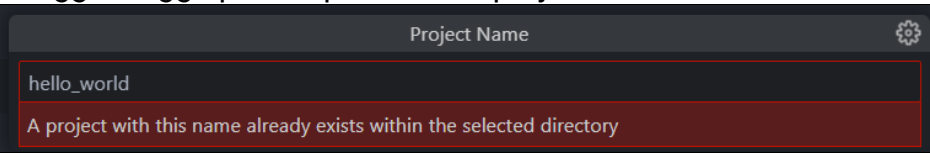
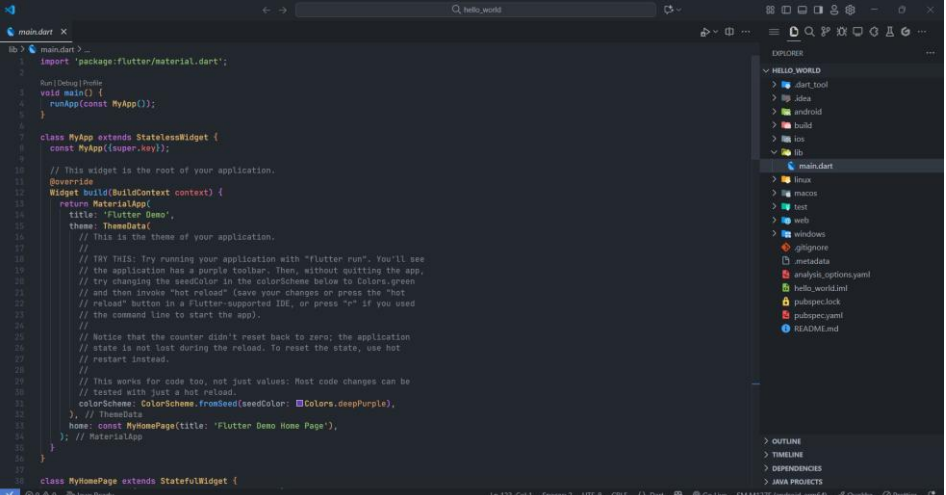
PERTEMUAN 5
APLIKASI PERTAMA & WIDGET DASAR FLUTTER
PEMROGRAMAN MOBILE



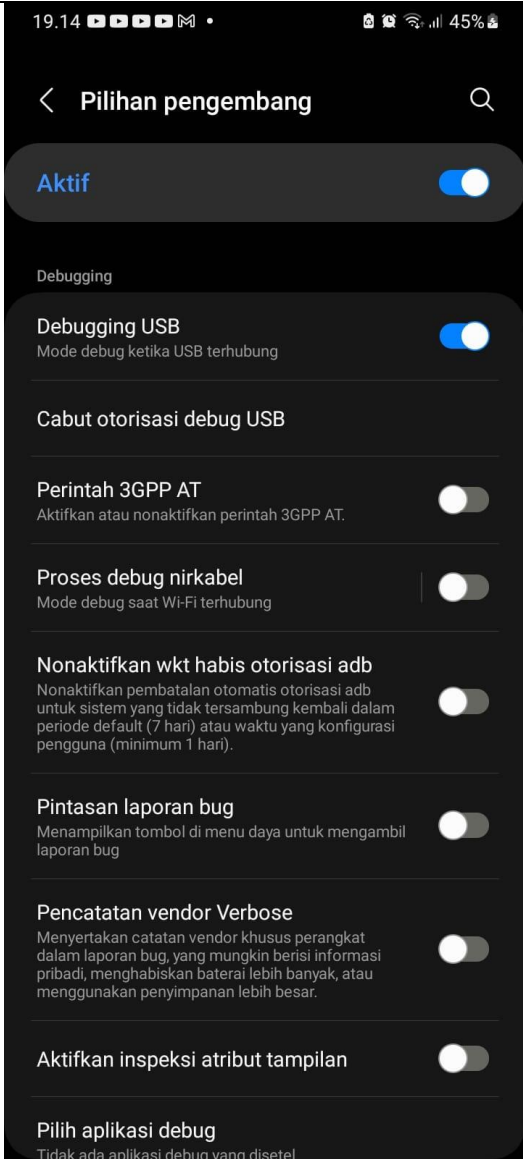
GALUNG ERLYAN TAMA (12)
2341720054
TI-3E

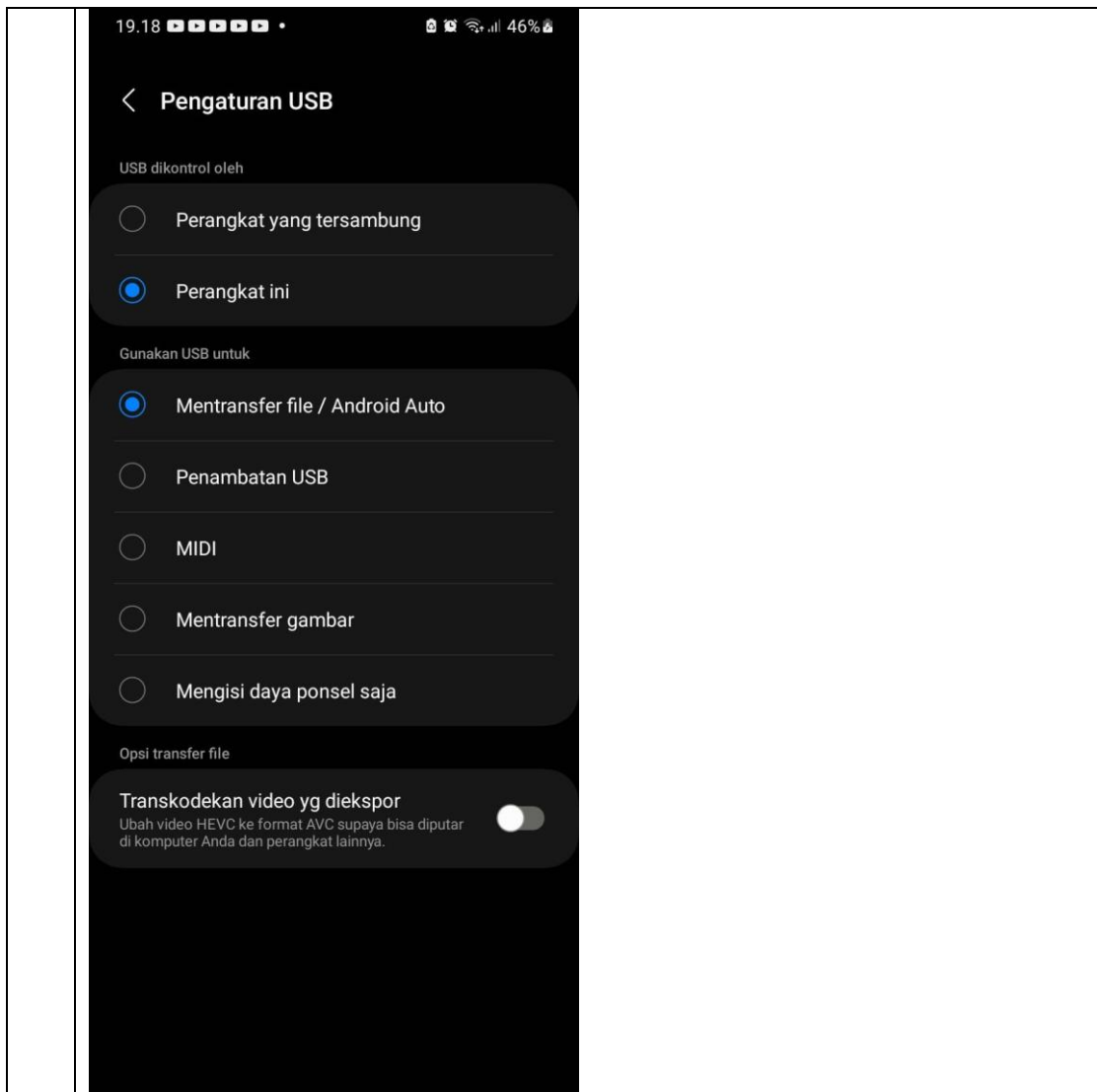
JURUSAN TEKNOLOGI INFORMASI
D-IV TEKNIK INFORMATIKA
POLITEKNIK NEGERI MALANG
2025

1. Praktikum 1: Membuat Project Flutter Baru

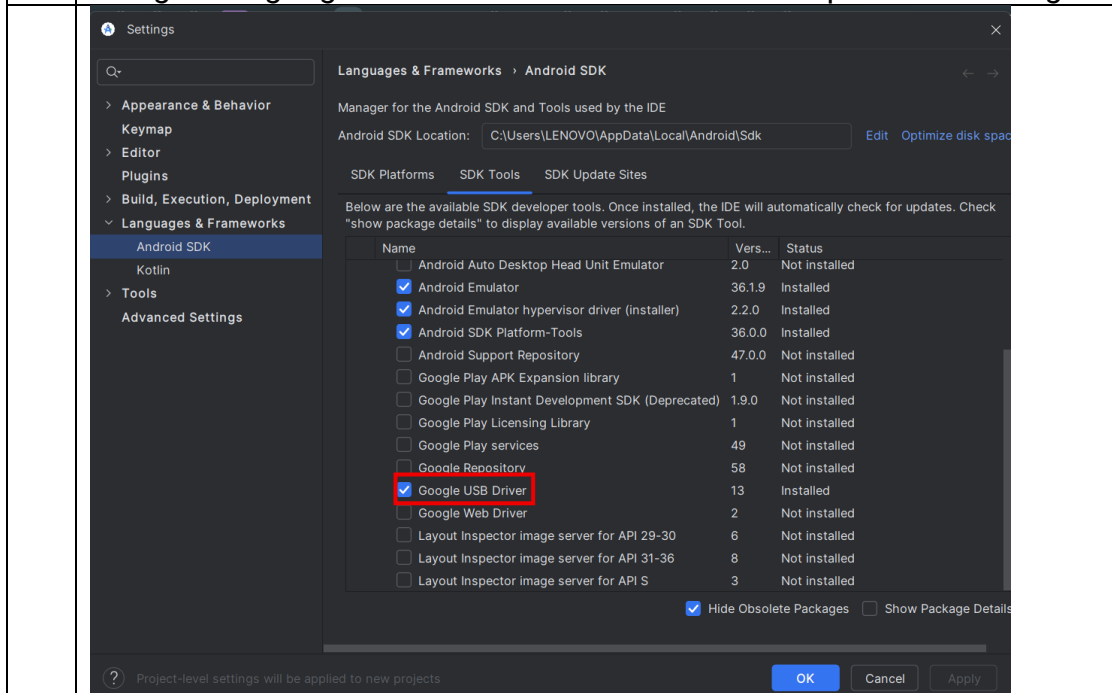
No	Penjelasan
1	Buka VS Code, lalu tekan tombol Ctrl + Shift + P maka akan tampil Command Palette, lalu ketik Flutter. Pilih New Application Project.
	
2	Kemudian buat folder sesuai style laporan praktikum yang Anda pilih. Disarankan pada folder dokumen atau desktop atau alamat folder lain yang tidak 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.
	

2. Praktikum 2: Menghubungkan Perangkat Android atau Emulator

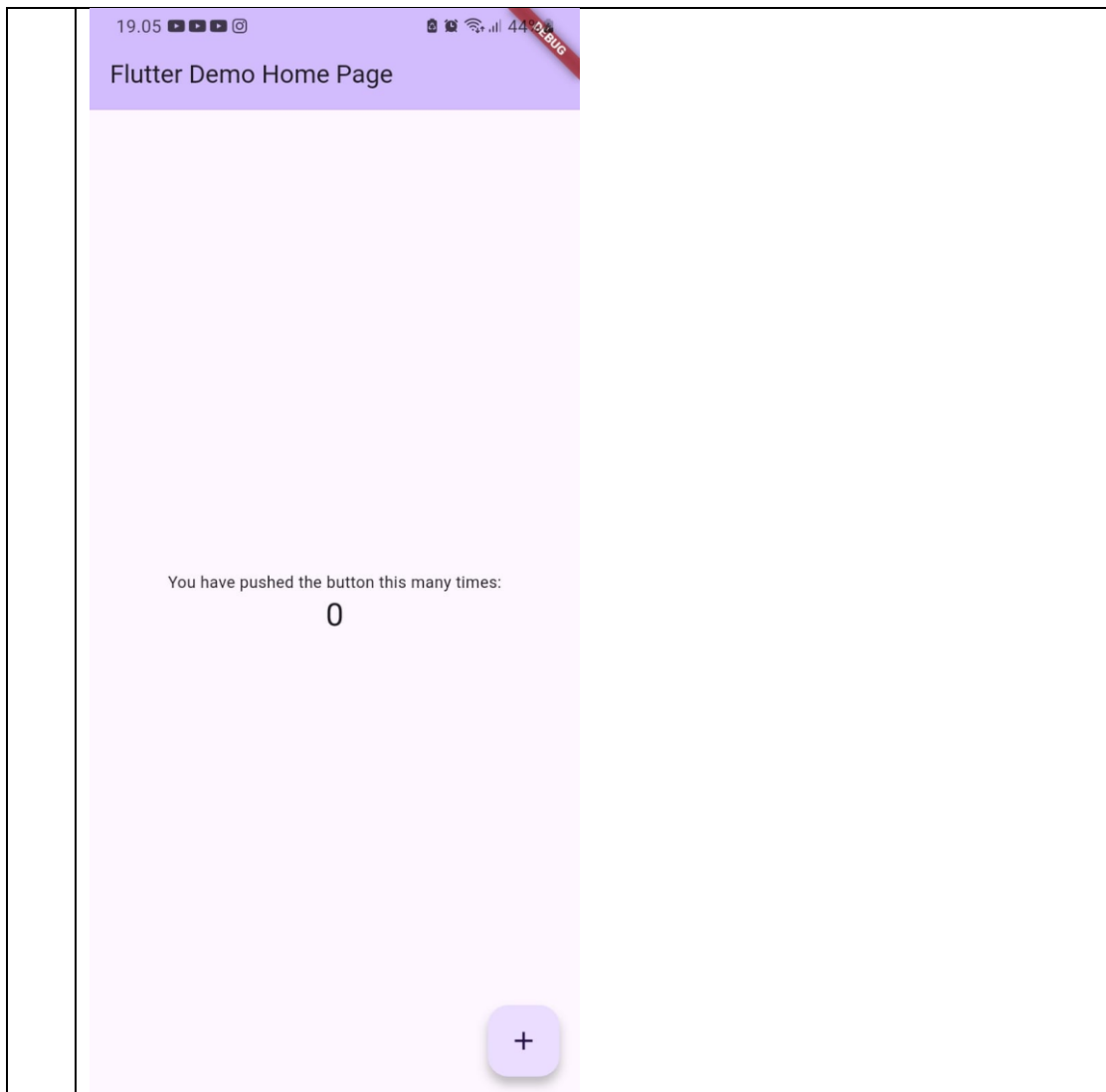
No	Penjelasan
1	Mengaktifkan Developer mode dan USB Debugging
	 <p>19.14 100% 45%</p> <p>< Pilihan pengembang</p> <p>Aktif</p> <p>Debugging</p> <p>Debugging USB Mode debug ketika USB terhubung</p> <p>Cabut otorisasi debug USB</p> <p>Perintah 3GPP AT Aktifkan atau nonaktifkan perintah 3GPP AT.</p> <p>Proses debug nirkabel Mode debug saat Wi-Fi terhubung</p> <p>Nonaktifkan wkt habis otorisasi adb Nonaktifkan pembatalan otomatis otorisasi adb untuk sistem yang tidak tersambung kembali dalam periode default (7 hari) atau waktu yang konfigurasi pengguna (minimum 1 hari).</p> <p>Pintasan laporan bug Menampilkan tombol di menu daya untuk mengambil laporan bug</p> <p>Pencatatan vendor Verbose Menyertakan catatan vendor khusus perangkat dalam laporan bug, yang mungkin berisi informasi pribadi, menghabiskan baterai lebih banyak, atau menggunakan penyimpanan lebih besar.</p> <p>Aktifkan inspeksi atribut tampilan</p> <p>Pilih aplikasi debug Tidak ada aplikasi debug yang disetel</p>
2	Menyambungkan usb dari laptop ke smartphone



3 Menginstall google usb driver dalam android studio pada sdk manager

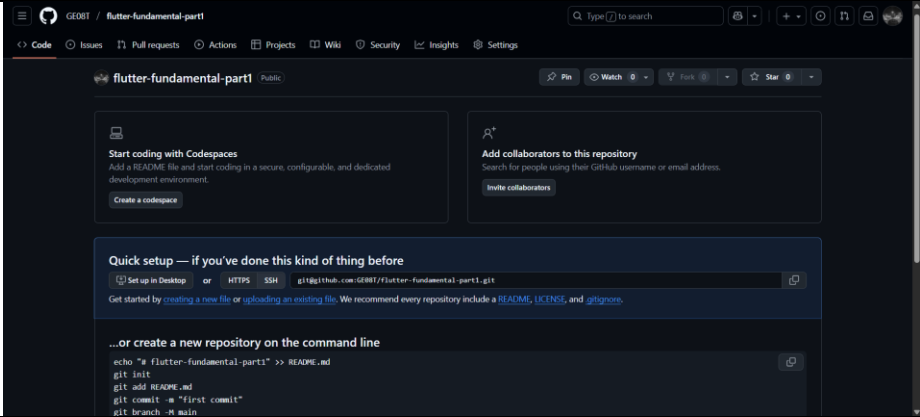
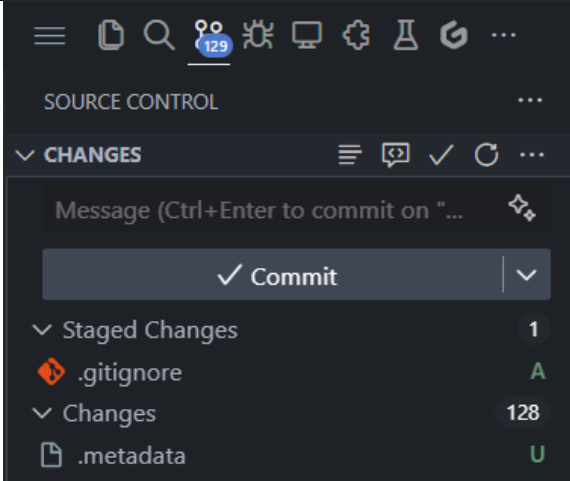



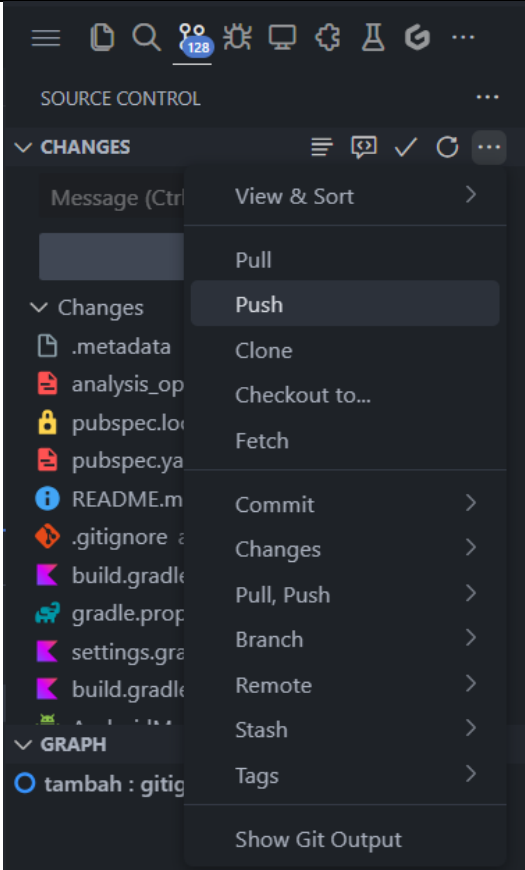
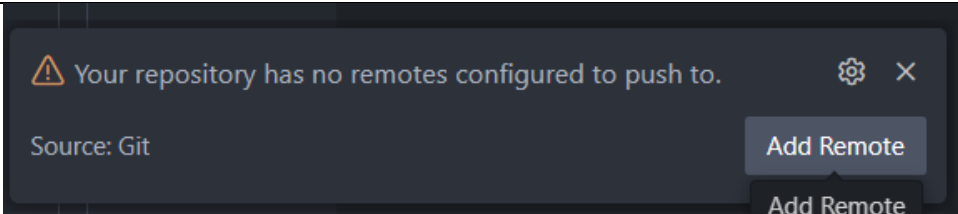
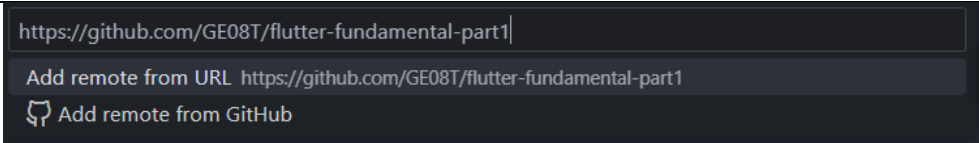
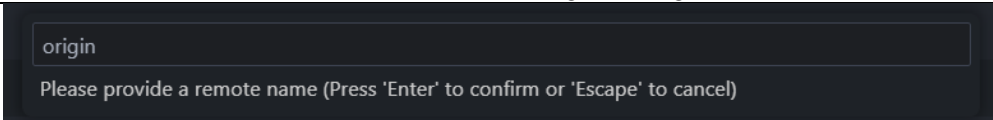
4	<p>List semua device yang terhubung di laptop emulator maupun physical</p> <pre> PS D:\Kuliah\Semester 5\Pemrograman Mobile\Pertemuan 5\hello_world> flutter devices Found 4 connected devices: SM M127F (mobile) • RR8T104QZJP • android-arm64 • Android 13 (API 33) Windows (desktop) • windows • windows-x64 • Microsoft Windows [Version 10.0.26100.6584] Chrome (web) • chrome • web-javascript • Google Chrome 140.0.7339.208 Edge (web) • edge • web-javascript • Microsoft Edge 140.0.3485.94 Run "flutter emulators" to list and start any available device emulators. If you expected another device to be detected, please run "flutter doctor" to diagnose potential issues. You may also try increasing the time to wait for connected devices with the "--device-timeout" flag. Visit https://flutter.dev/setup/ for troubleshooting tips. ❖PS D:\Kuliah\Semester 5\Pemrograman Mobile\Pertemuan 5\hello_world> </pre> <p>Pastikan device physical yang ingin disambungkan ada dilist</p>
5	<p>Running aplikasi flutter dengan id dari device physical yang tersambung</p> <pre> PS D:\Kuliah\Semester 5\Pemrograman Mobile\Pertemuan 5\hello_world> flutter run -d RR8T104QZJP Launching lib\main.dart on SM M127F in debug mode... Running Gradle task 'assembleDebug'... 8.6s ✓ Built build/app/outputs/flutter-apk/app-debug.apk Installing build/app/outputs/flutter-apk/app-debug.apk... 43.1s D/FlutterJNI(12211): Beginning load of flutter... D/FlutterJNI(12211): Flutter (null) was loaded normally! I/flutter (12211): [IMPORTANT:flutter/shell/platform/android/android_context_vk_impeller.cc(62)] Using the Impeller rendering backend (Vulkan). Syncing files to device SM M127F... 95ms Flutter run key commands. r Hot reload. R Hot restart. h List all available interactive commands. d Detach (terminate "flutter run" but leave application running). c Clear the screen q Quit (terminate the application on the device). </pre> <p>Jika sudah syncing files to device berarti aplikasi sedang diinstall oleh flutter</p>
6	<p>Jika sudah selesai maka akan diredirect ke aplikasi baru yang tadi</p>

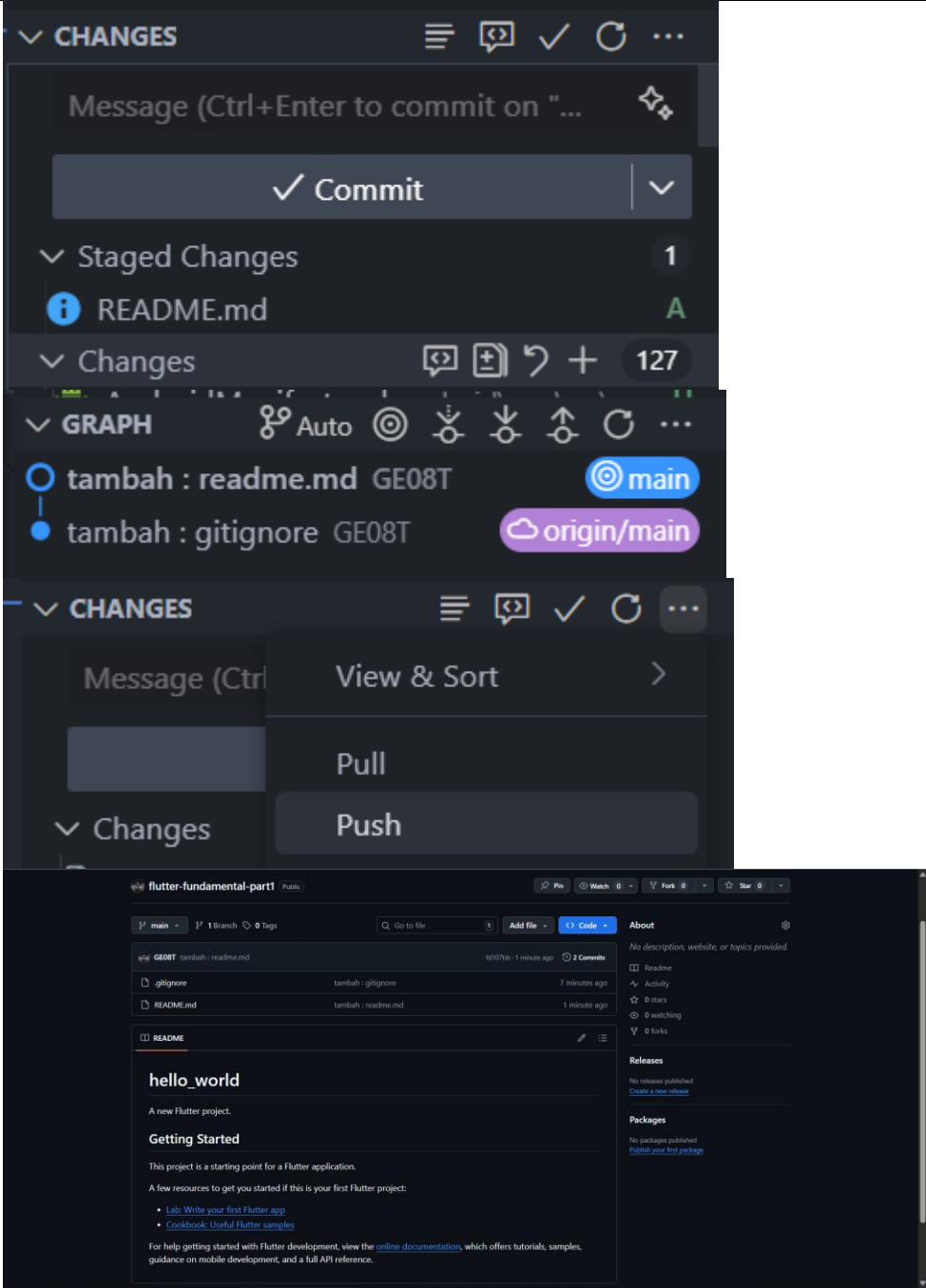
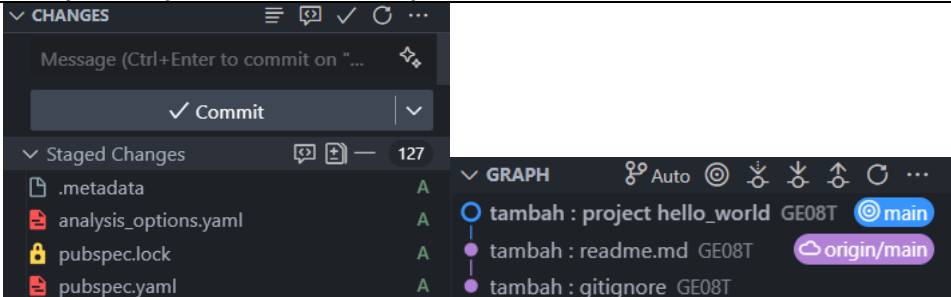


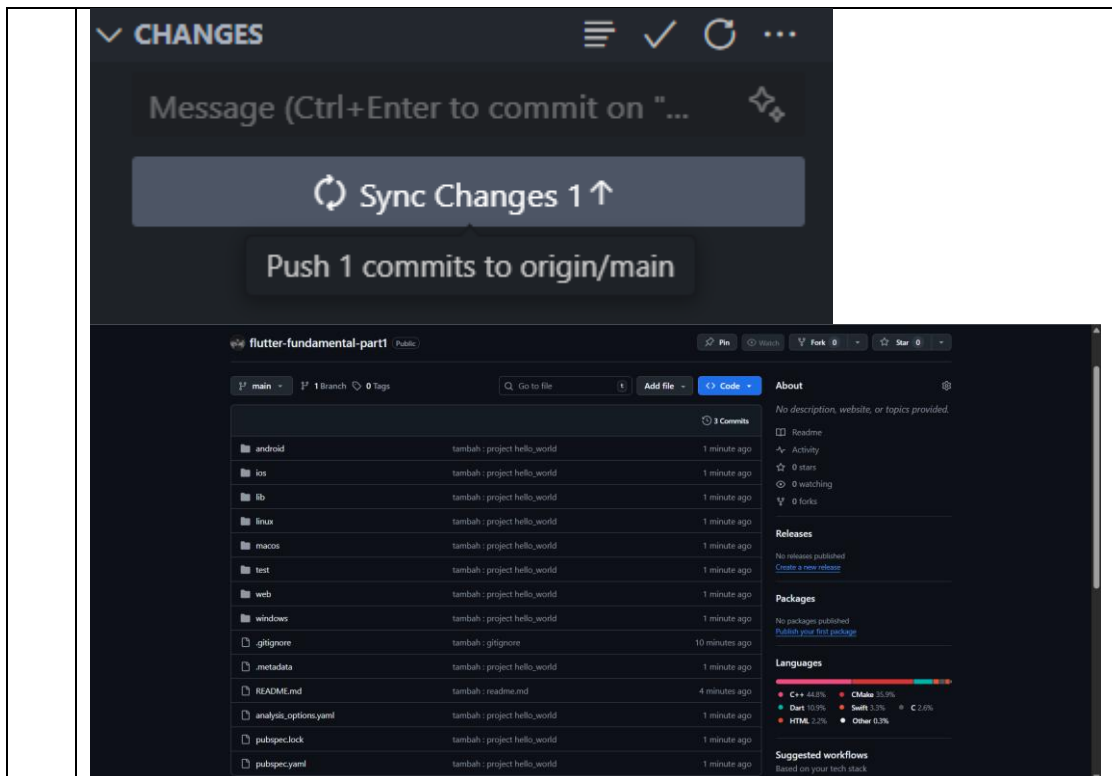
3. Praktikum 3: Membuat Repository GitHub dan Laporan Praktikum

No	Penjelasan
1	Login ke akun GitHub Anda, lalu buat repository baru dengan nama "flutter-fundamental-part1"
	
2	Lalu klik tombol "Create repository" lalu akan tampil seperti gambar berikut.

	
3	Kembali ke VS code, project flutter hello_world, buka terminal pada menu Terminal > New Terminal. Lalu ketik perintah berikut untuk inisialisasi git pada project Anda.
	<pre>PS D:\Kuliah\Semester 5\Pemrograman Mobile\Pertemuan 5\hello_world> git init Initialized empty Git repository in D:\Kuliah\Semester 5\Pemrograman Mobile\Pertemuan 5\hello_world\.git/ PS D:\Kuliah\Semester 5\Pemrograman Mobile\Pertemuan 5\hello_world></pre>
4	Pilih menu Source Control di bagian kiri, lalu lakukan stages (+) pada file .gitignore untuk mengunggah file pertama ke repository GitHub.
	
5	Beri pesan commit "tambah gitignore" lalu klik Commit (✓)
	
6	Lakukan push dengan klik bagian menu titik tiga > Push

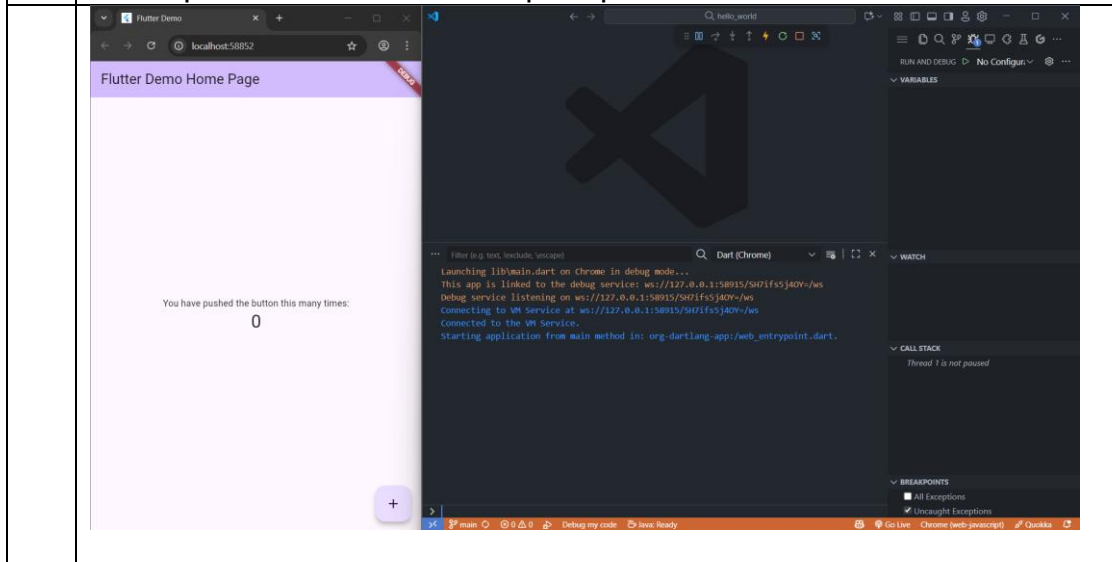
		
7	Di pojok kanan bawah akan tampil seperti gambar berikut. Klik "Add Remote"	
		
8	Salin tautan repository Anda dari browser ke bagian ini, lalu klik Add remote	
		
	Setelah berhasil, tulis remote name dengan "origin"	
		
9	Lakukan hal yang sama pada file README.md mulai dari Langkah 4. Setelah berhasil melakukan push, masukkan username GitHub Anda dan password berupa token yang 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.
	 <p>The screenshot shows the VS Code interface with the 'CHANGES' panel at the top, displaying 'Message (Ctrl+Enter to commit on "...)' and a 'Commit' button. Below it, 'Staged Changes' shows 'README.md'. The 'GRAPH' panel shows a commit history with 'tambah : readme.md' and 'tambah : gitignore'. A 'Pull' and 'Push' menu is visible. Below this, the GitHub repository 'flutter-fundamental-part1' is shown, displaying the 'README' file content, which includes 'hello_world' and 'Getting Started' sections.</p>
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.
	 <p>The screenshot shows the VS Code interface with the 'CHANGES' panel at the top, displaying 'Message (Ctrl+Enter to commit on "...)' and a 'Commit' button. Below it, 'Staged Changes' shows multiple files: '.metadata', 'analysis_options.yaml', 'pubspec.lock', and 'pubspec.yaml'. The 'GRAPH' panel shows a commit history with 'tambah : project hello_world', 'tambah : readme.md', and 'tambah : gitignore'.</p>



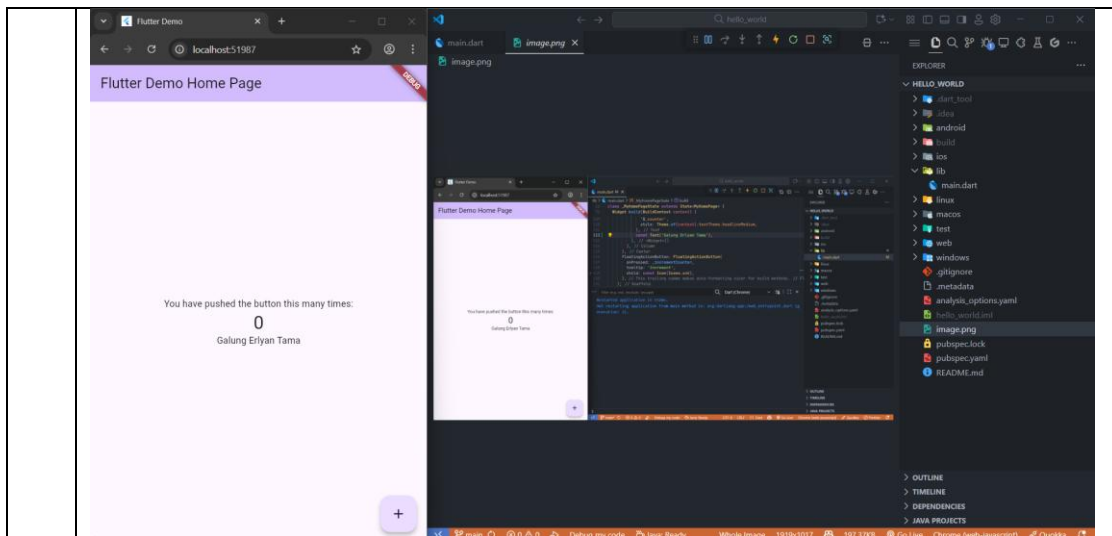
11

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.

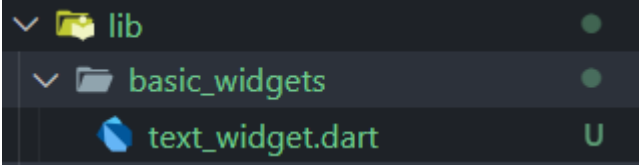
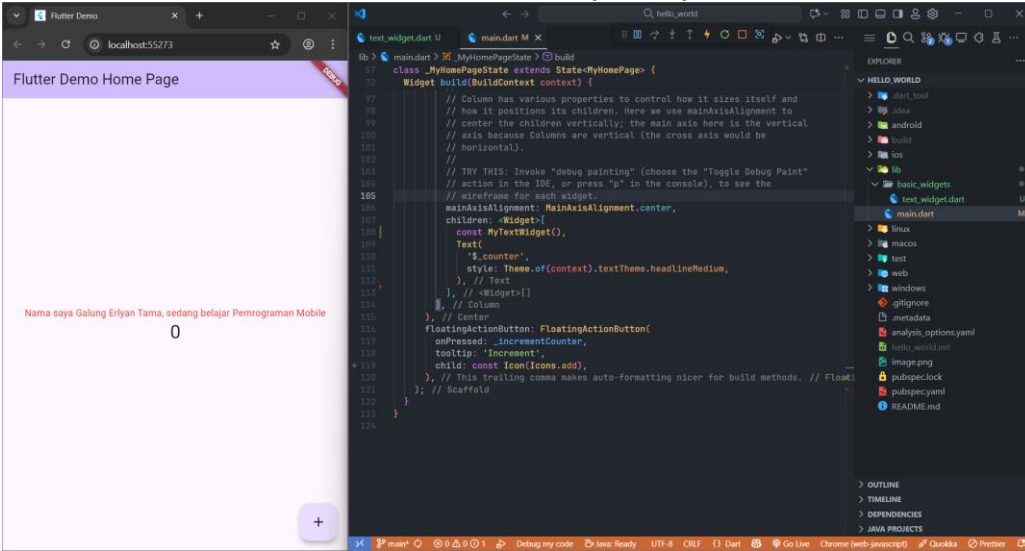


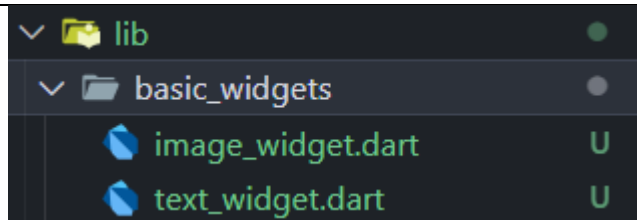
12

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.

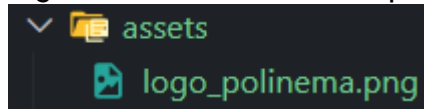


4. Praktikum 4: Menerapkan Widget Dasar

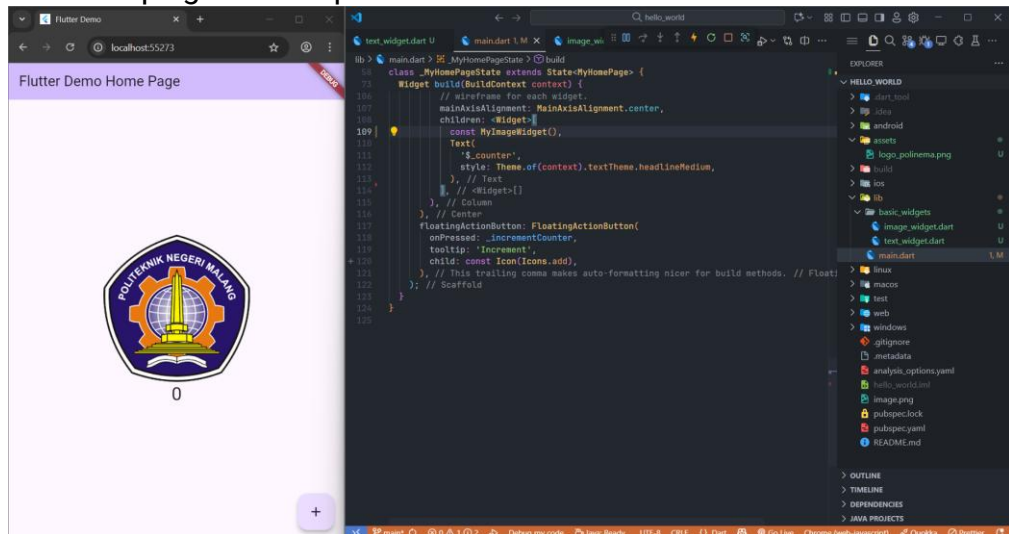
No	Penjelasan
1	<p>Buat folder baru <code>basic_widgets</code> di dalam folder <code>lib</code>. Kemudian buat file baru di dalam <code>basic_widgets</code> dengan nama <code>text_widget.dart</code>. Ketik atau salin kode program berikut ke project <code>hello_world</code> Anda pada file <code>text_widget.dart</code>.</p>
	 <p>Lakukan import file <code>text_widget.dart</code> ke <code>main.dart</code>, lalu ganti bagian text widget dengan kode di atas. Maka hasilnya seperti gambar berikut. Screenshot hasil milik Anda, lalu dibuat laporan pada file <code>README.md</code>.</p> 
2	<p>Buat sebuah file <code>image_widget.dart</code> di dalam folder <code>basic_widgets</code> dengan isi kode berikut.</p>



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

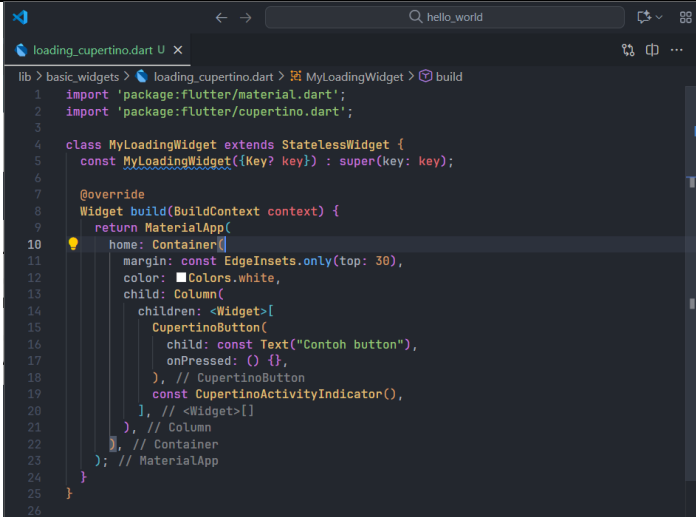
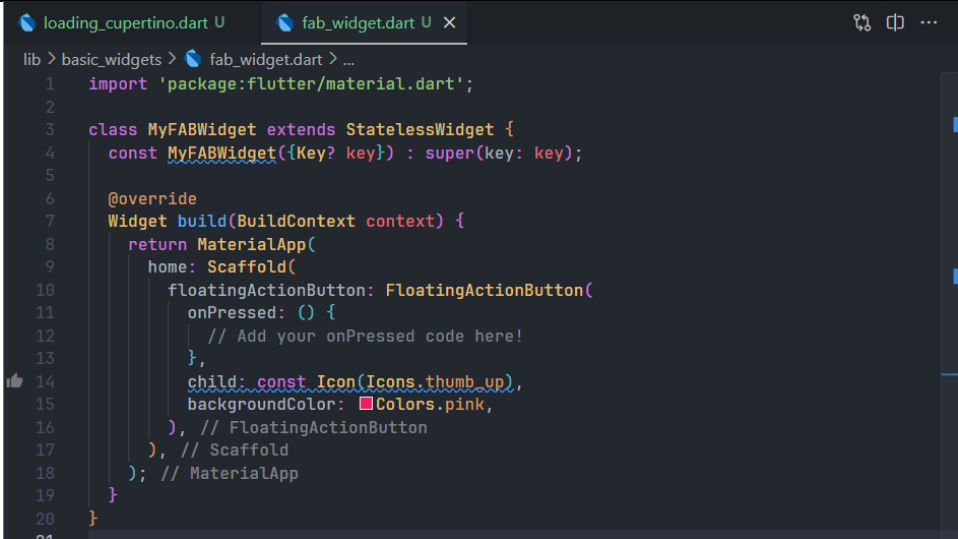
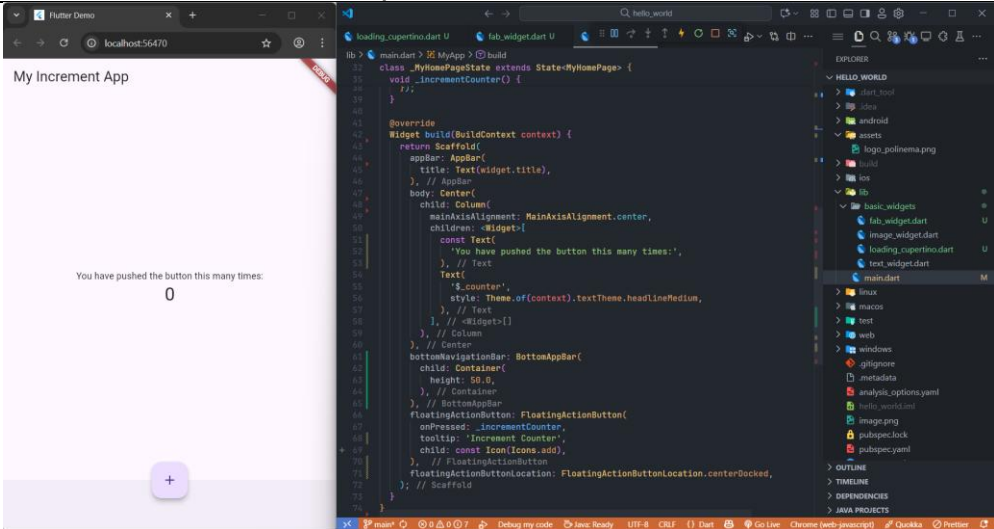




Jangan lupa sesuaikan kode dan import di file main.dart kemudian akan tampil gambar seperti berikut.

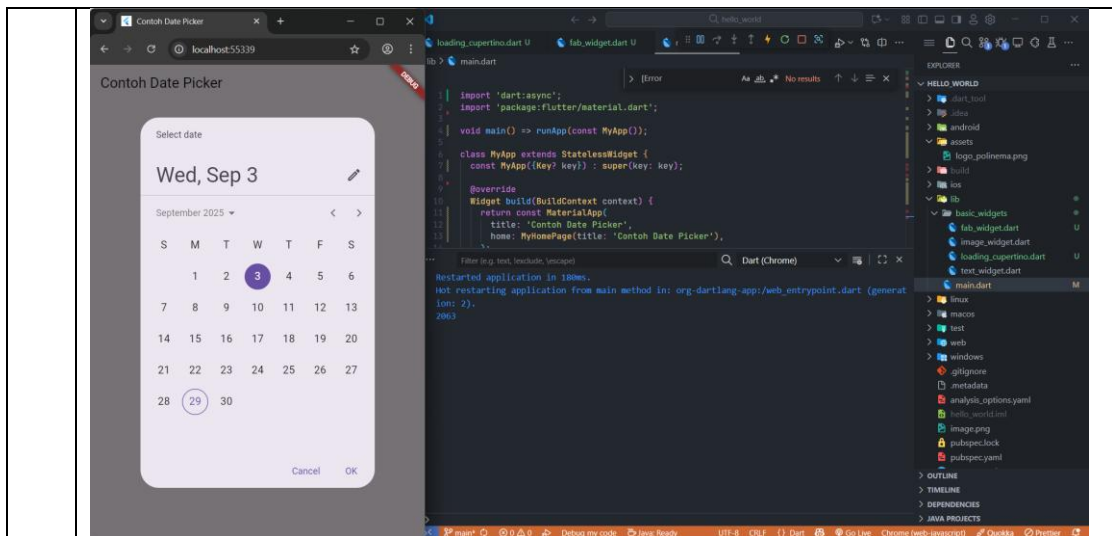


5. Praktikum 5: Menerapkan Widget Material Design dan iOS Cupertino

No	Penjelasan
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.

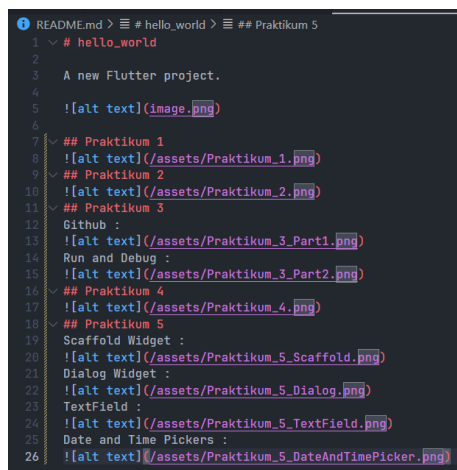
	
2	<p>Buat file di basic_widgets > fab_widget.dart. Import stateless widget dari material. Lalu isi kode di dalam method Widget build adalah sebagai berikut.</p>
	
3	<p>Scaffold Widget Ubah isi kode main.dart seperti berikut.</p>
	
4	<p>Dialog Widget</p>

	Ubah isi kode main.dart seperti berikut.
	 <pre> 1 class MyApp extends StatelessWidget { 2 // This widget is the root of your application. 3 Widget build(BuildContext context) { 4 return MaterialApp(5 home: Scaffold(6 body: MyLayout(), 7), // Scaffold 8); // MaterialApp 9 } 10 11 class MyLayout extends StatelessWidget { 12 const MyLayout({Key? key}) : super(key: key); 13 14 @override 15 Widget build(BuildContext context) { 16 return Padding(17 padding: const EdgeInsets.all(0.0), 18 child: ElevatedButton(19 child: const Text("Show alert"), 20 onPressed: () { 21 showAlert(context); 22 }, // ElevatedButton 23); // Padding 24); 25 26 showAlert(context); 27 28 // set up the AlertDialog 29 AlertDialog alert = AlertDialog(30 title: const Text("My title"), 31 content: const Text("This is my message"), 32 actions: [33 TextButton(34 child: const Text("OK"), 35 onPressed: () { 36 Navigator.pop(context); 37 }, // TextButton 38), 39], 40); 41 42 // set up the AlertDialog 43 showDialog(context: context, builder: (context) { 44 return alert; 45 }); 46 } 47 } 48 } </pre>
5	<p>Input dan Selection Widget</p> <p>Flutter menyediakan widget yang dapat menerima input dari pengguna aplikasi yaitu antara lain Checkbox, Date and Time Pickers, Radio Button, Slider, Switch, TextField.</p> <p>Contoh penggunaan TextField widget adalah sebagai berikut:</p>
	 <pre> 1 class MyApp extends StatelessWidget { 2 // This widget is the root of your application. 3 Widget build(BuildContext context) { 4 return MaterialApp(5 home: Scaffold(6 appBar: AppBar(7 title: const Text("Contoh TextField"), 8), 9 body: const Padding(10 // 2. Beri nilai padding yang diinginkan 11 // EdgeInsets.all(16.0) akan memberi jarak 16 piksel di semua sisi 12 padding: EdgeInsets.all(16.0), 13 child: TextField(14 obscureText: false, 15 decoration: InputDecoration(16 border: OutlineInputBorder(), 17 labelText: 'Nama', 18), 19), 20), // Scaffold 21); // MaterialApp 22 } 23 } </pre>
6	<p>Date and Time Pickers termasuk pada kategori input dan selection widget, berikut adalah contoh penggunaan Date and Time Pickers.</p>

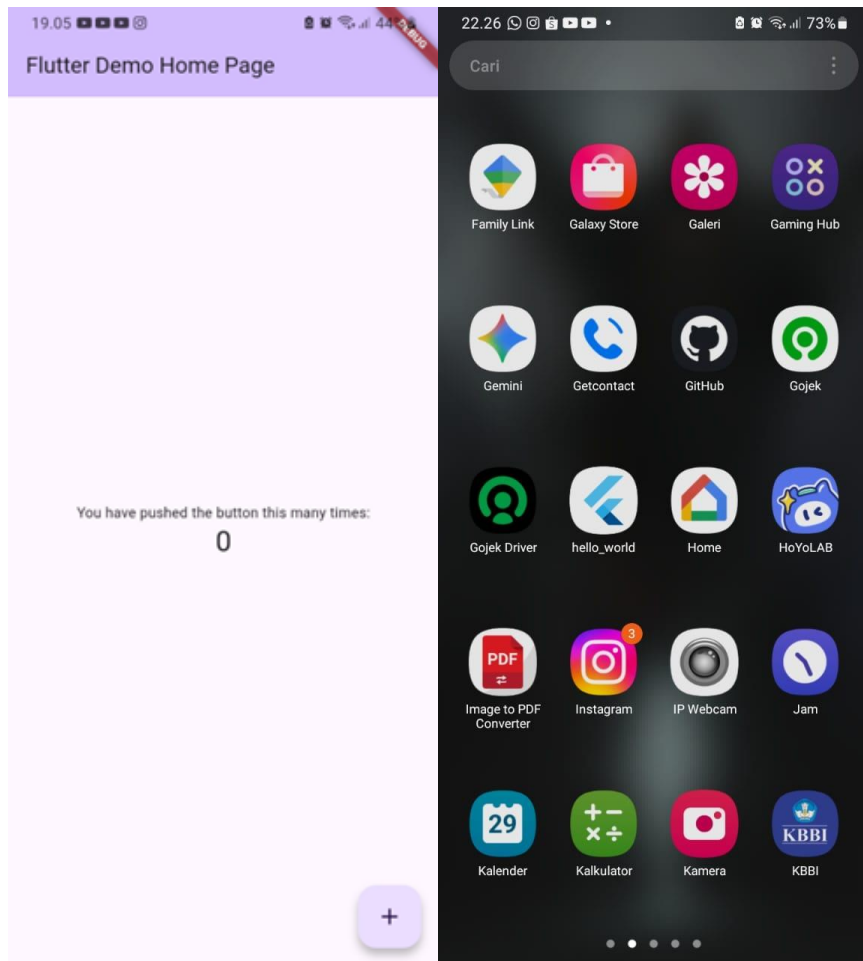


6. Tugas Praktikum

1. Selesaikan Praktikum 1 sampai 5, lalu dokumentasikan dan push ke repository Anda berupa screenshot setiap hasil pekerjaan beserta penjelasannya di file `README.md`!



2. Selesaikan Praktikum 2 dan Anda wajib menjalankan aplikasi `hello_world` pada perangkat fisik (device Android/iOS) agar Anda mempunyai pengalaman untuk menghubungkan ke perangkat fisik. Capture hasil aplikasi di perangkat, lalu buatlah laporan praktikum pada file `README.md`.



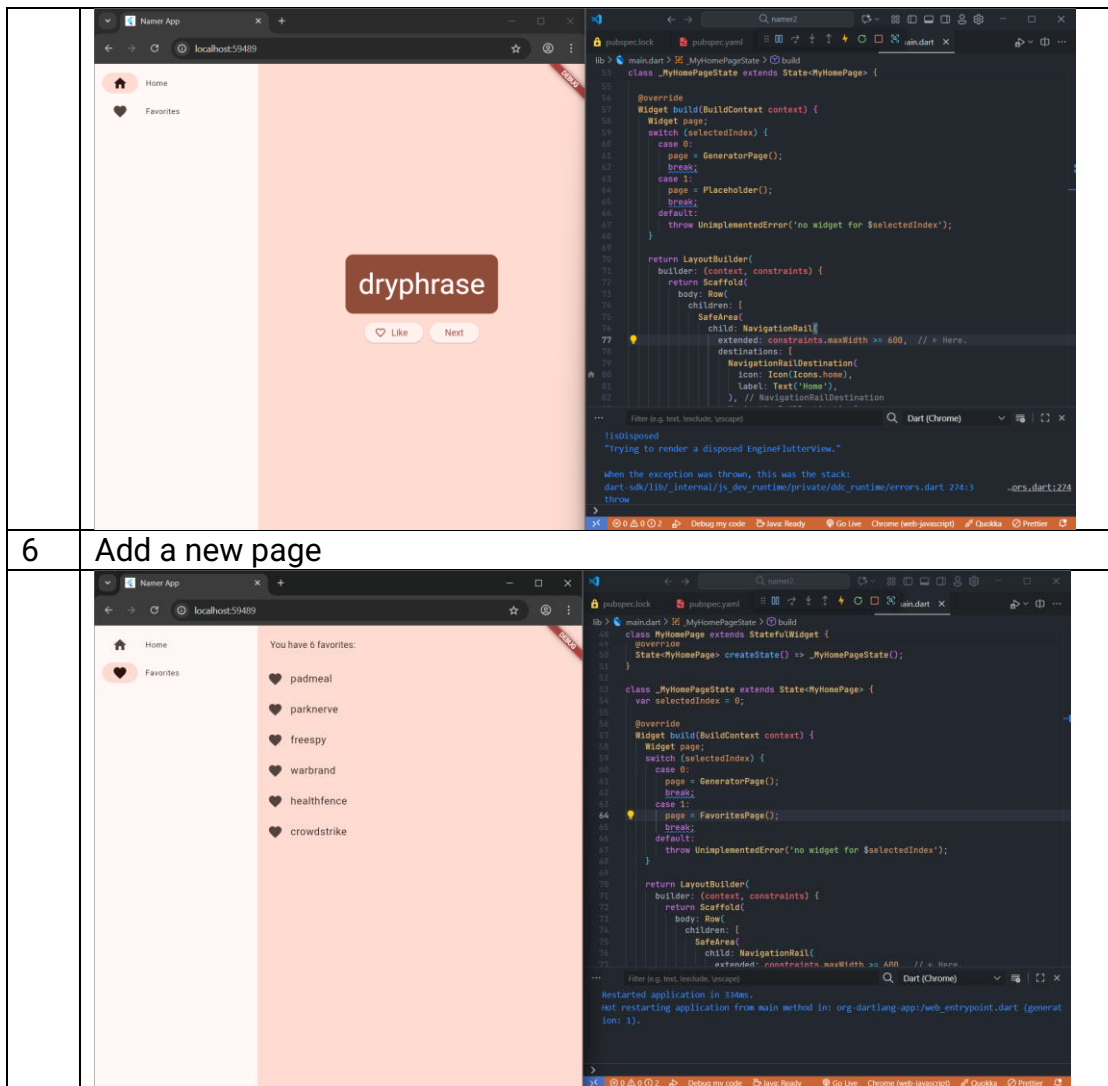
3. Pada praktikum 5 mulai dari Langkah 3 sampai 6, buatlah file widget tersendiri di folder `basic_widgets`, kemudian pada file `main.dart` cukup melakukan import widget sesuai masing-masing langkah tersebut!

Jawab : sudah di Praktikum 5

4. Selesaikan [Codelabs: Your first Flutter app](#), lalu buatlah laporan praktikumnya dan push ke repository GitHub Anda!

No	Langkah - Langkah
1	Create a project.

2	Add a button
	 
3	Make the app prettier
	 
4	Add functionality
	 
5	Add navigation rail



5. README.md berisi: capture hasil akhir tiap praktikum (*side-by-side*, bisa juga berupa file GIF agar terlihat proses perubahan ketika ada aksi dari pengguna) dengan menampilkan NIM dan Nama Anda sebagai ciri pekerjaan Anda.
6. Kumpulkan berupa link repository/commit GitHub Anda kepada dosen yang telah disepakati!

Github : <https://github.com/GE08T/flutter-fundamental-part1>