PEMROGRAMAN PERANGKAT BERGERAK

Tahun Ajaran 2025/2026

PRAKTIKUM 2 :

PENGENALAN FLUTTER

Pengembang Modul :

Yudha Islami Sulistya, S.Kom., M.Cs.

Nita Fitrotul Mar'ah

Aflah Rizkyadhafin Nurfikri

PENGENALAN FLUTTER DAN WIDGET DASAR

Tujuan Praktikum

- Mahasiswa mampu memahami konsep layout pada Flutter
- Mahasiswa dapat mengimplementasikan desain user interface pada Flutter

Apa Itu Flutter?

Flutter ditulis menggunakan bahasa C, C++ dan Dart dengan Google's Skia Graphics Engine untuk user interface. Engine yang digunakan untuk produk ini dikenal seperti Google Chrome, Chrome OS, Chromium OS, Mozilla Firefox, Mozilla Thunderbird, Android, Firefox OS dan sekarang Flutter. Flutter berjalan menggunakan Dart Virtual Machine (VM) di sistem operasi Windows, Linux, dan macOS. Dart VM menggunakan kompilasi kode just-in-time (JIT) yang menyediakan fitur hot-reload untuk menghemat waktu pengembangan.

Apa itu Widget?

Widget adalah elemen dasar yang digunakan untuk membangun antarmuka pengguna (UI). Setiap elemen visual dalam aplikasi Flutter, seperti tombol, teks, gambar, atau layout, direpresentasikan sebagai widget. Flutter menggunakan arsitektur berbasis widget, artinya hampir semua yang dilihat di layar adalah widget atau kombinasi dari widget-widget lainnya.

Beberapa widget pada Flutter:

1. **Stateless Widget & Stateful Widget**

   a. Stateless Widget

      Stateless Widget adalah widget yang nilai state-nya tidak dapat berubah (immutable) maka widget tersebut lebih bersifat statis dan memiliki interaksi yang terbatas.

```
class Heading extends StatelessWidget {


  final String text;
  const Heading({Key? key, required this.text}) : super(key: key);
```

```
  @override
  Widget build(BuildContext context){
    return Text(
      text,
      style: const TextStyle(
        fontSize: 24.0,
        fontWeight: FontWeight.bold,
      ),
    );
  }
}
```

b. Stateful Widget

Kebalikan dari Stateless Widget, Stateful Widget ialah widget yang state-nya dapat berubah-ubah nilainya, yang berarti Stateful Widget bersifat dinamis dan memiliki interaksi yang tak terbatas.

Penulisan Stateful Widget sangat berbeda dengan Stateless Widget, berikut penulisannya:

```
class ContohStateful extends StatefulWidget {

    final String parameterWidget; // ini parameter widget

     const ContohStateful({Key? key, required this.parameterWidget})
: super(key: key);

    @override
    _ContohStatefulState createState() => _ContohStatefulState();
}

class _ContohStatefulState extends State<ContohStateful>{
    String _dataState; // ini state dari Widget ContohStateful

    @override
    Widget build(BuildContext context){
        // isi sebuah widget
    }
```

```
}
```

Seperti yang kita lihat, penulisan Stateful Widget lebih panjang. Tetapi yang harus diperhatikan adalah properti dari setiap class-nya. Pada class Contoh Stateful propertinya hanya berupa parameter ketika memanggil ContohStateful, parameter tersebut tidak wajib ada. Sedangkan pada class _ContohStatefulState, properti _dataState merupakan state yang sebenarnya. Kita akan mengubah nilai _dataState.

2. Scaffold

Scaffold merupakan sebuah widget yang digunakan untuk membuat tampilan dasar material design pada aplikasi Flutter, yang dapat disebut juga dasar sebuah halaman pada aplikasi Flutter. Tampilan dasar tersebut seperti berikut:

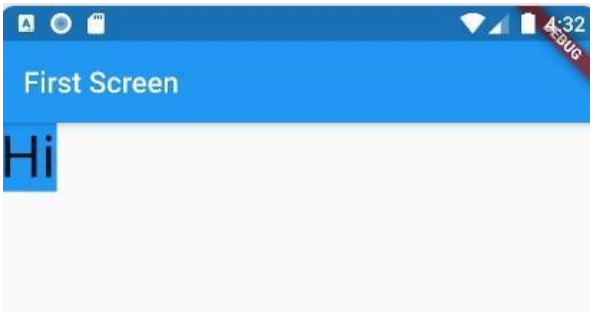
3. Container

Container adalah widget yang digunakan untuk melakukan styling, membuat sebuah shape (bentuk), dan layout pada widget child-nya. Sebagai contoh kode:

```
Container(
  color: Colors.blue,
  child: const Text(
    'Hi',
```

```
    style: TextStyle(fontSize: 40),
  ),
),
```

Pada kode di atas kita membuat sebuah Text "Hi" yang dibungkus oleh widget Container dan kita beri parameter color dengan nilai Colors.blue. Kita letakkan Container di dalam parameter body Scaffold.



Text "Hi" akan memiliki background berwarna biru.
Kita dapat mengatur lebar (width) dan tinggi (height) suatu Container seperti berikut:

```
Container(
  color: Colors.blue,
  width: 200,
  height: 100,
  child: const Text(
    'Hi',
    style: TextStyle(fontSize: 40),
  ),
),
```

Kode di atas ketika dijalankan hasilnya akan seperti berikut:

Dekorasi Container

Decoration merupakan bagian dari Container untuk styling. Pada decoration kita dapat menentukan warna background (solid/gradient color), shadow, border, border radius (membulatkan sudut), mengatur shape (bentuk), dan lain-lain.

4. Padding

Padding merupakan sebuah widget yang khusus untuk memberikan padding pada suatu widget.

Contoh penggunaan widget Padding seperti berikut:

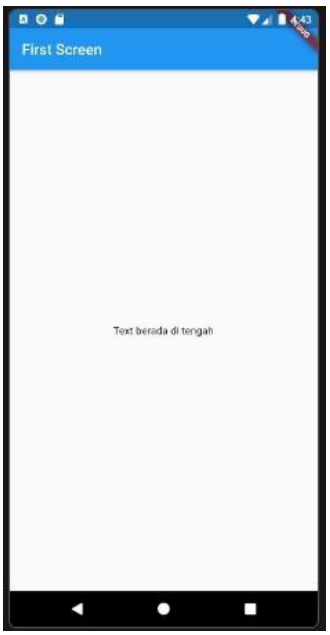
```
Padding(
  padding: const EdgeInsets.all(30),
  child: const Text('Ini Padding')
)
```

5. Center

Widget Center merupakan sebuah widget yang digunakan untuk membuat suatu widget berada pada posisi tengah. Penggunaan widget Center sangatlah simpel, yakni seperti berikut:

```
Center(
  child: const Text('Text berada di tengah'),
)
```

Widget Center hanya membutuhkan parameter child untuk membuat widget di dalamnya berada pada posisi tengah. Hasil dari Center seperti berikut:

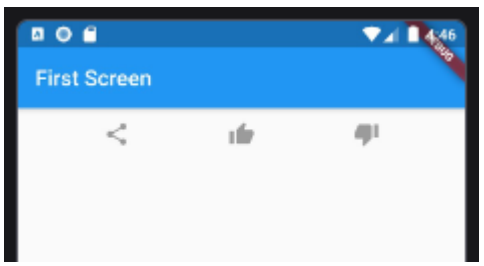


6. Row & Column

- Widget Row

Untuk membuat widget-widget berjajar secara horizontal kita harus memasukkan widget-widget tersebut ke dalam parameter children. Parameter children berisi kumpulan atau list dari widget karena kita dapat menyusun beberapa widget sekaligus di dalamnya.

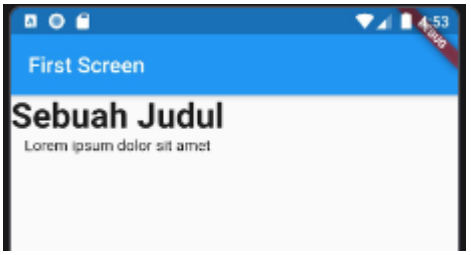
```
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: const <Widget>[
    Icon(Icons.share),
    Icon(Icons.thumb_up),
    Icon(Icons.thumb_down),
  ],
)
```



- **Widget Column**

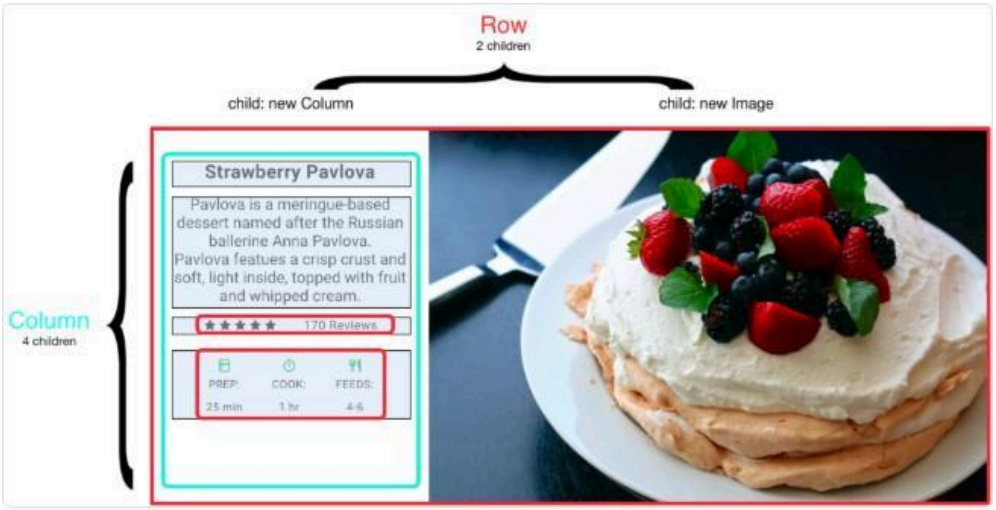
Kebalikan dari Row, Column merupakan suatu widget yang digunakan untuk membuat widget-widget tersusun berjajar secara vertikal. Column memiliki sintaks mirip dengan Row.

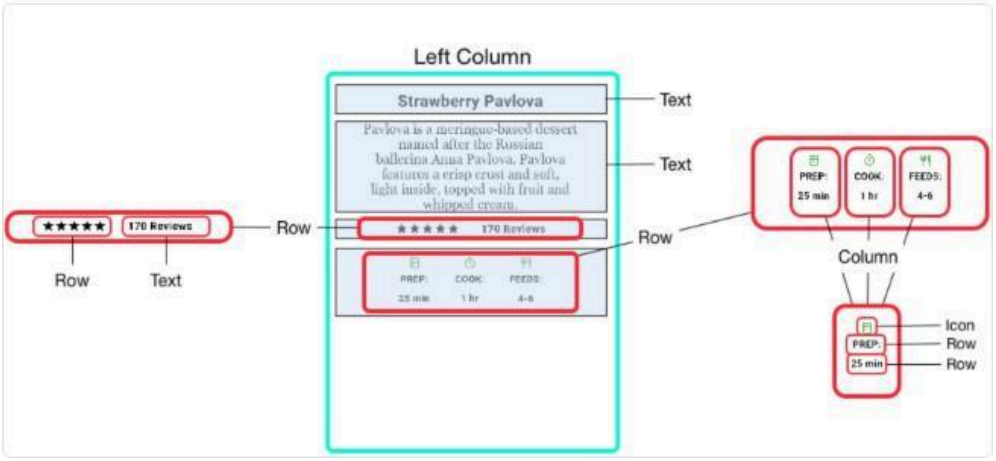
```
Column(
  children: const <Widget>[
    Text(
      'Sebuah Judul',
      style: TextStyle(fontSize:32, fontWeight: FontWeight.bold),
    ),
    Text('Lorem ipsum dolor sit amet'),
  ],
)
```



Kesimpulan

Untuk membuat sebuah widget-widget berjajar kita dapat menggunakan widget Row atau Column. Sebenarnya penggunaan Row dan Column dapat dipadukan sehingga dapat membuat sebuah layout yang kompleks seperti berikut:


Row
2 children
child: new Column
child: new Image
Column
4 children
Strawberry Pavlova
Pavlova is a meringue-based dessert named after the Russian ballerina Anna Pavlova. Pavlova features a crisp crust and soft, light inside, topped with fruit and whipped cream.
★★★★★ 170 Reviews
PREP: 25 min
COOK: 1 hr
FEEDS: 4-6


Left Column
Strawberry Pavlova
Text
Pavlova is a meringue-based dessert named after the Russian ballerina Anna Pavlova. Pavlova features a crisp crust and soft, light inside, topped with fruit and whipped cream.
Text
★★★★★ 170 Reviews
Row
Row
Text
Row
PREP: 25 min
COOK: 1 hr
FEEDS: 4-6
Column
Icon
Row
Row

Running Project Flutter Flutter

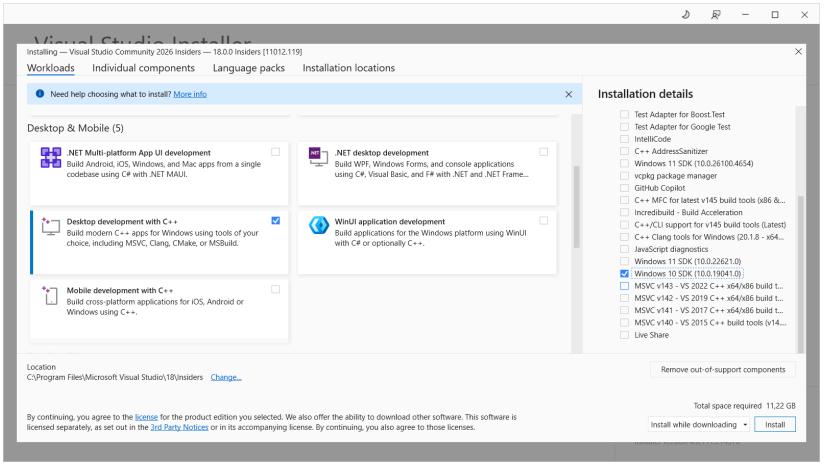
```
PS D:\Tugas\Asprak\Pemrograman Perangkat Bergerak\Modul 2\Flutter\flutter_application_1> flutter run
Connected devices:
Windows (desktop) • windows • windows-x64    • Microsoft Windows [Version 10.0.19045.6332]
Chrome (web)      • chrome  • web-javascript • Google Chrome 140.0.7339.128
Edge (web)        • edge    • web-javascript • Microsoft Edge 140.0.3485.81
[1]: Windows (windows)
[2]: Chrome (chrome)
[3]: Edge (edge)
Please choose one (or "q" to quit):
```

Ketika akan running project Flutter, kita bisa menggunakan 4 opsi device yang akan menampilkan antarmuka untuk aplikasi Flutter. Opsi tersebut diantaranya :

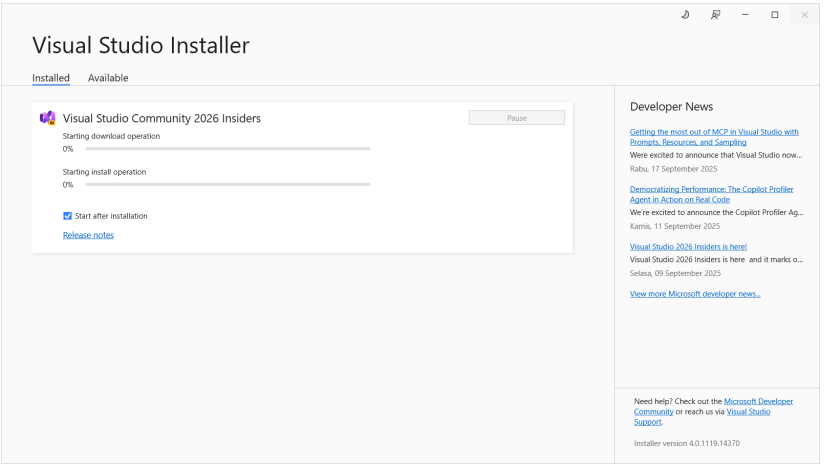
1. **Windows**

   Untuk running project Flutter menggunakan windows, kita perlu meng-install terlebih dahulu **"Dekstop development with C++"**, dari aplikasi **Visual Studio.** Untuk mendownload Visual Studio bisa melalui link berikut : https://visualstudio.microsoft.com/

   Setelah download installer Visual Studio, buka dan jalankan installer hingga selesai, lalu buka aplikasi Visual Studio. Pada halaman awal, scroll ke bawah hingga menemukan dan pilih modul **"Dekstop development with C++"**. Setelah memilih modul, pada bagian panel kanan, pilih sdk Windows sesuai dengan Windows yang terinstall pada laptop kalian.



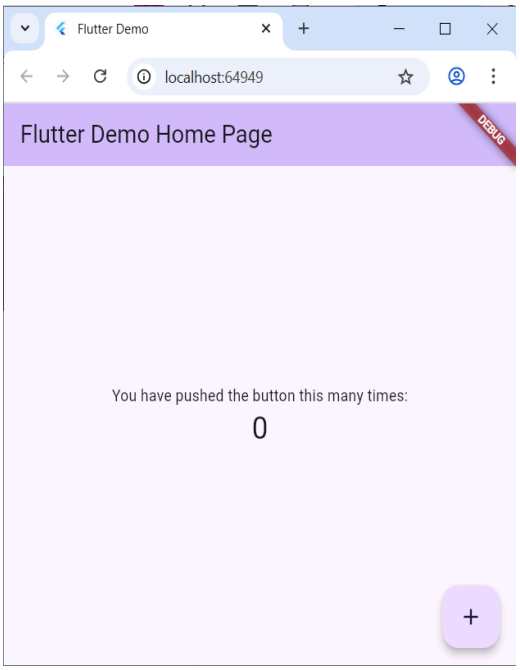
Klik install dan tunggu hingga proses download dan instalasi selesai.



Setelah instalasi modul selesai, kembali ke Visual Studio Code dan running kembali modul kalian. Pilih opsi 1, untuk running project di Windows. Tampilan aplikasi akan seperti berikut :

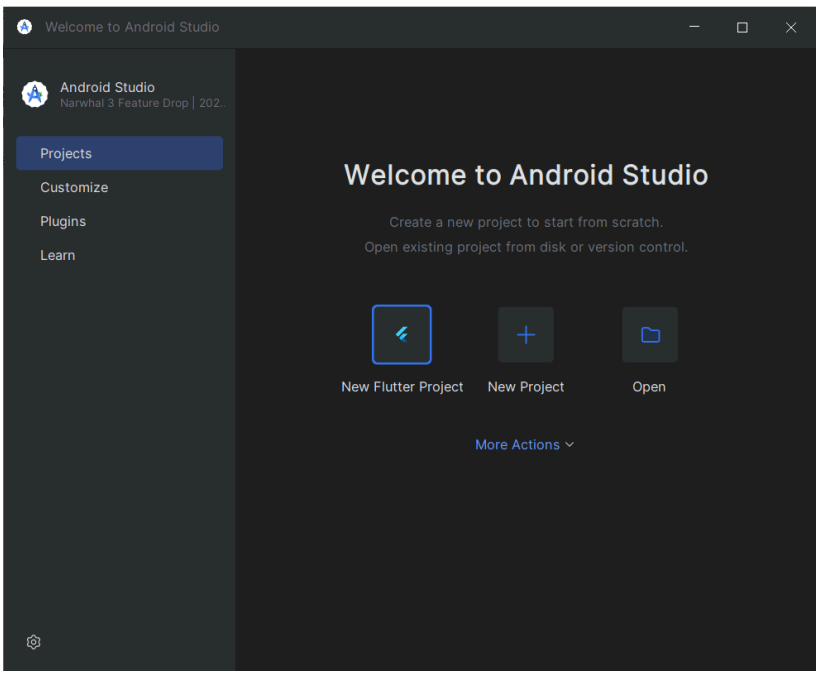
2. Chrome / Edge

Untuk running project pada web (Chrome/Edge), kita harus memastikan aplikasi Chrome/Edge sudah terinstall pada PC. Selanjutnya running project Flutter dengan memilih opsi 2 atau 3.

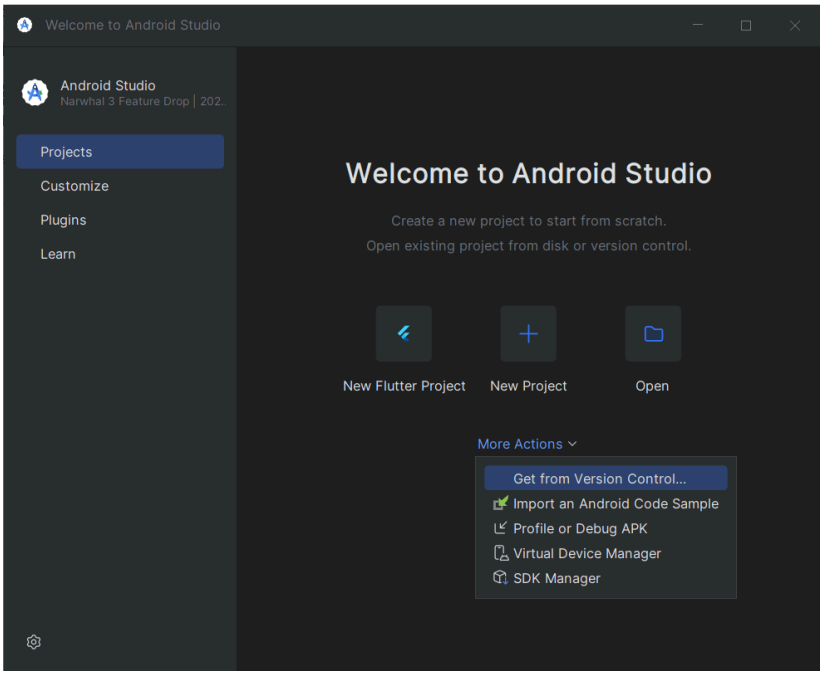


3. Emulator Android Studio

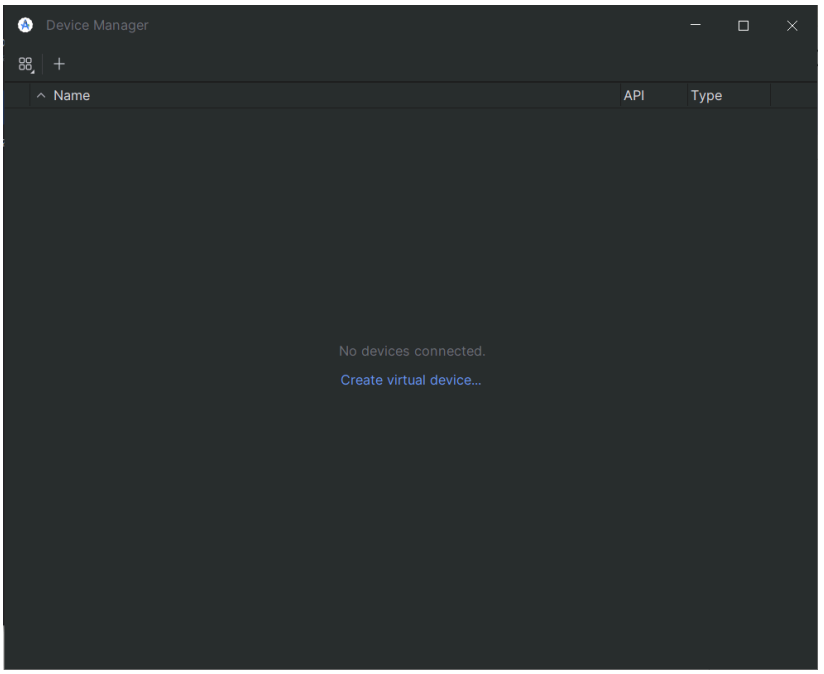
Langkah awal untuk menggunakan emulator android studio adalah mempersiapkan emulator pada aplikasi android studio. Buka aplikasi Android Studio lalu pilih ***more action***.

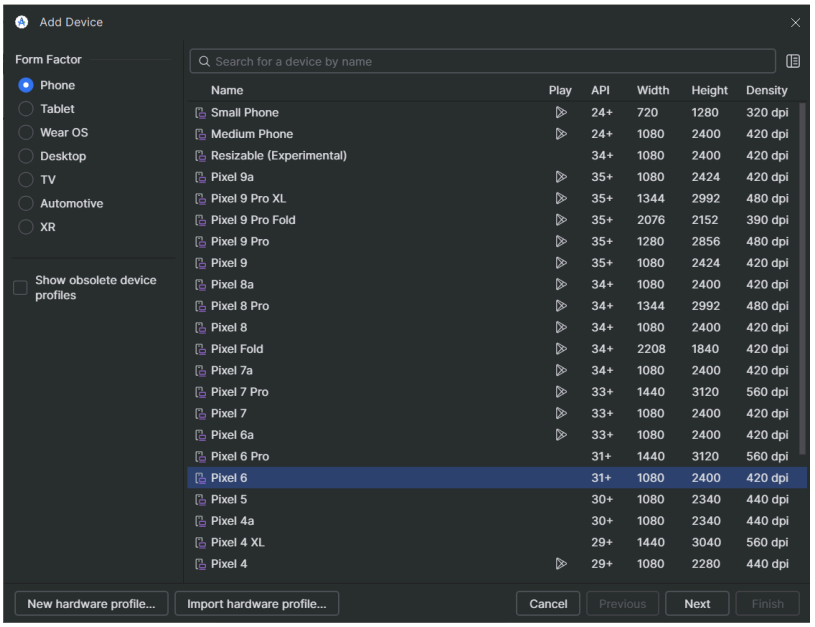


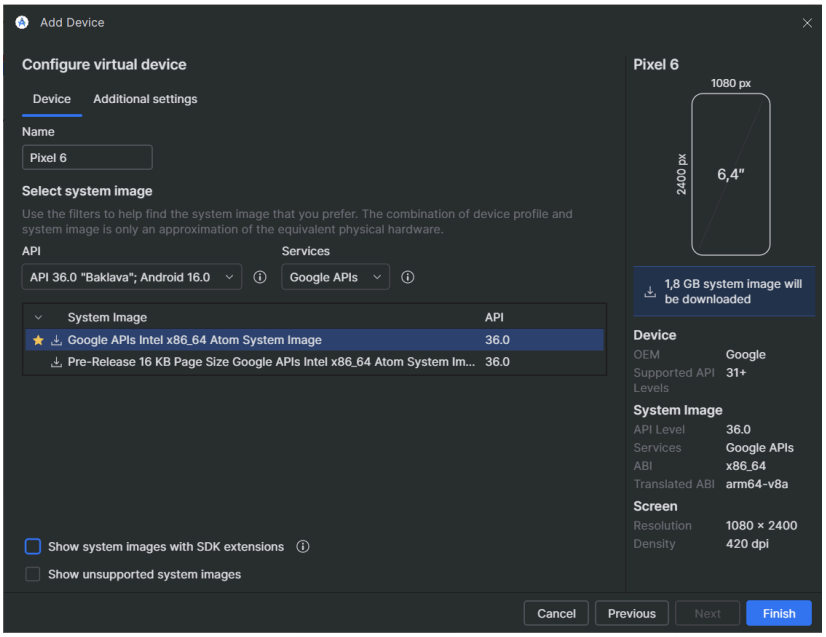
Lalu pada dropdown, pilih ***Virtual Device Manager***



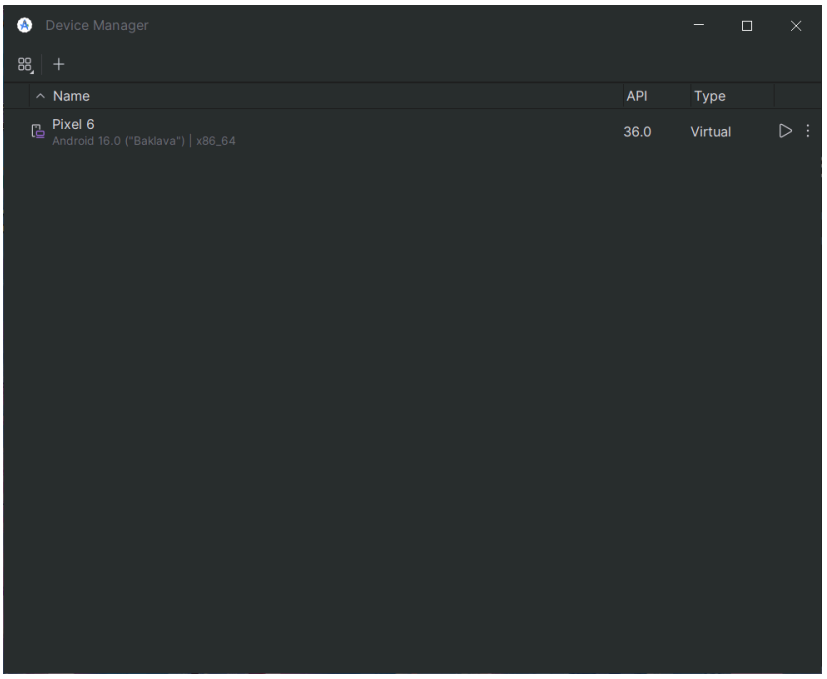
Create virtual device dan sesuaikan bentuk, nama serta spek yang diperlukan.



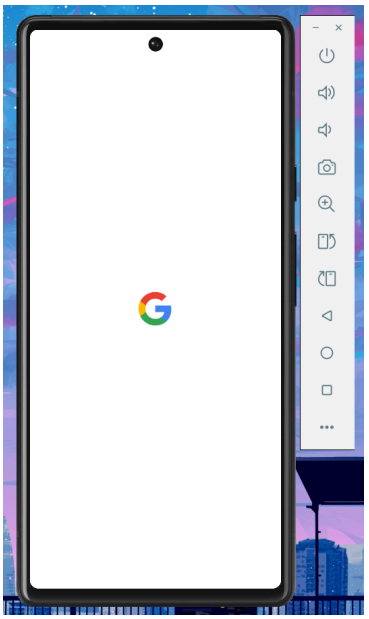




Selesai membuat emulator, klik tombol play yang terdapat di sebelah kanan list emulator yang telah dibuat.

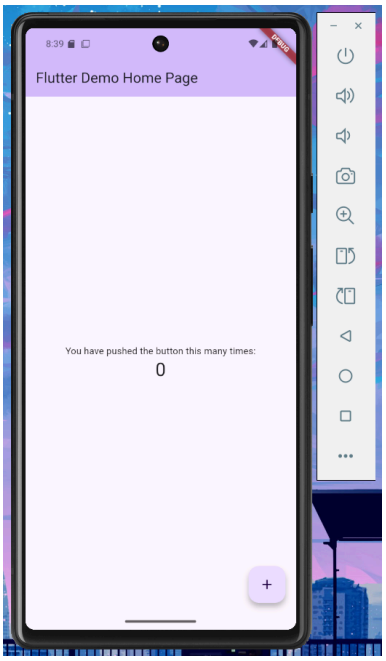


Tampilan awal emulator akan seperti berikut



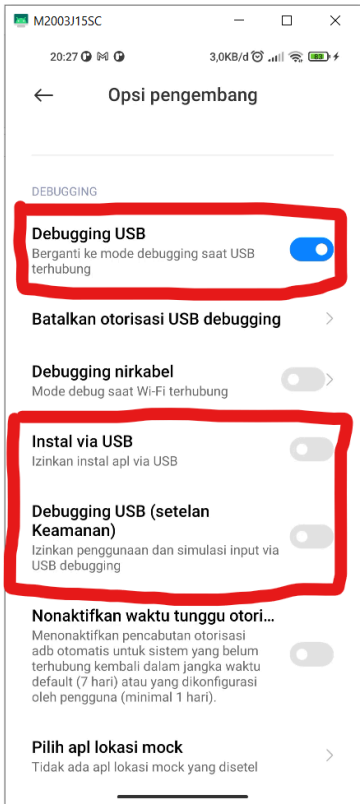
Saat emulator sudah masuk ke halaman awal android, kembali ke Visual Studio Code dan running kembali project Flutter. Kali ini tidak perlu memilih opsi karena Flutter akan otomatis membaca emulator yang tersedia.

Tampilan project Flutter yang telah berhasil dirunning pada emulator Android Studio.



4. **Device External**

Untuk menggunakan device external, kita harus mempersiapkan kabel USB serta device yang akan digunakan. Langkah awal set **Mode Debug, Install via USB dan Debugging USB (Setelan Keamanan)** pada opsi pengembang di device yang akan digunakan.



Untuk melihat apakah device sudah berhasil terkoneksi ke komputer, buka terminal lalu ketik ***adb devices***. Apabila ada device yang masuk di list, maka sambungan ke komputer telah berhasil.

```
PS D:\Tugas\Asprak\Pemrograman Perangkat Bergerak\Modul 2\Flutter\flutt
er_application_1> adb devices
List of devices attached
207750960409    device
```

Running kembali project Flutter, untuk menjalankan di device eksternal.

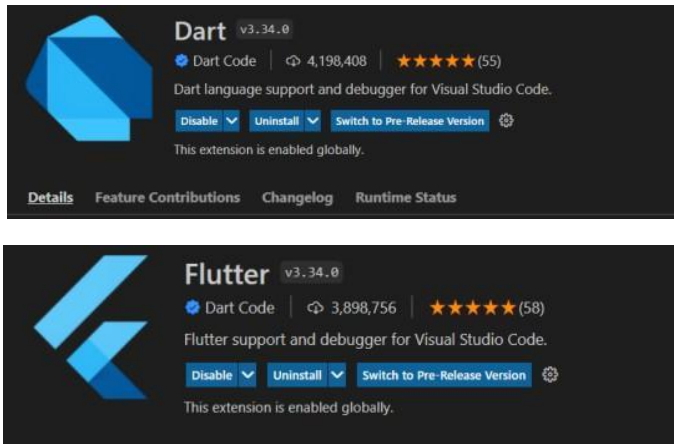
```
PS D:\Tugas\Asprak\Pemrograman Perangkat Bergerak\Modul 2\Flutter\flutt
er_application_1> flutter run
Launching lib\main.dart on M2003J15SC in debug mode...
Running Gradle task 'assembleDebug'...
/
```

Hello World Pada Flutter

Pada pengenalan Flutter kali ini, kita akan membuat Hello World sebagai permulaan ketika menggunakan Flutter. Ikuti langkah berikut untuk membuat Hello World di Flutter:

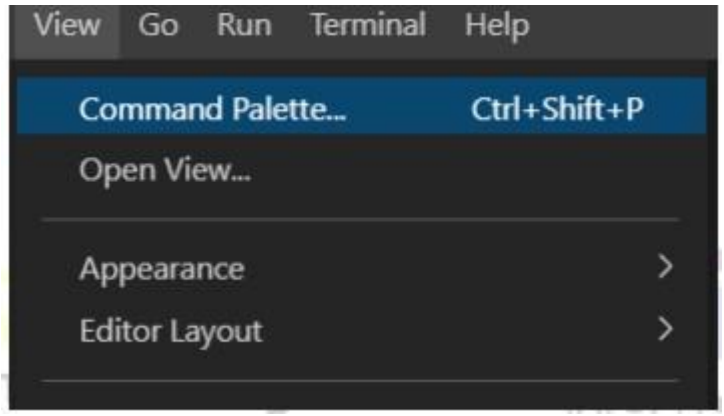
1. Buka visual studio code kalian untuk membuat project Flutter

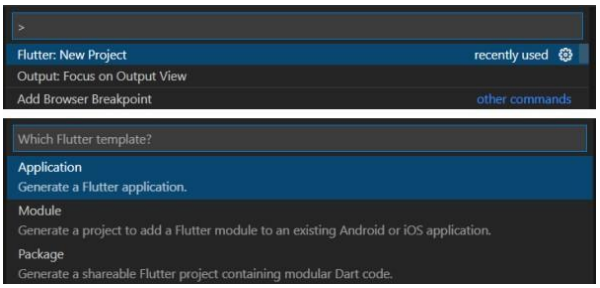
2. Pastikan extension Flutter dan Dart sudah terpasang di visual studio code



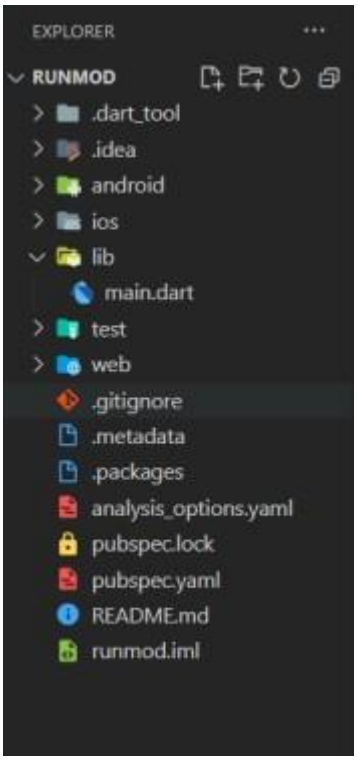
3. Lalu buat project Flutter pada menu View -> Command Palette

   -> lalu pilih Flutter: New Project -> pilih Application untuk membuat aplikasi dengan Flutter -> pilih folder yang ingin dijadikan sebagai project Flutter -> lalu beri nama project dan tekan Enter, tunggu hingga proses pembuatan project Flutter selesai



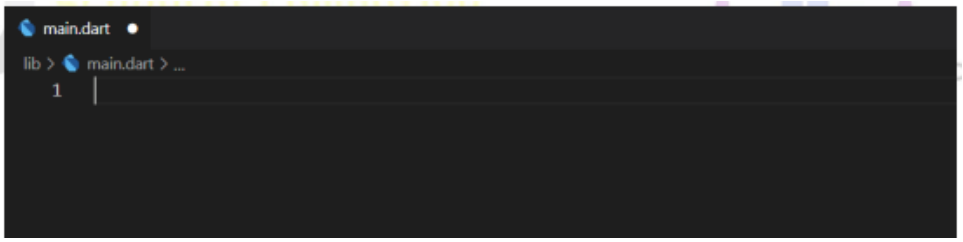


4. Ketika project selesai dibuat, maka tampilan folder akan seperti ini



5. Buka file main.dart untuk memulai membuat Hello World

6. Hapus semua kode yang ada pada main.dart



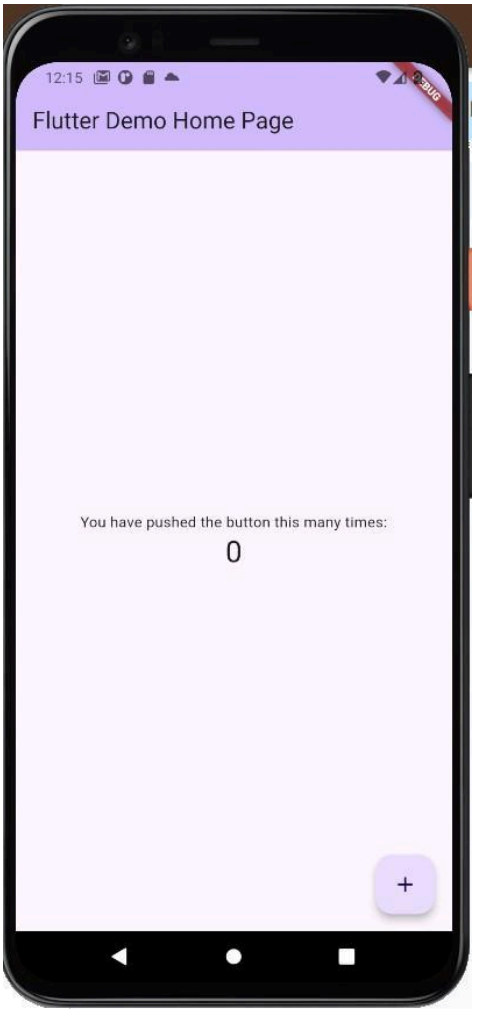
7. Lalu tambahkan barisan kode berikut untuk mebuat Hello World

```
// ignore_for_file: prefer_const_constructors import
'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
// This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: "Hello World",
      home: const MyHomePage(title: "Flutter Hello World
Page"),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({Key? key, required this.title}) :
super(key: key);
  final String title;
  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ),
      body: Center(
```

```
          child: Text(
            'Hello World',
          ),
        ),
      );
    }
}
```

Tugas Praktikum 2

1. Jelaskan apa itu Dart & Flutter beserta contoh widget yang ada pada Flutter.
2. Buatlah sebuah project Flutter.
3. Setelah project dibuat, jalankan di emulator atau pada *real device* (jika pada tampilan telah keluar project Flutter seperti gambar di bawah, maka telah berhasil).



4. Setelah berhasil, modifikasi halaman diatas untuk menampilkan biodata kalian, minimal 5 widget!! (bebas, buatlah sekreatif mungkin).

NOTE : Harap diperhatikan lagi dalam penulisan format nama serta folder pengumpulan tugasnya, terima kasih.