Google Developer Student Clubs
Institut Teknologi Telkom Surabaya

Intro to Flutter Development

Developing Mobile Apps on Multiplatform Technology

Ahmad Mu'min Faisal
Core Team, GDSC ITTS

Ahmad Mu'min Faisal

Core Team at Google Developer Student Clubs Institut Teknologi Telkom Surabaya

"A dedicated Flutter development enthusiast with a profound love for Linux and Free/Libre and Open-Source Software (FLOSS)"

- Division Head of PKM Center – Punggawa Inspiratif
- Lecturer Assistant of DSA, Comp. System Programming, and OOPs
- Head of Public Relation – Kelompok Linux Arek Suroboyo

- 2nd Place – IT Convert 2023
- 1st Place – PLAYBOX Season 4
- 3x Funded – PKM 2022-2023

ahmadmfaisal

fzl-22

contact@ahmadfaisal.online

fzl-22

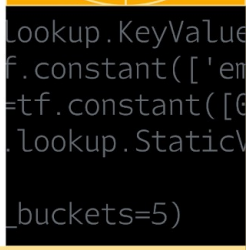
Google Developer Student Clubs

Ask me :)

bit.ly/ask-mobile-dev

Google Developer Student Clubs
Institut Teknologi Telkom Surabaya

Get to Know About Mobile Devices



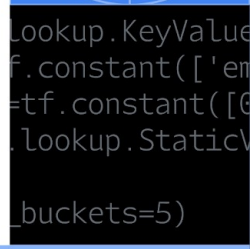
A mobile device is a **portable computing device** such as a smartphone or tablet computer

On each mobile device, there is an **operating system** embedded as an **interface** between **device hardware** and **application** on top of it.

OS	Vendor	Description
Android	Google	• using Linux kernel • open source (Android Open Source Project) • embedded on most mobile devices • Can be forked by other vendor, such as MIUI by Xiaomi.
iOS	Apple	• using XNU kernel • closed source • only embedded on Apple's iPhone and iPad

Google Developer Student Clubs
Institut Teknologi Telkom Surabaya

How to Build Application on Mobile Devices?



Native Development

By default, mobile apps is developed **natively**.

Native development means creating application for each **specific platform**, such as Android and iOS.

Native Development

The consequences is we must use **different** programming languages and tools for **each platform**.

Android	iOS
Java	Objective-C
Kotlin	SwiftUI

or

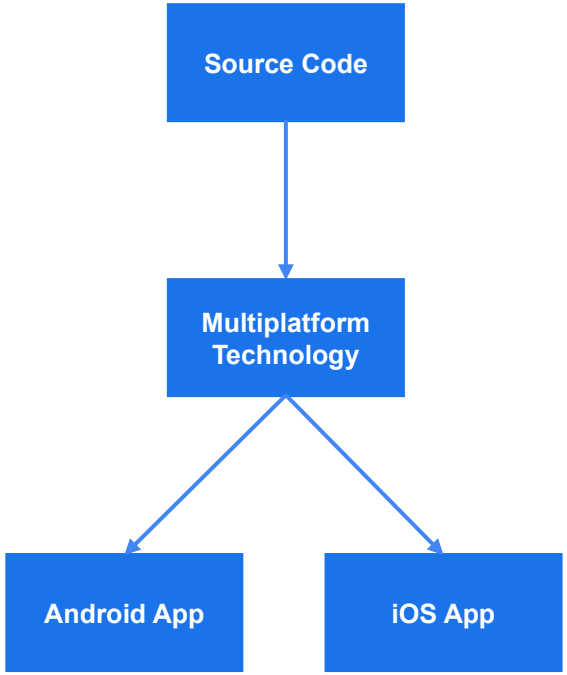
or

Multiplatform Development

Alternatively, mobile apps can be developed using **multiplatform technology**.

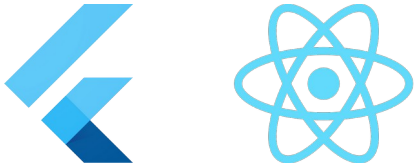
Application developed by this approach is expected to be **compatible** with **multiple operating system**.

The magic is, **it is**.



Multiplatform Development

Application developed this way requires a **framework** to do so. Currently, the most popular and widely-used multiplatform frameworks are **Flutter** and **React Native**.

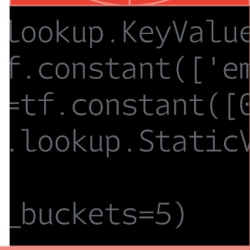


Flutter	React Native
Using Dart	Using Javascript
Created by Google in 2017	Created by Facebook in 2015
Can build Android, iOS, Linux, MacOS, Windows, and Web Application. Even on embedded device also	Can build iOS and Android application.
Compiled into target platform's native code	Rely on bridging with native modules
Near-native performance	Good, but not as good as Flutter
github.com/flutter/flutter	github.com/facebook/react-native

Google Developer Student Clubs

Institut Teknologi Telkom Surabaya

What is Flutter?



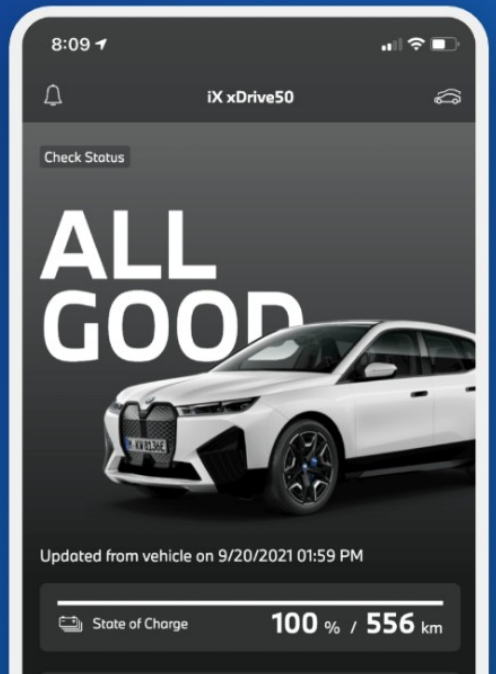
“**Flutter** is an open source framework by Google for building **beautiful**, **natively compiled**, **multi-platform applications** from a **single codebase**.”

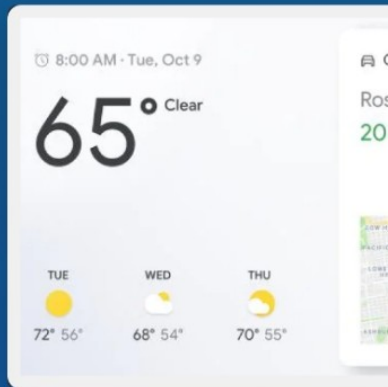
You can visit Flutter’s webpage at
flutter.dev

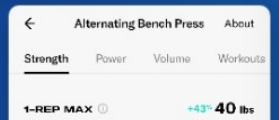
Flutter

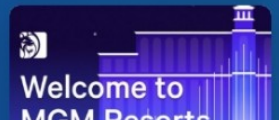
Multi-Platform Development Ecosystem Showcase Docs Get started

Build apps for any screen

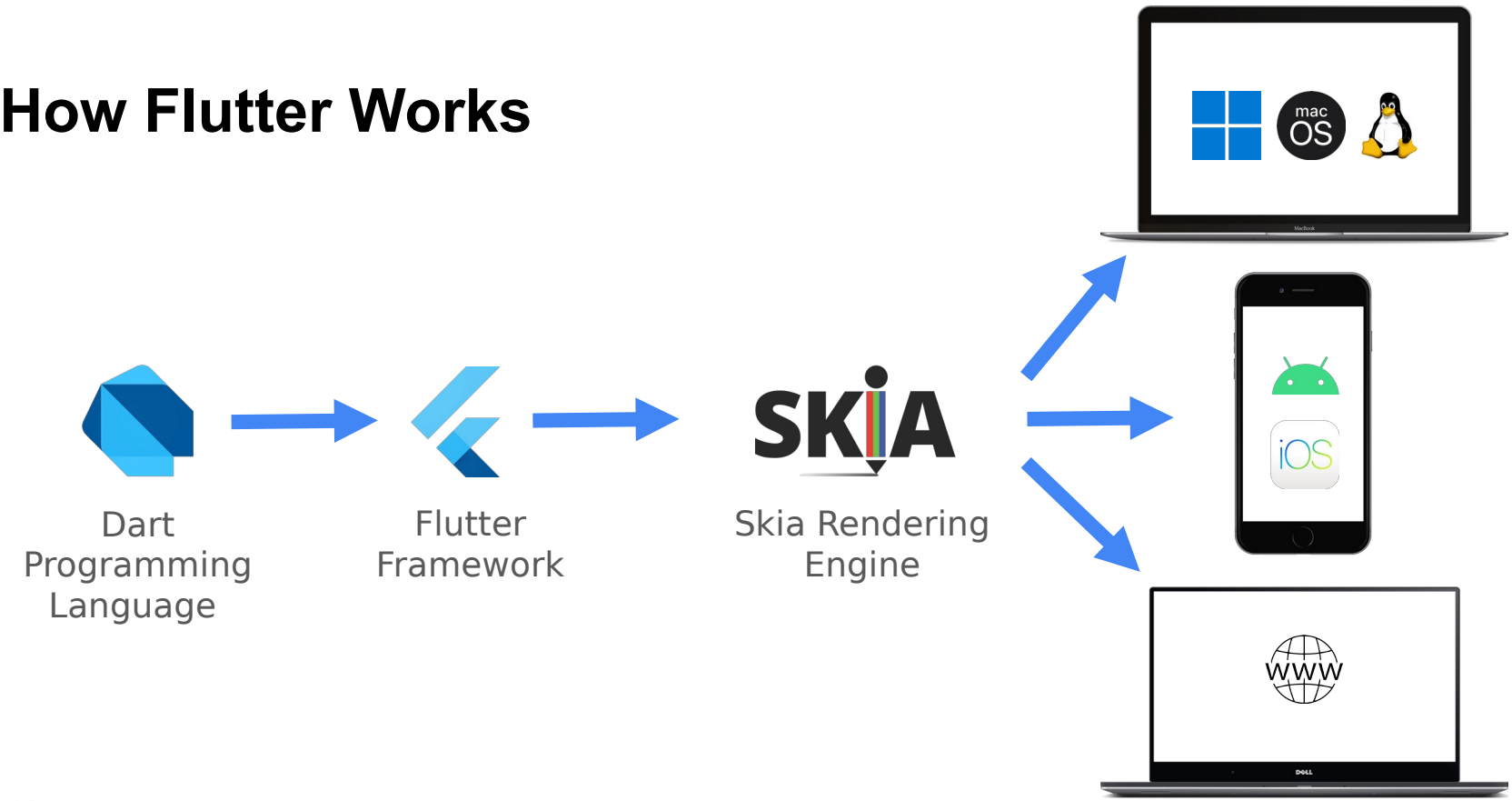




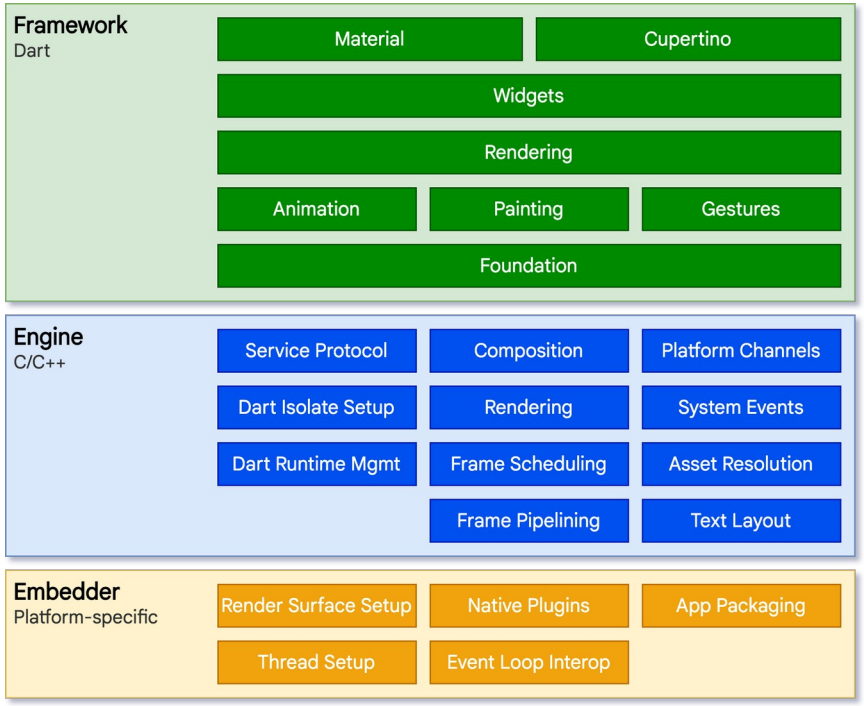




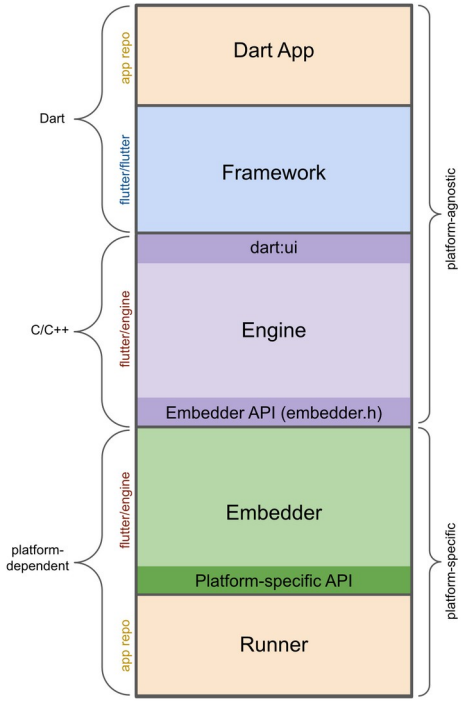
How Flutter Works



Flutter Architectural Layer

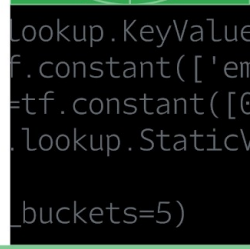


Flutter App Anatomy



Google Developer Student Clubs
Institut Teknologi Telkom Surabaya

Let’s Get Our Hand Dirty!



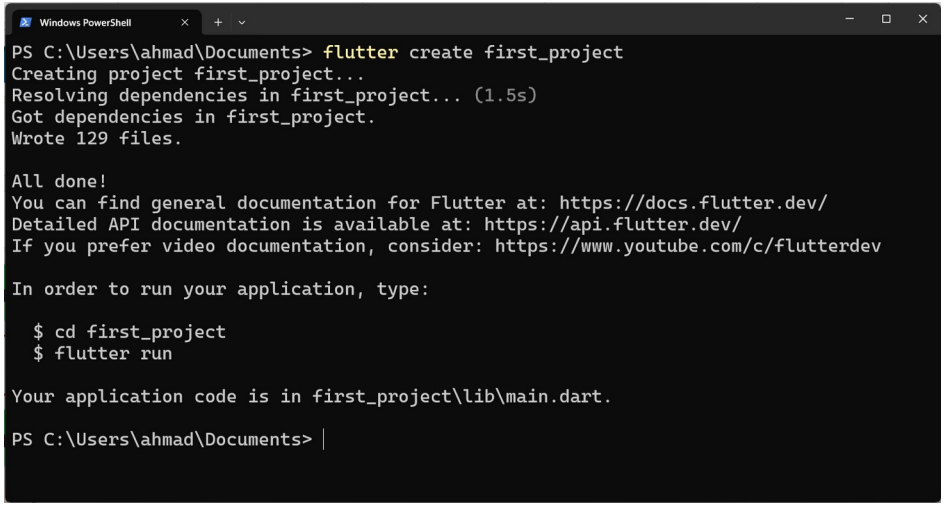
Create a Flutter Project

Flutter project can be easily created by running this following command on terminal:

```
flutter create {project_name}
```

For example,

```
flutter create first_project
```



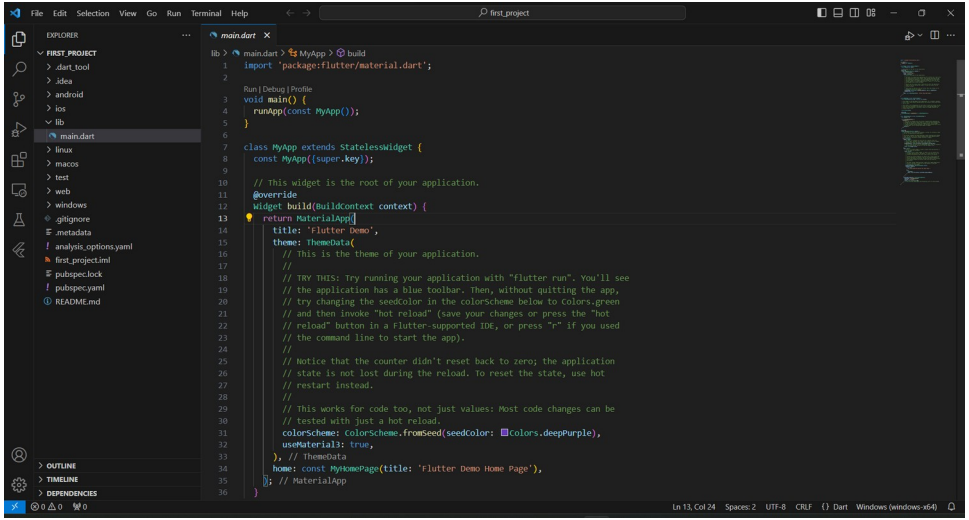
Now, Flutter initial project is successfully created.

Open Flutter Project

Open the project in Visual Studio Code by running this command:

```
code first_project
```

Make sure that **Dart** and **Flutter official extensions** on VS Code is **already installed.**



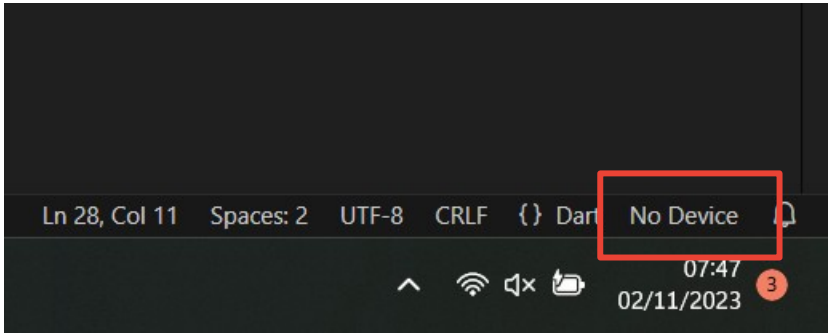
Note: your application code is in **`lib/main.dart`**

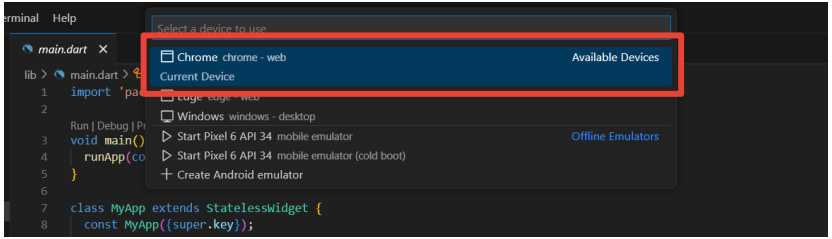
Select Target Platform (Web)

You can select target platform in bottom-right corner.

The pop up will appear to let you choose desired target platform.

Select **Chrome** or **Edge** to run Flutter on the web.

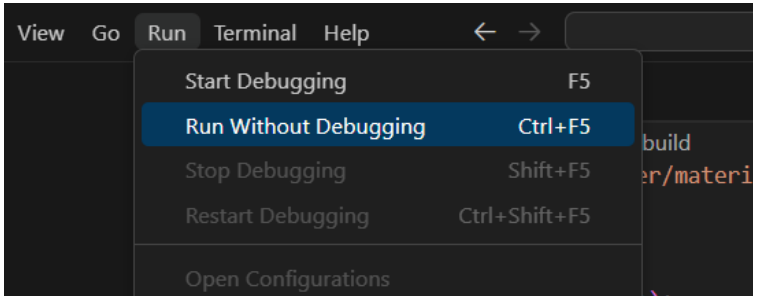


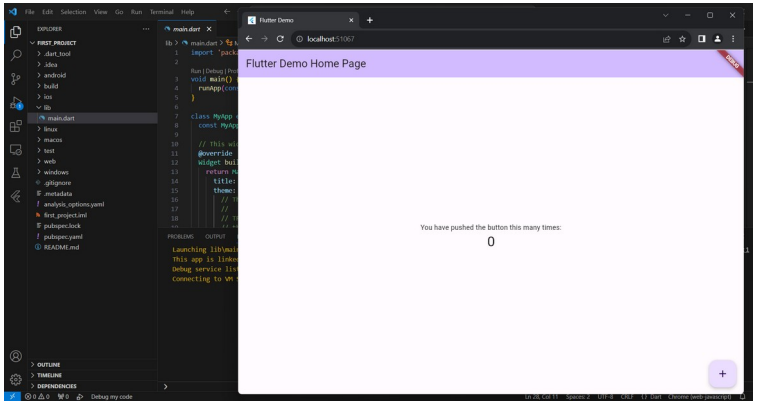


Run Flutter Project (Web)

It’s quite easy to run Flutter project. Just go to top bar and select **`Run > Run Without Debugging`**.

Just wait a little bit, and Flutter will serve you a web in a browser.





Select Target Platform (Desktop)

You can only select desktop OS **depending** on your **current host device.**

For example, your current OS need to be a GNU/Linux to develop Linux desktop application.

And, you are **required** to install current platform's **toolchain** before you develop desktop app there.

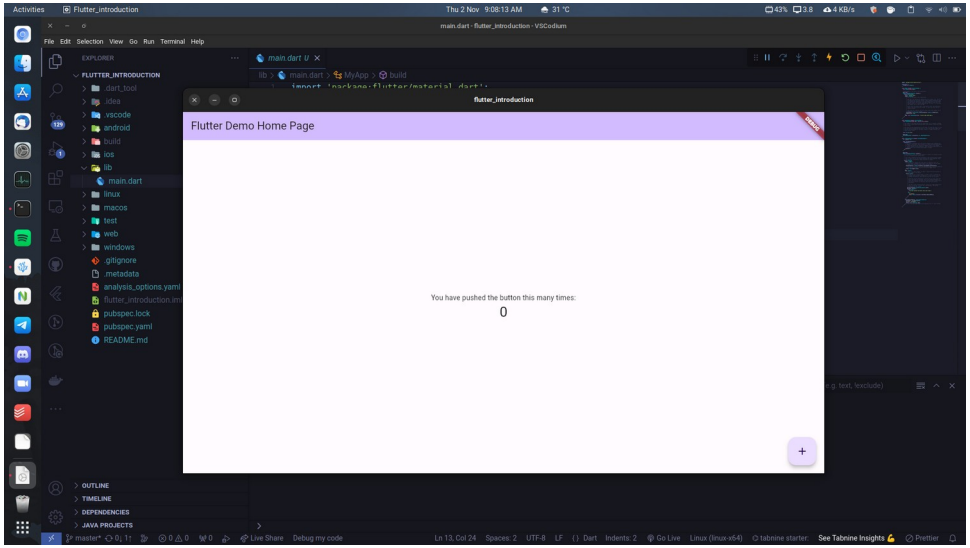
```
flutter_introduction on  master [↑$?] is 📦 1.0.0+1 …
→ flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.13.4, on Ubuntu 22.04.3 LTS 5.19.0-43-generic, locale
    en_US.UTF-8)
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.2)
[✓] Chrome - develop for the web
[✓] Linux toolchain - develop for Linux desktop
[✓] Android Studio (version 2022.3)
[✓] IntelliJ IDEA Community Edition (version 2023.2)
[✓] Connected device (2 available)
[✓] Network resources

• No issues found!
```

Run Flutter Project (Desktop)

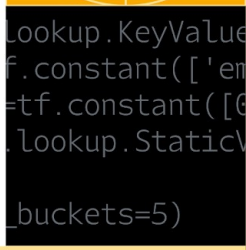
Then, you can run the project by selecting desktop platform (Linux, Windows, or MacOS) and click **Run > Run Without Debugging** in the top bar.

Then, you will have your flutter application as a Windows/Linux/MacOS desktop application.



Google Developer Student Clubs
Institut Teknologi Telkom Surabaya

Mobile Development using Flutter



What’ll We Build in Mobile Development Track?

We will build **Android application**, why not iOS?

Easy, because I **don’t have** any Apple’s product (MacOS, iPhone, and etc) 🏝

You can build iOS application **on your own** if you want to.

Requirements

Software Requirements:

- Android SDK & Android Emulator (from Android Studio)
- Visual Studio Code (including Dart and Flutter official extensions)

Hardware Requirements:

- x86_64 CPU architecture with support for a Windows Hypervisor
- 8 GB RAM or more
- 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
- 1280 x 800 minimum screen resolution

Create Android Emulator

You can create Android Emulator from Android Studio.

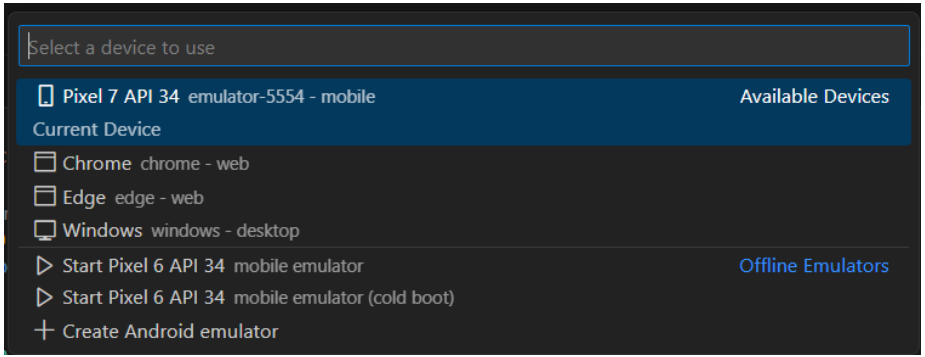
1. Open Android Studio
2. Select **More Actions > Virtual Device Manage > Create Device**.
3. Choose emulator profile with selected options in the table on the right
4. Click “**Finish**”
5. Run the newly created Android emulator

Configuration	Selected Option
Hardware	Phone, Pixel 7
System Image	API 34 (Google APIs)
Additional Configuration	Default

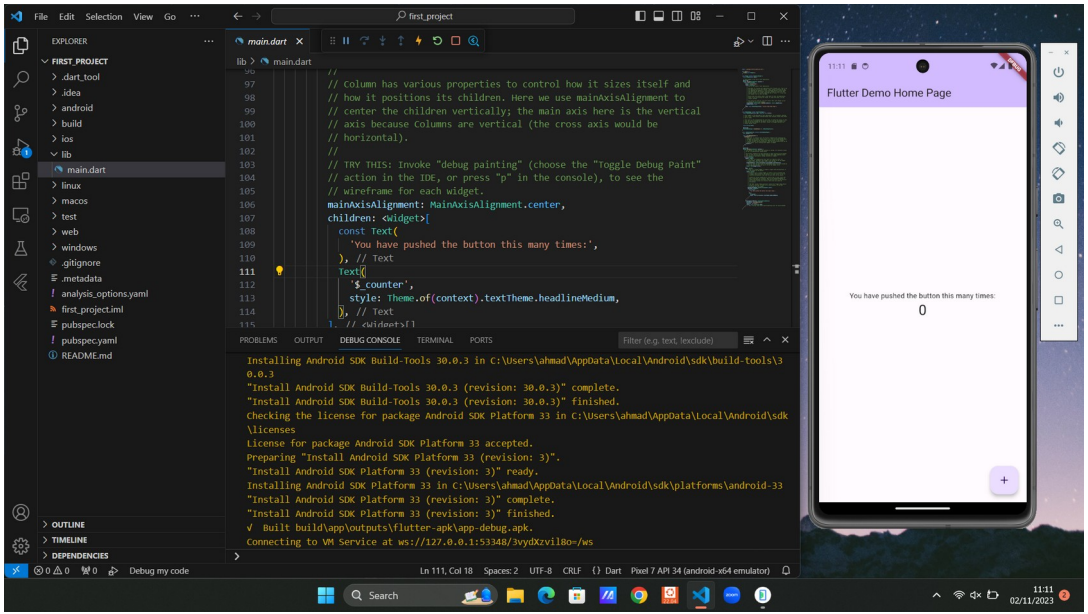
Run Flutter Project

To run Flutter project on Android emulator, do these following instructions:

1. Select your newly created emulator as target device
2. Click **Run > Run Without Debugging**
3. Wait a little bit, then your Flutter project is compiled on your emulator.



Run Flutter Project



Google Developer Student Clubs

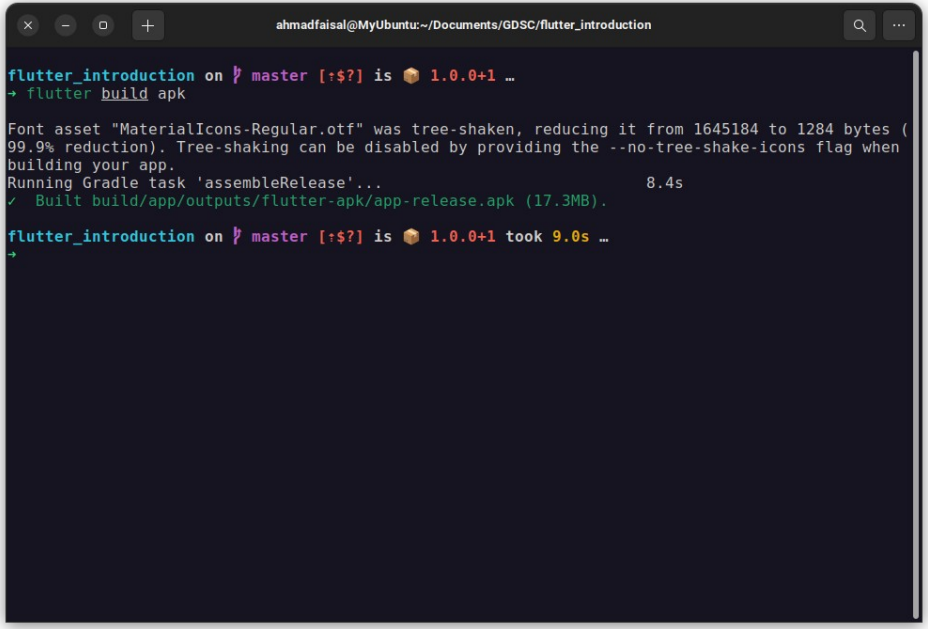
Let’s CODE and exploring Hot Reload and Hot Restart!

Let’s Package Your Flutter App

You can package your Flutter app to the form of **.apk** by running this command:

flutter build apk

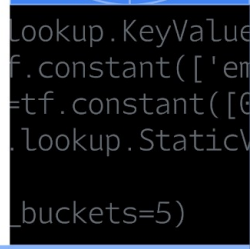
Your built **.apk** file will be stored in **build/app/outputs/flutter-apk/app-release.apk** so that you can install it on your own physical device.



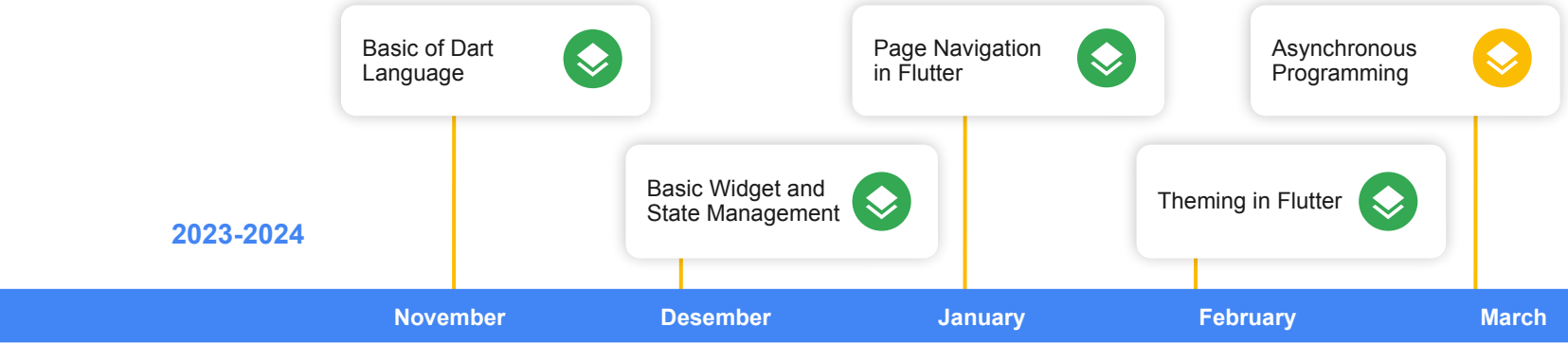
Google Developer Student Clubs

Institut Teknologi Telkom Surabaya

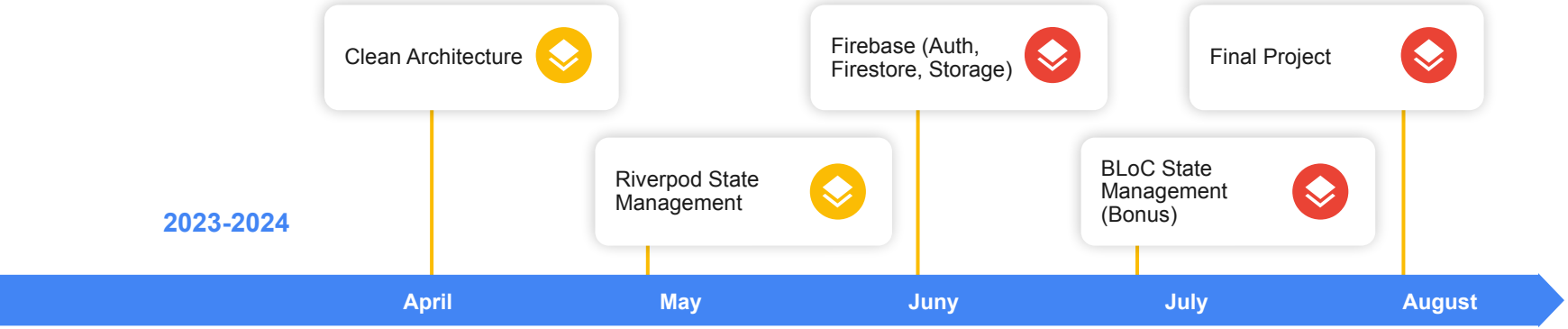
So, What's Next?



Learning Curriculum of Mobile Development Track



Learning Curriculum of Mobile Development Track



Google Developer Student Clubs

“You don’t have to be great to start, but you have to start to be great”

Zig Ziglar

Thank you!

Any Questions?

https://bit.ly/ask-mobile-dev

Google Developer Student Clubs