

README.md

# Layanan Mobile & Web di Kelas IV

## Pengertian Mobile

Mobile adalah perangkat elektronik yang bisa dibawa ke mana-mana dan dipakai untuk berkomunikasi atau menjalankan berbagai aplikasi. Contoh perangkat mobile termasuk smartphone, tablet, dan smartwatch.

## Smartphone

Smartphone adalah perangkat mobile yang bisa menjalankan banyak aplikasi dan digunakan untuk berbagai kebutuhan sehari-hari. Smartphone punya beberapa jenis sistem operasi, seperti:

- **Android:** Dikembangkan oleh Google, digunakan oleh banyak merek ponsel.
- **iOS:** Dikembangkan oleh Apple dan hanya digunakan pada iPhone.
- **Windows Phone:** Dikembangkan oleh Microsoft, tapi sekarang sudah tidak lagi dikembangkan.

## Pengertian Web Services

Web Services adalah sebuah API yang memungkinkan aplikasi-aplikasi di platform yang berbeda untuk saling terhubung dan berkomunikasi.

## Bahasa Pemrograman

### Platform Web

- **PHP:** Bahasa pemrograman server-side yang sangat populer untuk membuat website. PHP bisa berjalan di server seperti Apache dan Nginx.
- **Java:** Digunakan untuk membangun aplikasi web, terutama di platform J2EE, yang menyediakan alat dan lingkungan untuk menjalankan aplikasi bisnis besar.

- **JavaScript:** Awalnya dibuat untuk web client-side, tapi dengan Node.js, JavaScript sekarang bisa dipakai juga untuk server-side.
- **ASP.NET:** Framework dari Microsoft untuk membuat website yang dinamis, berjalan di atas server IIS (Internet Information Services).
- **Python:** Bahasa pemrograman serbaguna yang punya framework web populer seperti Django dan Flask.

## Platform Mobile

- **Android (Java):** Java digunakan di platform J2EE untuk mengembangkan aplikasi Android.
- **Kotlin:** Bahasa modern yang dirancang untuk Android, dapat berjalan di atas JVM (Java Virtual Machine), dan bisa bekerja bersama dengan Java.
- **iOS (Swift):** Bahasa yang dikembangkan oleh Apple untuk membuat aplikasi iOS, macOS, dan lainnya. Lebih cepat dan mudah dipelajari dibandingkan Objective-C.
- **Objective-C:** Bahasa yang dulu dipakai untuk membuat aplikasi iOS sebelum adanya Swift. Objective-C berbasis C dengan sintaks mirip Smalltalk.

## Platform Desktop

- **Delphi:** Bahasa pemrograman yang sering dipakai untuk membuat aplikasi desktop dengan bantuan IDE yang memudahkan pengembangan.
- **Java:** Selain untuk web, Java juga dipakai untuk aplikasi desktop melalui JavaFX, teknologi yang memudahkan pembuatan aplikasi dengan tampilan grafis.
- **Electron.js:** Framework yang memungkinkan pengembangan aplikasi desktop menggunakan HTML, CSS, dan JavaScript, bisa dijalankan di berbagai sistem operasi seperti Windows, macOS, dan Linux.

## Platform IoT (Embedded)

- **Arduino:** Platform open-source yang dipakai untuk membuat prototipe perangkat elektronik, menggunakan bahasa pemrograman C++.

## Database

# RDBMS (Relational Database Management System)

- **MySQL:** Database open-source yang paling populer di dunia, banyak digunakan untuk aplikasi web.
- **PostgreSQL:** Database open-source yang kuat dan handal, cocok untuk aplikasi yang membutuhkan keamanan dan konsistensi data.
- **SQLite:** Database ringan yang bisa dijalankan di berbagai platform, sering dipakai untuk aplikasi mobile.
- **Oracle:** Database yang kuat dan handal, banyak digunakan di perusahaan-perusahaan besar.
- **SQL Server:** Database dari Microsoft yang punya integrasi yang baik dengan platform Windows.

## NoSql (Not Only SQL)

- **MongoDB:** Database NoSQL yang berbasis dokumen, cocok untuk aplikasi yang membutuhkan fleksibilitas dan skalabilitas.
- **GraphQL:** Query language yang memungkinkan aplikasi client untuk meminta data dengan format yang diinginkan, sangat cocok untuk aplikasi mobile dan web.
- **Firebase:** Platform dari Google yang menyediakan berbagai layanan, seperti database, autentikasi, dan analitik, untuk aplikasi mobile dan web.

## Code Sederhana Menampilkan Hello World

### PHP

```
<?php  
echo "Hello, World!";  
?>
```

#### PHP Online Compiler

main.php		Output
1	echo "Hello, World!";	php /tmp/dTIR4fEBaq.php
2		echo "Hello, World!";

Menggunakan program php sangat mudah, hanya dengan menggunakan `<?php` dan `?>` kita bisa menampilkan Hello World.

# Java

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

## Programiz Java Online Compiler

Main.java	Output
<pre>1 // Online Java Compiler 2 // Use this editor to write, compile and run your Java code online 3 4 public class HelloWorld { 5     public static void main(String[] args) { 6         System.out.println("Hello, World!"); 7     } 8 }</pre>	<pre>java -cp /tmp/hZ4Rq80wWB/HelloWorld Hello, World!  === Code Execution Successful ===</pre>

Program Java menurut saya cukup sulit, karena harus menggunakan class dan method dan harus mendeklarasikan class terlebih dahulu.

# JavaScript

```
console.log("Hello, World!");
```

## Programiz JavaScript Online Compiler

main.js	Output
<pre>1 console.log("Hello, World!");</pre>	<pre>node /tmp/ZQu1c7aarf.js Hello, World!</pre>

JavaScript sangat mudah karna saya sudah terbiasa menggunakan JavaScript, hanya dengan menggunakan `console.log` kita bisa menampilkan Hello World.

# ASP.NET

```
using System;
```

```

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello, World!");
        }
    }
}

```

## OneCompiler C# Online Compiler

The screenshot shows the OneCompiler C# Online Compiler interface. On the left, the code editor displays the following C# code:

```

1 using System;
2
3 namespace HelloWorld
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             Console.WriteLine("Hello, World!");
10        }
11    }
12 }

```

On the right, the output section shows the result of the compilation and execution:

Output:  
Hello, World!

Program ASP.NET menurut saya cukup sulit, seperti Java, kita harus mendeklarasikan class terlebih dahulu dan menggunakan method untuk menampilkan Hello World.

## Python

```
print("Hello, World!")
```

## Programiz Python Online Compiler

The screenshot shows the Programiz Python Online Compiler interface. On the left, the code editor displays the following Python code:

```

1 # Online Python compiler (interpreter) to run Python online.
2 # Write Python 3 code in this online editor and run it.
3 print("Try programiz.pro")

```

On the right, the output section shows the result of the compilation and execution:

Output:  
Try programiz.pro  
=== Code Execution Successful ===

Python sangat mudah, hanya dengan menggunakan `print` kita bisa menampilkan Hello World.

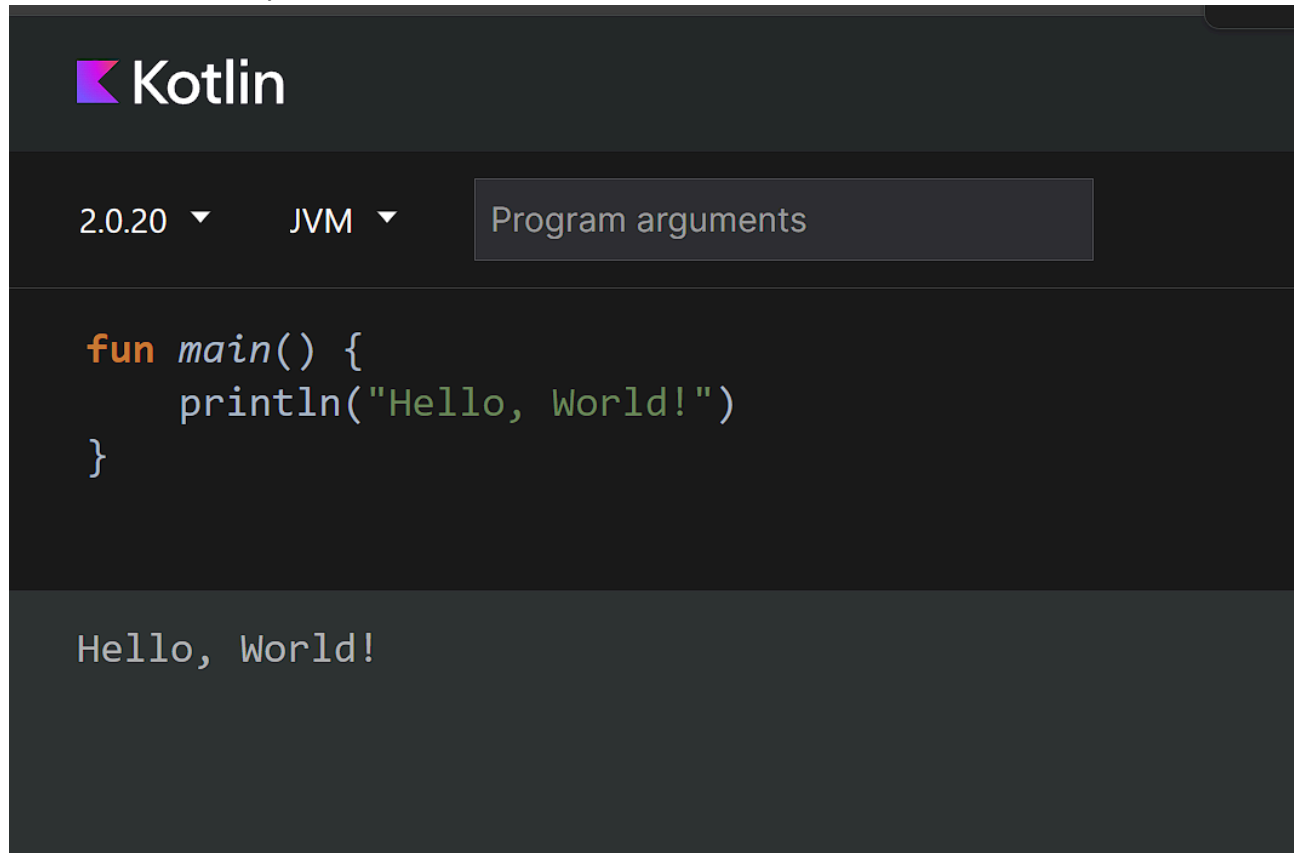
## Kotlin

```

fun main() {
    println("Hello, World!")
}

```

## Kotlin Online Compiler

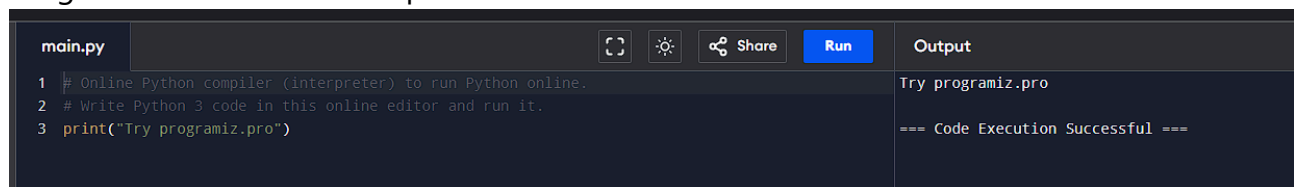


Kotlin menurut saya cukup mudah, hanya dengan menggunakan `fun` dan `println` kita bisa menampilkan Hello World.

## Swift

```
print("Hello, World!")
```

## Programiz Swift Online Compiler



Seperti python dan javascript, Swift sangat mudah, hanya dengan menggunakan `print` kita bisa menampilkan Hello World, lebih mirip dengan Python.

## Objective-C

```
#import <Foundation/Foundation.h>

int main(int argc, const char * argv[]) {
    @autoreleasepool {
        NSLog(@"Hello, World!");
    }
    return 0;
}
```

code tutorial dari : <https://www.digitalocean.com/community/tutorials/objective-c-hello-world-tutorial>

## Code Tutorial

```
Error(s):
1676451563/source.m: In function 'main':
1676451563/source.m:4:5: error: stray '@' in program
    @autoreleasepool {
    ^
1676451563/source.m:4:6: error: 'autoreleasepool' undeclared (first use in this function); did you mean 'NSAutoreleasePool'?
    @autoreleasepool {
    ^~~~~~
    NSAutoreleasePool
1676451563/source.m:4:6: note: each undeclared identifier is reported only once for each function it appears in
1676451563/source.m:4:22: error: expected ';' before '{' token
    @autoreleasepool {
    ^
1676451563/source.m:8:1: warning: control reaches end of non-void function [-Wreturn-type]
    }
    ^
```

Di beberapa compiler error tapi di online compiler ini bisa: OneCompiler Objective-C  
Online Compiler

HelloWorld.m
42t63ugga

```

1  #import <Foundation/Foundation.h>
2
3  int main(int argc, const char * argv[]) {
4      @autoreleasepool {
5          NSLog(@"Hello, World!");
6      }
7      return 0;
8  }
```

STDIN
  
Input for the program (Optional)
  


---

Output:
  
2024-09-24 17:25:57.121 a.out[3:3] Hello, World!

Objective-C menurut saya cukup sulit, saya masih belum terlalu paham dengan Objective-C, tapi saya sudah mencoba membuat Hello World dengan Objective-C.

## Delphi

```
program HelloWorld;

begin
```

```

    WriteLn('Hello, World!');
end.

```

## OneCompiler Delphi Online Compiler

HelloWorld.pas

```

1  program HelloWorld;
2
3  begin
4      WriteLn('Hello, World!');
5  end.

```

STDIN  
  
Input for the program ( Optional )  
  


---

Output:  
  
Hello, World!

Bahasa Delphi menurut saya cukup sulit karna belum pernah saya melihat code yang di tulis seperti ini, tapi saya sudah mencoba membuat Hello World dengan Delphi.

## C++

```

#include <iostream>

int main() {
    std::cout << "Hello, World!" << std::endl;
    return 0;
}

```

## programiz C++ Online Compiler

main.cpp

Share

Run

```

1  #include <iostream>
2
3  int main() {
4      std::cout << "Hello, World!" << std::endl;
5      return 0;
6  }

```

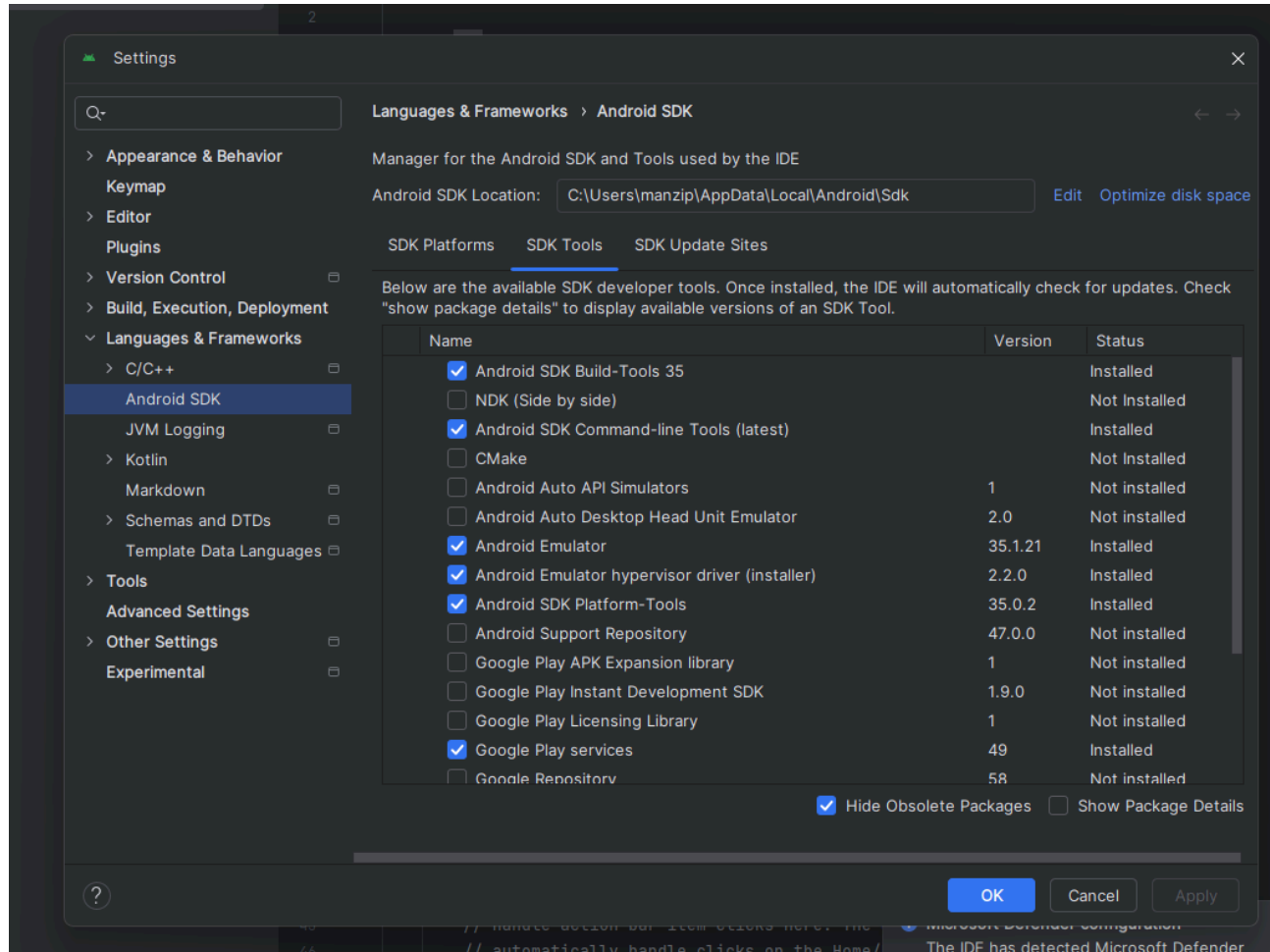
Output  
/tmp/tTcIcmVQyk.o  
Hello, World!  
  
=== Code Execution Successful ===

C++ menurut saya mudah mudah sulit, karna saya sudah pernah menggunakan C++ untuk Arduino, tapi saya masih belum terlalu paham dengan C++.



# Membuat Aplikasi Mobile Dengan Flutter

Berhubung saya sudah menginstall Android Studio pada laptop saya, maka selanjutnya saya akan menginstall SDK tools pada Android Studio.



Setelah itu, saya perlu menginstall Flutter. Saya mengambil dari GitHub: Flutter

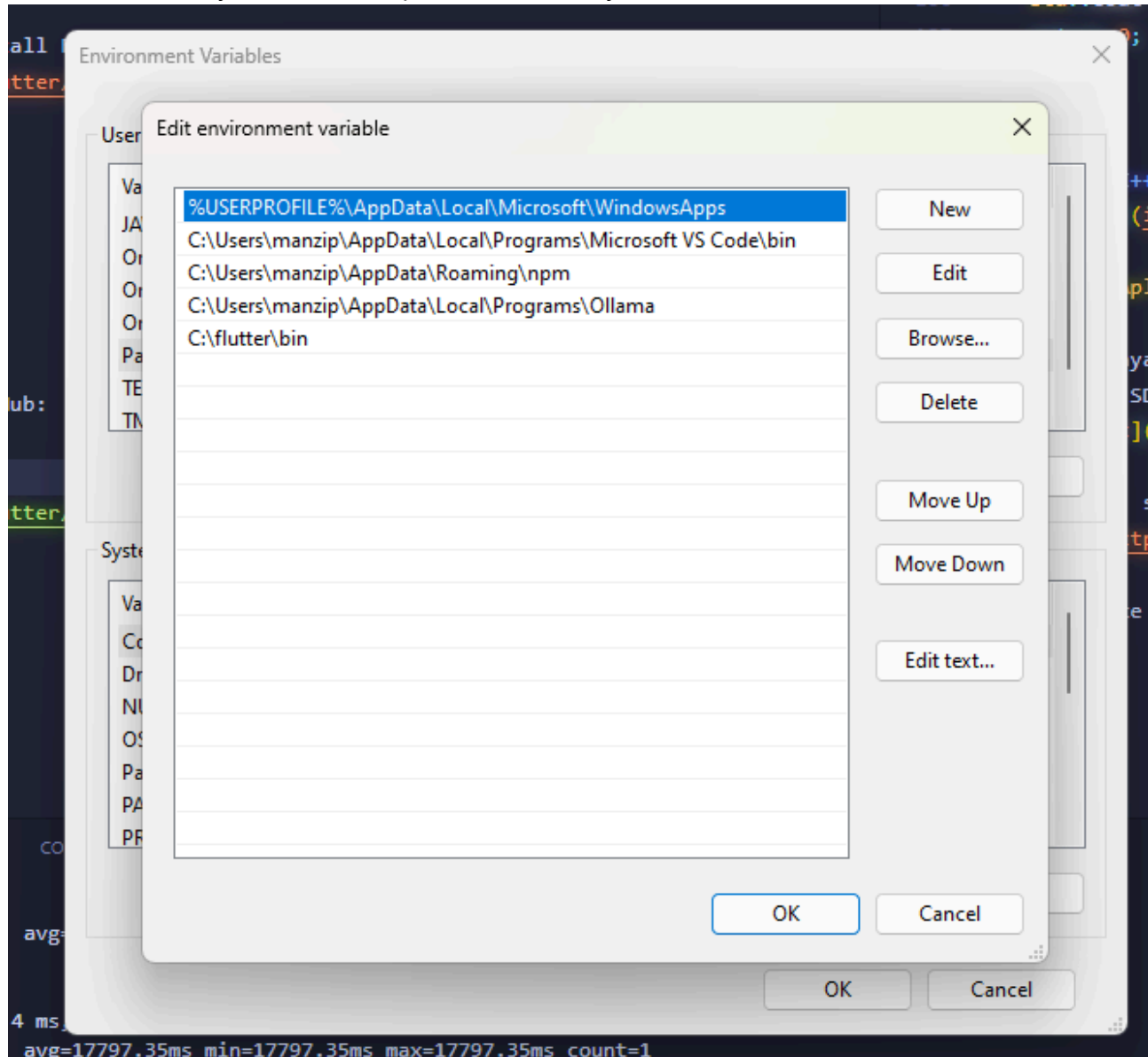
Lalu masuk ke disk C:

```
cd c:  
cd \
```

Kemudian clone Flutter dari GitHub:

```
git clone https://github.com/flutter/flutter.git -b stable
```

Setelah itu, saya perlu menambahkan path Flutter ke Environment Variables. Saya masuk ke Control Panel > System and Security > System > Advanced System Settings > Environment Variables. Lalu saya tambahkan path Flutter ke System Variables.



Lalu saya buka Command Prompt dan ketikkan:

```
flutter doctor --android-licenses
```

Dan saya setuju semua lisensi yang muncul.

Setelah itu, saya coba jalankan Flutter:

```
flutter doctor
```

```

C:\Windows\system32\cmd.e: X + v
[✓] Chrome - develop for the web
[X] Visual Studio - develop Windows apps
    X Visual Studio not installed; this is necessary to develop Windows apps.
      Download at https://visualstudio.microsoft.com/downloads/.
      Please install the "Desktop development with C++" workload, including all of its default components
[✓] Android Studio (version 2024.1)
[✓] VS Code (version 1.93.1)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 1 category.

C:\Users\manzip>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.24.3, on Microsoft Windows [Version 10.0.22631.4169], locale en-US)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 35.0.0)
[✓] Chrome - develop for the web
[X] Visual Studio - develop Windows apps
    X Visual Studio not installed; this is necessary to develop Windows apps.
      Download at https://visualstudio.microsoft.com/downloads/.
      Please install the "Desktop development with C++" workload, including all of its default components
[✓] Android Studio (version 2024.1)
[✓] VS Code (version 1.93.1)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 1 category.

C:\Users\manzip>

```

Setelah semua selesai, saya coba buat aplikasi Flutter sederhana

```
flutter create helloworld
```

```

C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\manzip\Documents\Kampus\Flutter>flutter create helloworld
Creating project helloworld...
Resolving dependencies in 'helloworld'... (1.4s)
Downloading packages...
Got dependencies in 'helloworld'.
Wrote 129 files.

All done!
You can find general documentation for Flutter at: https://docs.flutter.dev/
Detailed API documentation is available at: https://api.flutter.dev/
If you prefer video documentation, consider: https://www.youtube.com/c/Flutterdev

In order to run your application, type:

$ cd helloworld
$ flutter run

Your application code is in helloworld\lib\main.dart.

C:\Users\manzip\Documents\Kampus\Flutter>

```

Kemudian saya buka aplikasi Flutter yang sudah saya buat di Visual Studio Code.

```
cd helloworld
code .
```

Lalu saya edit file `main.dart` di folder `lib/main.dart` dari:

```
home: const MyHomePage(title: 'Flutter Demo Home Page'),
```

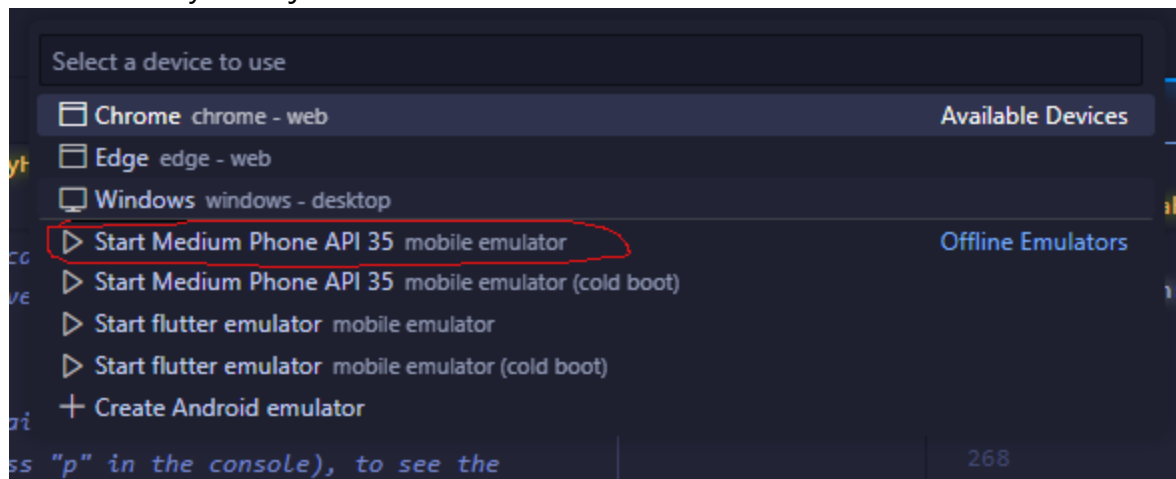
Menjadi:

```
home: const MyHomePage(title: 'Hello World'),
```

Berhubung saya tidak ingin merubah banyak hal terlebih dahulu, maka saya menambahkan 'Hello World' saja:

```
children: <Widget>[
  const Text("Hello World"),
  const Text(
    'You have pushed the button this many times:',
  ),
  Text(
    '$_counter',
    style: Theme.of(context).textTheme.headlineMedium,
  ),
],
```

Kemudian saya menjalankan android emulator melalui vscode



lalu

saya jalankan aplikasi Flutter yang sudah saya buat dengan menggunakan perintah:

```
flutter run
```

Dan ini adalah hasilnya:

