

**LAPORAN PRAKTIKUM**  
**REVIEW BAHASA PEMROGRAMAN DAN PENJELASAN**  
**INSTALASI FLUTTER BESERTA UJI COBA**

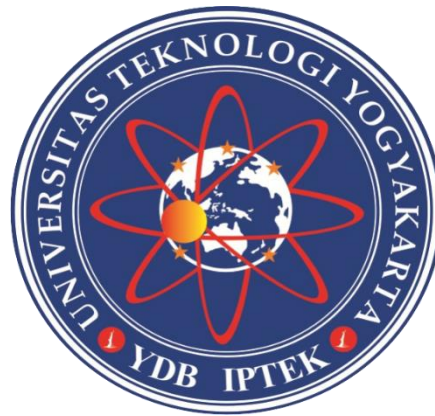
Disusun untuk Memenuhi Tugas Mata Kuliah Mobile & Web Service Praktik

Dosen Pengampu:

**Suyud Widiono, S.Pd., M.Kom.**

Asisten Dosen:

**Margareta Dyah Ayu Christiasih**



Disusun oleh:

Alfian Setya Dwi Saputra (5220411164)

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS SAINS & TEKNOLOGI**  
**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**2024**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>i</b>
<b>BAB I REVIEW BAHASA PEMROGRAMAN WEB, MOBILE, &amp; DESKTOP .....</b>	<b>1</b>
1.1. Web .....	1
1.1.1. Python.....	1
1.1.2. JSP (Java Server Pages).....	4
1.1.3. Node.js.....	6
1.1.4. PHP (Hypertext Preprocessor) .....	8
1.1.5. ASP (Active Server Pages).....	11
1.2. Mobile .....	14
1.2.1. Android .....	14
1.2.2. iOS (iPhone Operating System).....	19
1.3. Dekstop .....	25
1.3.1. .Net.....	25
1.3.2. C#.....	27
1.3.3. Delphi .....	30
<b>BAB II INSTALASI DAN UJI COBA SDK FLUTTER .....</b>	<b>34</b>
2.1. Teori Dasar.....	34
2.1.1. Android Studio .....	34
2.1.2. SDK Flutter .....	35
2.2. Praktikum.....	36
2.2.1. Instalasi SDK Flutter .....	36
2.2.2. Uji Coba .....	40
<b>REFERENSI.....</b>	<b>46</b>

# BAB I

## REVIEW BAHASA PEMROGRAMAN WEB, MOBILE, & DESKTOP

### 1.1. Web

#### 1.1.1. Python

- Pengertian

Python adalah bahasa pemrograman tingkat tinggi yang serbaguna dan mudah dibaca, serta mendukung berbagai paradigma pemrograman seperti berorientasi objek dan prosedural. Python dikenal karena sintaksnya yang sederhana, sehingga mempermudah pengembang dalam menulis kode yang bersih dan mudah dipelihara. Dalam konteks pengembangan web, Python menjadi sangat populer berkat framework powerful seperti Django dan Flask, yang menyederhanakan proses pengembangan aplikasi web, mulai dari membangun situs web sederhana hingga aplikasi berbasis skala besar.

Python sendiri tidak terbatas pada pengembangan web, namun dalam bidang ini, ia digunakan terutama di sisi server (back-end) untuk menangani logika bisnis, mengelola basis data, menangani request HTTP, serta berinteraksi dengan front-end yang biasanya dijalankan di browser pengguna. Salah satu keunggulan Python untuk web adalah kemampuan untuk mengintegrasikan berbagai teknologi dengan mudah berkat beragam pustaka dan modul yang tersedia, memungkinkan pengembang untuk menangani kebutuhan kompleks dengan lebih sedikit kode.

Python dapat digunakan untuk berbagai jenis aplikasi web, termasuk situs web dinamis, aplikasi berbasis API, aplikasi berbasis database, hingga platform real-time. Python bekerja sangat baik untuk mengelola data dalam skala besar dan bisa dengan mudah diintegrasikan dengan berbagai jenis database, baik SQL seperti MySQL atau PostgreSQL, maupun NoSQL seperti MongoDB. Kombinasi kemudahan penggunaan dan fleksibilitas ini menjadikan Python sangat efisien dan populer di kalangan pengembang web modern.

Python juga dikenal karena kemampuannya dalam menangani arsitektur aplikasi web yang scalable dan modular, terutama dengan framework seperti Django yang menyediakan pendekatan berbasis MVC (Model-View-Controller) untuk membangun aplikasi yang terstruktur dengan baik. Selain itu, Python memungkinkan pengembang untuk menerapkan server-side rendering, di mana

halaman web dirender di server sebelum dikirim ke browser pengguna. Hal ini dapat meningkatkan kecepatan halaman serta optimasi SEO, menjadikannya pilihan yang menarik bagi situs web yang memerlukan interaksi cepat dan real-time.

- Kelebihan

- Mudah Dipelajari: Python terkenal dengan sintaks yang sederhana, sehingga pemula dapat dengan mudah mempelajari dan menggunakannya untuk pengembangan web.
- Banyak Framework: Python memiliki beberapa framework web yang kuat dan populer, seperti Django dan Flask, yang memudahkan pengembangan aplikasi web dari tahap awal hingga produksi.
- Skalabilitas Tinggi: Dengan dukungan dari framework seperti Django, Python memungkinkan pengembangan aplikasi web yang sangat skalabel, mulai dari situs sederhana hingga aplikasi kompleks dengan lalu lintas tinggi.
- Komunitas dan Ekosistem Besar: Python memiliki ekosistem besar dengan banyak pustaka dan modul yang tersedia, sehingga pengembang dapat dengan mudah menemukan solusi untuk berbagai masalah pengembangan web.
- Keamanan: Framework Python seperti Django memiliki fitur keamanan bawaan yang membantu melindungi aplikasi dari ancaman umum seperti serangan SQL Injection dan Cross-Site Scripting (XSS).
- Mendukung Integrasi Database: Python memiliki dukungan yang luas untuk berbagai jenis basis data, baik SQL (seperti MySQL, PostgreSQL) maupun NoSQL (seperti MongoDB), yang memungkinkan fleksibilitas dalam pengelolaan data.
- Testing dan Debugging: Python mendukung unit testing dan debugging yang efektif, memungkinkan pengembang untuk dengan mudah menemukan dan memperbaiki bug dalam aplikasi web mereka.

- Kekurangan

- Kecepatan Eksekusi: Python mungkin sedikit lebih lambat dibandingkan bahasa seperti Node.js atau Go dalam menangani aplikasi web yang membutuhkan performa real-time atau pemrosesan yang intensif.
- Tidak Ideal untuk Aplikasi Real-time: Meskipun Python dapat digunakan untuk aplikasi real-time, bahasa ini kurang efisien jika dibandingkan dengan

platform seperti Node.js yang lebih optimal untuk penanganan permintaan simultan dan I/O non-blocking.

- Penggunaan Memori: Python cenderung menggunakan lebih banyak memori dibandingkan beberapa bahasa lain, yang bisa menjadi kendala pada aplikasi besar dengan kebutuhan memori tinggi.
- Kurva Pembelajaran untuk Framework: Meskipun sintaks Python sederhana, beberapa framework web seperti Django memiliki kurva pembelajaran yang curam karena strukturnya yang kompleks dan kaya fitur.

- Contoh Penggunaan Kode

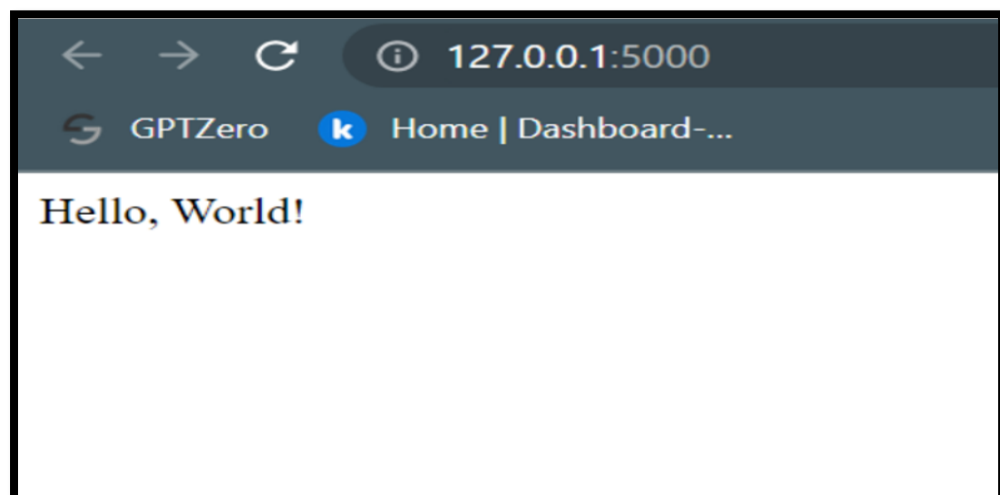
```
# Import Flask dari modul flask
from flask import Flask

# Inisialisasi aplikasi Flask
app = Flask(__name__)

# Rute utama (/) untuk halaman utama
@app.route("/")
def hello_world():
    return "Hello, World!"

# Jalankan aplikasi pada host lokal
if __name__ == "__main__":
    app.run(debug=True)
```

Hasil:



### 1.1.2. JSP (Java Server Pages)

- Pengertian

Java Server Pages (JSP) adalah teknologi berbasis Java yang digunakan untuk mengembangkan halaman web dinamis dengan cara menggabungkan kode Java dan HTML. JSP memungkinkan pengembang untuk menyisipkan kode Java langsung ke dalam halaman HTML, sehingga menghasilkan konten yang dapat berubah berdasarkan permintaan pengguna atau data yang diterima dari basis data. JSP merupakan bagian dari spesifikasi Java EE (Enterprise Edition) dan dirancang untuk mempermudah pengembangan aplikasi web dengan memisahkan logika aplikasi dari tampilan.

JSP bekerja dengan cara memproses halaman web di server sebelum mengirimkan hasilnya ke browser pengguna. Saat permintaan dibuat untuk halaman JSP, server akan mengkompilasi halaman tersebut menjadi servlet, yang merupakan kelas Java yang dapat dijalankan. Proses ini memungkinkan pengembang untuk menulis kode dinamis dengan sintaks yang mirip dengan HTML, sehingga lebih mudah dipahami dan dikelola.

Salah satu fitur utama dari JSP adalah kemampuannya untuk menggunakan tag libraries, yang memungkinkan pengembang untuk menyederhanakan dan mengorganisir kode dengan lebih baik. Dengan menggunakan tag JSP seperti JSTL (Java Server Pages Standard Tag Library), pengembang dapat mengakses berbagai fungsi seperti pengulangan, pengkondisian, dan pengolahan data, tanpa harus menulis banyak kode Java. Ini juga meningkatkan pemisahan antara logika dan presentasi dalam aplikasi, menjadikannya lebih mudah untuk dipelihara.

JSP sering digunakan dalam konteks aplikasi berbasis web yang memerlukan interaksi dengan pengguna, seperti aplikasi e-commerce, sistem manajemen konten (CMS), dan aplikasi intranet. Dengan integrasi yang kuat dengan Java dan kemampuan untuk berkolaborasi dengan framework lain seperti Spring dan Hibernate, JSP menawarkan platform yang fleksibel dan kuat untuk pengembangan aplikasi web yang kompleks.

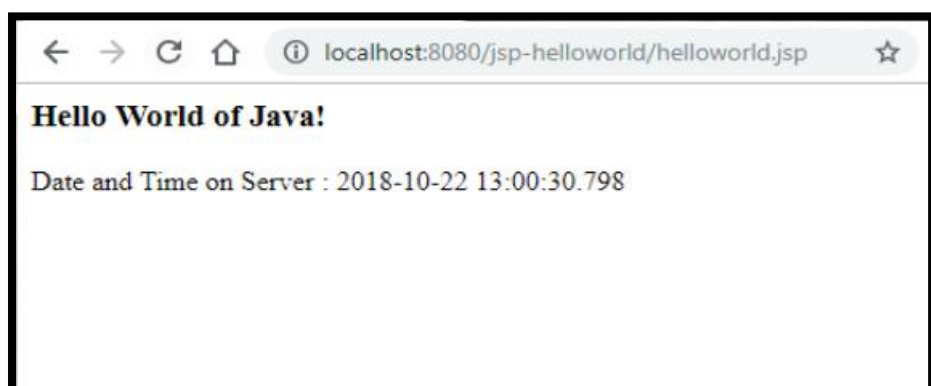
- Kelebihan

- Integrasi Java yang Kuat: Menggunakan Java sebagai basis, memungkinkan akses ke semua pustaka Java yang ada.

- Pemisahan Logika dan Tampilan: Memudahkan pengelolaan kode dengan memisahkan logika pemrograman dari markup HTML.
  - Tag Libraries: Penggunaan tag libraries membuat kode lebih bersih dan lebih mudah dibaca.
  - Dukungan untuk Ekstensi: Dapat dengan mudah diintegrasikan dengan framework dan teknologi lain dalam ekosistem Java.
- Kekurangan
    - Kompleksitas: Memerlukan pemahaman yang baik tentang Java dan konsep pemrograman berbasis objek.
    - Kinerja: Proses kompilasi halaman JSP menjadi servlet dapat menyebabkan overhead, terutama pada aplikasi dengan banyak permintaan.
    - Kurva Pembelajaran: Pengembang baru mungkin memerlukan waktu untuk beradaptasi dengan cara kerja JSP dan Java EE.
    - Keterbatasan dalam Pengembangan Cepat: Meskipun ada alat bantu, pengembangan aplikasi besar dengan JSP dapat menjadi lebih rumit dibandingkan dengan penggunaan framework modern lainnya.
  - Contoh Penggunaan Kode

```
<%@ page import="java.time.LocalDateTime" %>
<%@ page import="java.time.LocalDate" %>
<html>
<body>
    <h3>Hello World of Java!</h3>
    Date and Time on Server: <%= LocalDate.now() %> <%= LocalDateTime.now() %>
</body>
</html>
```

Hasil:



### 1.1.3. Node.js

- Pengertian

Node.js adalah platform open-source yang memungkinkan pengembang untuk menjalankan JavaScript di sisi server. Sebelum Node.js, JavaScript umumnya hanya digunakan di sisi klien (browser), namun dengan adanya Node.js, JavaScript dapat digunakan untuk membuat aplikasi server-side. Node.js didasarkan pada mesin JavaScript V8, yang merupakan mesin kompilasi JavaScript milik Google Chrome. Mesin V8 ini bertugas untuk mengeksekusi kode JavaScript dengan sangat cepat, karena mengubah JavaScript menjadi kode mesin (machine code) secara langsung.

Node.js memanfaatkan arsitektur event-driven, yang berarti ia bereaksi terhadap peristiwa (event) yang terjadi. Misalnya, saat ada request masuk ke server, Node.js merespons event ini dan menjalankan kode sesuai permintaan tersebut. Node.js juga memiliki arsitektur non-blocking I/O, yang memungkinkan server menangani banyak permintaan secara simultan tanpa menunggu satu permintaan selesai terlebih dahulu sebelum memproses yang lain. Pendekatan ini sangat ideal untuk aplikasi yang membutuhkan banyak operasi I/O, seperti membaca/menulis file atau berkomunikasi dengan database.

Node.js sering digunakan untuk membangun aplikasi web yang membutuhkan konektivitas real-time dan interaksi pengguna yang dinamis, seperti aplikasi chatting, sistem notifikasi, dan aplikasi streaming. Selain itu, karena sifatnya yang efisien dalam menangani I/O, Node.js juga sangat cocok untuk aplikasi yang menangani data dalam skala besar atau beban server yang tinggi.

Node.js memiliki fungsi utama untuk mengembangkan aplikasi web server-side dengan menggunakan JavaScript. Sebelumnya, JavaScript lebih sering digunakan di sisi klien untuk validasi form dan animasi, namun dengan Node.js, JavaScript dapat digunakan di sisi server. Platform ini memungkinkan pemrosesan asinkron dan memungkinkan pengembang menggunakan satu bahasa untuk aplikasi sisi server dan klien, menjadikannya lebih efisien dan cepat.

- Kelebihan

- Penggunaan Satu Bahasa: Node.js memungkinkan pengembang menggunakan JavaScript baik di sisi klien maupun server, sehingga mempermudah proses pembelajaran teknologi baru.



- Reuse of Code: Mendukung penggunaan ulang kode antara klien dan server, memudahkan kolaborasi pengembangan.
  - Dukungan JSON: Memudahkan pengelolaan data, terutama dari API pihak ketiga yang sering menggunakan format JSON.
  - Asinkron I/O: Node.js memungkinkan pemrosesan paralel tanpa harus menunggu operasi selesai, meningkatkan performa aplikasi.
  - Dukungan Database NoSQL: Platform ini terintegrasi dengan baik dengan database seperti MongoDB dan CouchDB yang juga menggunakan JavaScript.
  - Menggunakan Mesin V8: Node.js selalu mengikuti standar ECMAScript terbaru, memastikan kompatibilitas dengan browser modern.
  - Sumber Terbuka: Karena open-source, Node.js berkembang dengan cepat, terutama melalui kontribusi komunitas di GitHub.
  - Cepat Dipelajari: Bagi pengembang yang sudah familiar dengan JavaScript di sisi klien, mempelajari sisi server dengan Node.js lebih mudah.
  - Server-Side Rendering: Memungkinkan rendering di sisi server dengan JavaScript, meningkatkan kecepatan dan SEO.
  - Aplikasi Realtime: Node.js mendukung pengembangan aplikasi realtime, seperti chat dan notifikasi.
- Kekurangan
    - Single Threaded: Node.js berbasis single-thread, yang bisa menjadi kendala jika aplikasi harus menangani tugas CPU intensif.
    - Kurva Pembelajaran: Meskipun berbasis JavaScript, pengembang yang terbiasa dengan model pemrograman tradisional mungkin merasa perlu adaptasi dengan arsitektur asinkron.
    - Tidak Cocok untuk Proses Berat: Node.js lebih cocok untuk aplikasi berbasis I/O tinggi daripada pemrosesan yang berat pada CPU.
    - Callback Hell: Karena menggunakan pola asinkron, kode Node.js bisa menjadi sulit diikuti akibat banyaknya callback yang bersarang.

- Contoh Penggunaan Kode

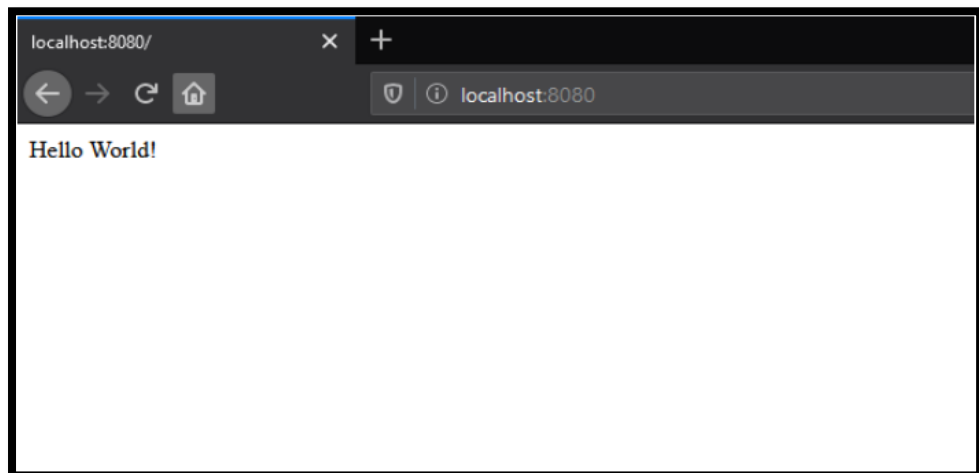
```
// Memuat modul HTTP bawaan Node.js
const http = require('http');

// Membuat server
const server = http.createServer((req, res) => {
  // Mengatur header respons
  res.statusCode = 200; // Status OK
  res.setHeader('Content-Type', 'text/plain'); // Tipe konten teks biasa

  // Mengirimkan output "Hello World!"
  res.end('Hello World!\n');
});

// Menjalankan server pada port 3000
const port = 3000;
server.listen(port, () => {
  console.log(`Server berjalan di http://localhost:${port}/`);
});
```

Hasil:



#### 1.1.4. PHP (Hypertext Preprocessor)

PHP (Hypertext Preprocessor) adalah bahasa pemrograman open-source yang secara khusus dirancang untuk pengembangan aplikasi web dan skrip yang berjalan di sisi server (server-side). PHP memungkinkan pembuatan situs web dinamis dan interaktif, di mana konten halaman web dapat diubah berdasarkan input pengguna, berinteraksi dengan basis data, dan memberikan keluaran yang berbeda setiap kali halaman diakses. PHP digunakan secara luas karena fleksibilitasnya,

kompatibilitasnya dengan berbagai server web, dan kemampuannya untuk menangani tugas-tugas server-side secara efisien.

Pada awalnya, PHP diciptakan oleh Rasmus Lerdorf pada tahun 1994 sebagai serangkaian skrip CGI (Common Gateway Interface) untuk melacak kunjungan ke websitenya. Seiring berjalannya waktu, bahasa ini berkembang menjadi salah satu teknologi server-side terpopuler, digunakan di lebih dari 75% situs web di seluruh dunia, termasuk platform terkenal seperti WordPress, Wikipedia, dan Facebook pada masa-masa awalnya. Saat ini, PHP telah berevolusi menjadi bahasa pemrograman yang matang dengan berbagai fitur dan kemampuan yang canggih.

Salah satu kekuatan utama PHP adalah kemampuannya untuk berjalan pada hampir semua sistem operasi, termasuk Windows, Linux, macOS, dan Unix. PHP juga dapat diintegrasikan dengan berbagai server web, seperti Apache, Nginx, dan Microsoft IIS, menjadikannya pilihan yang sangat fleksibel dalam lingkungan pengembangan.

PHP bekerja dengan cara memproses kode di server, yang kemudian menghasilkan halaman HTML yang dikirim ke browser pengguna. Pengguna hanya melihat hasilnya, tanpa melihat kode PHP yang sebenarnya. Kode PHP dapat disisipkan langsung ke dalam HTML, memungkinkan pengembang untuk mencampur antara kode server-side dan markup HTML, yang memudahkan dalam membuat halaman web yang dinamis.

PHP juga mendukung integrasi dengan berbagai sistem manajemen basis data seperti MySQL, PostgreSQL, SQLite, dan Oracle, sehingga pengembang dapat dengan mudah membuat aplikasi web berbasis data. Selain itu, PHP mendukung berbagai format data dan protokol, seperti HTTP, HTTPS, FTP, serta format data populer seperti JSON dan XML, yang membuatnya sangat fleksibel dalam berkomunikasi dengan API atau layanan eksternal.

PHP telah berkembang pesat dan mendukung pemrograman berorientasi objek (OOP), menjadikannya lebih modular dan mudah untuk pengembangan aplikasi besar dan terstruktur. Seiring berkembangnya teknologi web, berbagai framework PHP seperti Laravel, Symfony, dan CodeIgniter telah diciptakan untuk membantu pengembang membuat aplikasi dengan cepat dan efisien, sambil tetap mengikuti standar pengembangan yang baik.

Salah satu alasan utama mengapa PHP sangat populer di kalangan pengembang adalah kemudahan penggunaannya. PHP memiliki sintaks yang mirip dengan bahasa

pemrograman populer lainnya seperti C, Perl, dan Java, sehingga memudahkan bagi pengembang yang sudah familiar dengan bahasa tersebut untuk belajar PHP. Ditambah dengan dokumentasi yang sangat lengkap dan komunitas yang besar, PHP menawarkan lingkungan pengembangan yang ramah bagi pemula sekaligus cukup kuat untuk pengembangan aplikasi web yang kompleks.

PHP sering digunakan dalam pengembangan situs e-commerce, Content Management System (CMS) seperti WordPress dan Joomla, serta berbagai aplikasi web yang memerlukan interaksi intensif dengan basis data. Dalam pengembangan modern, PHP juga sering digunakan bersama dengan teknologi front-end seperti HTML, CSS, dan JavaScript untuk membangun aplikasi web penuh yang interaktif.

- Kelebihan

- Mudah Dipelajari: PHP memiliki sintaks yang sederhana dan mudah dipahami, sehingga cocok untuk pemula dalam pengembangan web.
- Komunitas Besar: Dengan komunitas global yang sangat besar, PHP memiliki banyak sumber daya, tutorial, serta dukungan dari pengembang lain.
- Platform Independen: PHP dapat berjalan di berbagai sistem operasi seperti Windows, Linux, dan macOS, serta kompatibel dengan banyak server web seperti Apache dan Nginx.
- Integrasi Database: PHP mendukung hampir semua sistem manajemen basis data, baik SQL (seperti MySQL) maupun NoSQL (seperti MongoDB), yang membuatnya sangat fleksibel untuk aplikasi berbasis data.
- Framework yang Kaya Fitur: PHP memiliki banyak framework seperti Laravel, Symfony, dan CodeIgniter yang memudahkan pengembangan aplikasi dengan lebih cepat dan terstruktur.
- Penggunaan yang Luas: PHP digunakan oleh banyak situs web terkenal, termasuk Facebook, Wikipedia, dan WordPress. Lebih dari 75% situs web di internet menggunakan PHP dalam beberapa bentuk.

- Kekurangan

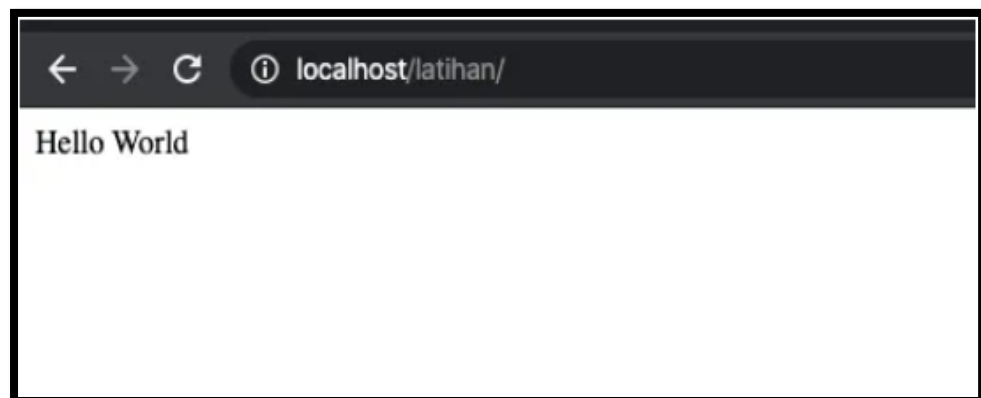
- Performa: Meskipun PHP cocok untuk aplikasi web dinamis, performa PHP cenderung lebih rendah dibandingkan bahasa pemrograman lain seperti Node.js atau Python saat menangani aplikasi yang sangat besar atau kompleks.

- Keamanan: Jika tidak ditangani dengan benar, PHP rentan terhadap berbagai ancaman keamanan seperti SQL injection dan XSS (Cross-Site Scripting). Meskipun PHP memiliki mekanisme untuk melindungi dari serangan ini, pengembang perlu menerapkan praktik keamanan terbaik.
- Model Pemrograman Lama: PHP secara historis memiliki reputasi buruk terkait dengan model pemrogramannya yang lama dan kurang terstruktur, meskipun ini sudah diperbaiki dengan rilis PHP versi terbaru dan framework modern seperti Laravel.
- Kesulitan untuk Aplikasi Real-Time: PHP tidak dirancang secara native untuk aplikasi real-time seperti chat atau notifikasi, sehingga memerlukan teknologi tambahan seperti WebSocket atau kombinasi dengan Node.js untuk mencapai tujuan tersebut.

- Contoh Penggunaan Kode

```
<?php
// Menampilkan teks "Hello, World!" di browser
echo "Hello, World!";
?>
```

Hasil:



### 1.1.5. ASP (Active Server Pages)

- Pengertian

Active Server Pages (ASP) adalah teknologi server-side scripting yang dikembangkan oleh Microsoft yang memungkinkan pembuatan halaman web dinamis dan interaktif. ASP pertama kali diperkenalkan pada tahun 1996 sebagai bagian dari Internet Information Services (IIS), server web milik Microsoft.

Teknologi ini memungkinkan pengembang untuk menyisipkan skrip dalam halaman web yang akan dieksekusi di sisi server sebelum dikirimkan ke klien (browser). Skrip ASP biasanya ditulis menggunakan VBScript atau JScript, meskipun bahasa lain juga bisa digunakan melalui komponen ActiveX.

ASP berfungsi dengan cara memproses kode di server, yang artinya pengguna hanya akan menerima hasil akhir berupa HTML, CSS, atau JavaScript di browser mereka, tanpa dapat melihat kode server yang sebenarnya. Hal ini memberikan keuntungan dari segi keamanan dan fleksibilitas dalam membangun aplikasi web yang dinamis, misalnya untuk memproses input pengguna, mengakses database, atau menampilkan data berdasarkan preferensi pengguna secara real-time.

ASP sangat terkenal di era 1990-an hingga awal 2000-an karena integrasinya yang kuat dengan ekosistem Microsoft, terutama dengan Microsoft Access dan SQL Server. Salah satu fitur utama dari ASP adalah kemampuannya untuk mengakses data melalui ActiveX Data Objects (ADO), sehingga memungkinkan pengembangan aplikasi berbasis database dengan cepat dan efisien. Dengan kemampuan ini, ASP menjadi pilihan populer untuk membuat situs web e-commerce, forum, dan aplikasi berbasis data lainnya.

ASP juga mendukung state management, yang memungkinkan penyimpanan informasi pengguna selama sesi berlangsung. Dengan menggunakan Session Object, pengembang dapat melacak aktivitas pengguna selama mereka berinteraksi dengan aplikasi, misalnya untuk fungsi login atau keranjang belanja di situs e-commerce.

Meskipun ASP klasik masih digunakan hingga kini dalam beberapa aplikasi legacy, sebagian besar pengembang modern telah beralih ke ASP.NET, versi lebih canggih dari ASP yang berbasis pada .NET Framework. ASP.NET menawarkan lebih banyak fitur dan bahasa pemrograman yang lebih modern seperti C#, memungkinkan pembuatan aplikasi web yang lebih kompleks dan efisien.

- Kelebihan

- Integrasi dengan Windows Server: ASP bekerja dengan sangat baik pada server yang menjalankan sistem operasi Windows dan IIS, membuatnya cocok untuk pengembangan di lingkungan berbasis Microsoft.

- Simpel dan Mudah Dipelajari: ASP klasik mudah dipelajari dan cocok bagi pengembang pemula yang ingin membuat aplikasi web dinamis dengan cepat menggunakan VBScript atau JScript.
  - Komponen yang Dapat Digunakan Kembali: ASP mendukung penggunaan komponen COM (Component Object Model) untuk memperluas fungsionalitas situs web, memungkinkan pengembang untuk menambahkan fitur tanpa menulis kode dari awal.
  - Dukungan Basis Data: ASP memiliki dukungan kuat untuk berkomunikasi dengan berbagai basis data, termasuk Microsoft Access, SQL Server, dan database lainnya melalui ADO (ActiveX Data Objects).
  - Server-side Scripting: Kode ASP dijalankan di server, sehingga pengguna tidak bisa melihat atau mengakses kode sumber seperti halnya JavaScript di sisi klien.
- Kekurangan
    - Bergantung pada Platform: ASP sangat terkait dengan platform Microsoft, terutama Windows dan IIS, sehingga kurang fleksibel untuk digunakan di lingkungan server non-Windows.
    - Kinerja yang Lebih Lambat: Dibandingkan teknologi modern seperti ASP.NET atau Node.js, kinerja ASP klasik relatif lebih lambat karena teknologi ini sudah tua dan tidak dioptimalkan untuk kebutuhan aplikasi web modern.
    - Dukungan Terbatas: Microsoft telah berhenti mengembangkan ASP klasik sejak munculnya ASP.NET. Meskipun masih didukung dalam IIS, tidak ada fitur baru atau peningkatan performa yang diberikan untuk ASP klasik.
    - Kurangnya Fitur Modern: ASP klasik tidak mendukung fitur modern seperti MVC (Model-View-Controller), yang sudah menjadi standar untuk aplikasi web modern.

- Contoh Penggunaan Kode

```
<!DOCTYPE html>
<html>
<body>

<p>ASP can output plain text:</p>

<%response.write("Hello World!")%>

</body>
</html>
```

Hasil:

ASP can output plain text:  
Hello World!

## 1.2. Mobile

### 1.2.1. Android

#### ❖ Kotlin

- Pengertian

Kotlin adalah bahasa pemrograman statis yang berjalan di atas Java Virtual Machine (JVM) dan dikembangkan oleh JetBrains, perusahaan yang sama yang menciptakan IntelliJ IDEA, salah satu Integrated Development Environment (IDE) terpopuler. Kotlin pertama kali diumumkan pada tahun 2011 dan sejak itu terus mengalami perkembangan hingga pada tahun 2017 Google mengumumkan dukungan resmi Kotlin sebagai bahasa pemrograman untuk pengembangan aplikasi Android, bersamaan dengan Java. Hal ini menjadikan Kotlin sebagai bahasa utama untuk Android bersama Java, mengingat sifatnya yang interoperabel dan kemudahan penggunaannya.



Salah satu alasan utama popularitas Kotlin dalam pengembangan Android adalah interoperabilitasnya yang penuh dengan Java. Pengembang dapat menulis kode Kotlin dan Java dalam proyek yang sama, dan kedua bahasa tersebut akan berjalan tanpa hambatan. Ini sangat penting bagi tim pengembang yang ingin beralih secara bertahap dari Java ke Kotlin tanpa harus menulis ulang seluruh kode dari awal. Selain itu, Kotlin dirancang untuk mengatasi beberapa kelemahan Java, seperti verbosity (kode Java yang sering kali terlalu panjang dan berulang), serta masalah null pointer exception yang terkenal di Java. Kotlin menyediakan fitur null safety bawaan yang secara signifikan mengurangi risiko kesalahan runtime akibat variabel null.

Kotlin tidak hanya memudahkan penulisan kode yang lebih ringkas dan bersih, tetapi juga mendukung berbagai paradigma pemrograman, termasuk pemrograman berorientasi objek (OOP) dan pemrograman fungsional. Ini memberi pengembang fleksibilitas dalam memilih gaya pemrograman yang sesuai dengan kebutuhan proyek mereka. Kotlin juga mendukung fitur modern seperti higher-order functions, lambdas, extension functions, dan coroutines, yang semuanya dirancang untuk membuat pengembangan aplikasi lebih produktif dan efisien. Coroutines, misalnya, memberikan pendekatan yang lebih sederhana dan bersih untuk menangani tugas-tugas asynchronous seperti operasi jaringan atau interaksi dengan basis data, yang sering kali membutuhkan manajemen thread yang rumit di Java.

Keunggulan lainnya adalah kemampuan Kotlin untuk dikompilasi tidak hanya ke bytecode Java, tetapi juga ke JavaScript dan native code melalui Kotlin/Native, yang memungkinkan pengembangan lintas platform. Ini berarti pengembang dapat menggunakan Kotlin tidak hanya untuk pengembangan Android, tetapi juga untuk aplikasi server, aplikasi desktop, bahkan aplikasi web, menjadikannya bahasa yang serbaguna. Seiring berkembangnya komunitas Kotlin, semakin banyak pustaka dan framework yang mendukung bahasa ini. Kotlin juga didukung oleh sejumlah besar tool dan integrasi yang ditawarkan oleh JetBrains, seperti IDE Android Studio, yang secara default mendukung pengembangan Kotlin sejak versi 3.0. Dukungan resmi dari Google dan JetBrains, ditambah dengan ekosistem yang terus berkembang, membuat Kotlin menjadi pilihan yang sangat relevan dan kuat untuk pengembangan Android di masa kini.

- Kelebihan

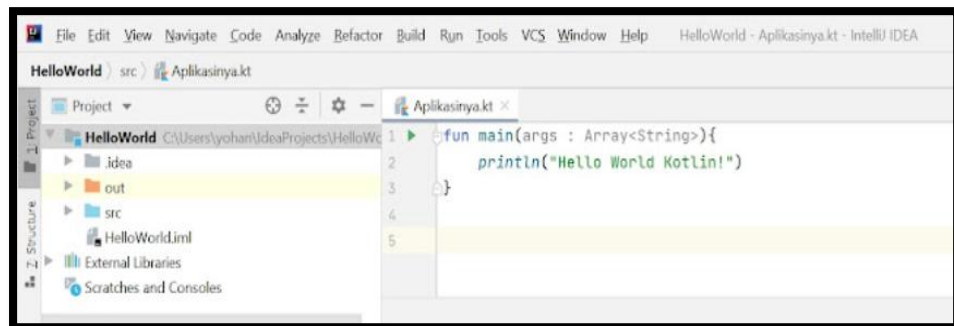
- Sintaks Lebih Sederhana: Kotlin dirancang untuk mengurangi verbosity yang sering ditemui pada Java. Kode Kotlin lebih ringkas dan mudah dibaca.
- Null Safety: Kotlin menyediakan sistem yang mencegah kesalahan NullPointerException, salah satu kesalahan yang paling umum di Java.
- Interoperabilitas dengan Java: Kotlin bisa digunakan bersamaan dengan Java dalam satu proyek tanpa masalah, sehingga pengembang dapat menggunakan Kotlin di proyek Java yang sudah ada.
- Coroutines untuk Pemrograman Asynchronous: Kotlin mendukung coroutines yang memudahkan penanganan operasi asynchronous seperti jaringan atau interaksi dengan database secara efisien dan tanpa kode yang kompleks.
- Ekspresif dan Fleksibel: Kotlin mendukung berbagai gaya pemrograman seperti pemrograman berorientasi objek dan pemrograman fungsional, memungkinkan pengembang untuk memilih gaya yang sesuai dengan kebutuhan mereka.
- Dukungan Resmi Google: Dengan dukungan resmi Google, Kotlin mendapatkan dokumentasi dan toolchain yang sangat kuat untuk pengembangan Android.
- Pengelolaan Memori yang Baik: Kotlin mengelola memori dengan lebih efisien, sehingga aplikasi Android yang dikembangkan cenderung lebih stabil dan memiliki kinerja yang lebih baik.

- Kekurangan

- Kurva Pembelajaran untuk Pengembang Java: Meskipun Kotlin mirip dengan Java, pengembang yang terbiasa dengan Java mungkin perlu waktu untuk beradaptasi dengan fitur-fitur baru yang ditawarkan oleh Kotlin.
- Komunitas yang Masih Berkembang: Meskipun sudah sangat populer, ekosistem Kotlin masih terus berkembang dan mungkin belum memiliki banyak pustaka atau alat yang sebesar ekosistem Java.

- Waktu Kompilasi yang Lebih Lambat: Dalam beberapa kasus, waktu kompilasi Kotlin bisa lebih lambat dibandingkan Java, terutama untuk proyek besar.
- Kesulitan dalam Debugging: Karena Kotlin adalah bahasa yang relatif baru, pengembang mungkin menemukan bahwa debugging dan pengujian bisa sedikit lebih rumit, terutama ketika berurusan dengan interoperabilitas Java.

- Contoh Penggunaan Kode



Hasil:



## ❖ Dart

- Pengertian

Dart adalah bahasa pemrograman yang dikembangkan oleh Google, pertama kali diperkenalkan pada tahun 2011. Dart didesain untuk membangun aplikasi yang berjalan pada berbagai platform, mulai dari web, server, hingga aplikasi mobile, terutama melalui framework Flutter. Bahasa ini memiliki sintaks yang mirip dengan JavaScript dan C++, sehingga mudah diadopsi oleh pengembang yang telah terbiasa dengan kedua bahasa tersebut. Dart memiliki dua mode eksekusi utama: Just-In-Time (JIT) untuk pengembangan cepat dan Hot Reload pada Flutter, serta Ahead-Of-Time (AOT) untuk kompilasi aplikasi menjadi kode native agar performa aplikasi menjadi lebih optimal.

Dart sangat erat kaitannya dengan pengembangan aplikasi Android melalui Flutter, framework UI yang dikembangkan Google untuk membuat aplikasi cross-platform. Dart memungkinkan pengembang untuk menulis satu kode yang dapat dijalankan di berbagai platform, termasuk Android dan iOS, tanpa harus menulis ulang logika aplikasi untuk setiap platform. Ini membuat Dart menjadi bahasa yang populer di kalangan pengembang aplikasi mobile, terutama bagi mereka yang ingin menghemat waktu pengembangan dan biaya pemeliharaan aplikasi di berbagai platform.

Dart juga dirancang untuk memberikan performa yang tinggi. Dengan fitur AOT compilation, Dart dapat mengkompilasi kode langsung menjadi native machine code, yang sangat ideal untuk aplikasi Android yang membutuhkan performa tinggi. Selain itu, Dart menyediakan dukungan untuk pemrograman asinkron, yang sangat berguna dalam menangani operasi I/O seperti jaringan dan interaksi dengan database, yang sering dijumpai dalam pengembangan aplikasi Android.

- Kelebihan

- Satu Kode untuk Banyak Platform: Dart melalui Flutter memungkinkan pengembang menulis satu kode yang dapat dijalankan di Android, iOS, web, dan desktop. Hal ini mengurangi waktu dan biaya pengembangan aplikasi untuk berbagai platform.
- Hot Reload: Fitur Hot Reload memungkinkan pengembang untuk melihat perubahan kode secara instan tanpa harus memulai ulang aplikasi, yang sangat membantu dalam proses debugging dan pengembangan antarmuka pengguna.
- Performansi Tinggi: Dengan kompilasi Ahead-of-Time (AOT), Dart mampu menghasilkan performa yang mendekati aplikasi native, menjadikannya ideal untuk aplikasi yang membutuhkan responsivitas dan kecepatan tinggi.
- Komunitas yang Kuat: Dukungan dari Google dan komunitas pengembang yang besar membuat ekosistem Dart dan Flutter berkembang pesat dengan banyaknya library dan tools yang tersedia.
- Kemudahan Asinkron: Dart mendukung pemrograman asinkron secara native, yang memungkinkan pengelolaan operasi asynchronous seperti

pengambilan data dari API atau membaca file dengan lebih efisien dan mudah.

- Kekurangan

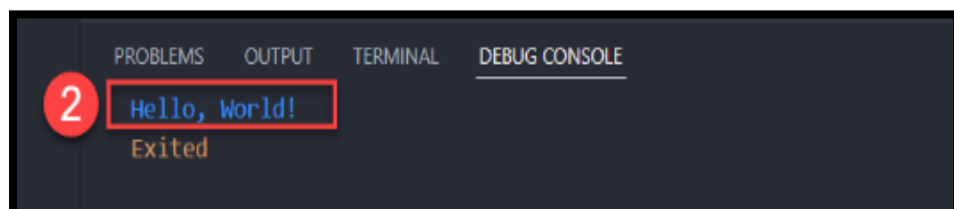
- Kurang Populer di Luar Flutter: Meskipun sangat populer untuk pengembangan aplikasi menggunakan Flutter, Dart masih kurang dikenal dan jarang digunakan di luar ekosistem Flutter, yang dapat membatasi adopsinya di luar pengembangan mobile cross-platform.
- Ukuran Aplikasi Besar: Aplikasi yang dibangun dengan Dart, terutama melalui Flutter, cenderung memiliki ukuran file yang lebih besar dibandingkan aplikasi native murni, yang bisa menjadi masalah untuk pengguna dengan perangkat dengan penyimpanan terbatas.
- Komunitas yang Lebih Kecil Dibandingkan Bahasa Lain: Meskipun komunitas Flutter berkembang pesat, komunitas Dart masih relatif lebih kecil dibandingkan bahasa pemrograman lain seperti Java atau JavaScript, sehingga dokumentasi dan support mungkin tidak sebanyak bahasa yang lebih matang.

- Contoh Penggunaan Kode



```
1 void main() {  
2   print("Hello, World!");  
3 }  
4
```

Hasil:



```
2 Hello, World!  
   Exited
```

### 1.2.2. iOS (iPhone Operating System)

- ❖ Swift

- Pengertian

Swift adalah bahasa pemrograman yang dikembangkan oleh Apple pada tahun 2014 sebagai pengganti Objective-C untuk pengembangan aplikasi di seluruh ekosistem Apple, termasuk iOS, macOS, watchOS, dan tvOS. Swift dirancang untuk menjadi lebih mudah digunakan, aman, dan cepat dibandingkan bahasa pemrograman sebelumnya. Dengan tujuan untuk memberikan pengalaman pengembangan yang lebih efisien dan modern, Swift memperkenalkan sintaks yang lebih bersih dan intuitif, sehingga memudahkan pengembang untuk menulis kode dengan lebih cepat dan lebih sedikit kesalahan.

Bahasa ini menggabungkan berbagai konsep dari paradigma pemrograman berorientasi objek dan fungsional. Dengan pendekatan ini, Swift memberikan fleksibilitas bagi pengembang untuk menggunakan berbagai teknik pemrograman sesuai dengan kebutuhan mereka. Salah satu aspek terpenting dari Swift adalah sistem keamanannya yang canggih, seperti pengelolaan memori otomatis melalui Automatic Reference Counting (ARC) dan sistem pengamanan null atau nil yang lebih ketat, sehingga meminimalkan terjadinya crash akibat referensi objek kosong. Hal ini membuat Swift menjadi salah satu bahasa yang lebih aman untuk pengembangan aplikasi dengan tingkat kompleksitas tinggi.

Swift juga menawarkan kompatibilitas dengan Objective-C, yang memungkinkan pengembang untuk menggunakan Swift di proyek-proyek yang sudah ada tanpa harus menulis ulang seluruh aplikasi. Ini berarti Swift dapat digunakan secara bertahap dalam proyek yang awalnya ditulis dalam Objective-C. Kompatibilitas ini juga menjadikan Swift sangat fleksibel untuk perusahaan yang ingin beralih ke teknologi yang lebih baru, tetapi masih memiliki komponen-komponen penting yang ditulis dalam bahasa yang lebih lama.

Swift diciptakan untuk memberikan kinerja tinggi. Ini berarti bahasa ini dioptimalkan untuk berjalan sangat cepat, baik dalam pengembangan aplikasi sederhana hingga aplikasi dengan beban berat seperti game atau aplikasi dengan grafik intensif. Menggunakan compiler LLVM yang canggih, Swift menerjemahkan kode menjadi instruksi mesin yang efisien, yang membuat aplikasi berjalan lebih cepat dan lebih responsif.

Secara keseluruhan, Swift adalah pilihan utama untuk pengembangan aplikasi pada platform Apple, menawarkan keseimbangan antara kemudahan

penggunaan, keamanan, dan kinerja tinggi. Berkat dukungan resmi dari Apple dan komunitas pengembang yang terus berkembang, Swift telah menjadi bahasa pemrograman yang mapan dan terus diperbarui secara berkala untuk mendukung fitur-fitur terbaru dari perangkat Apple.

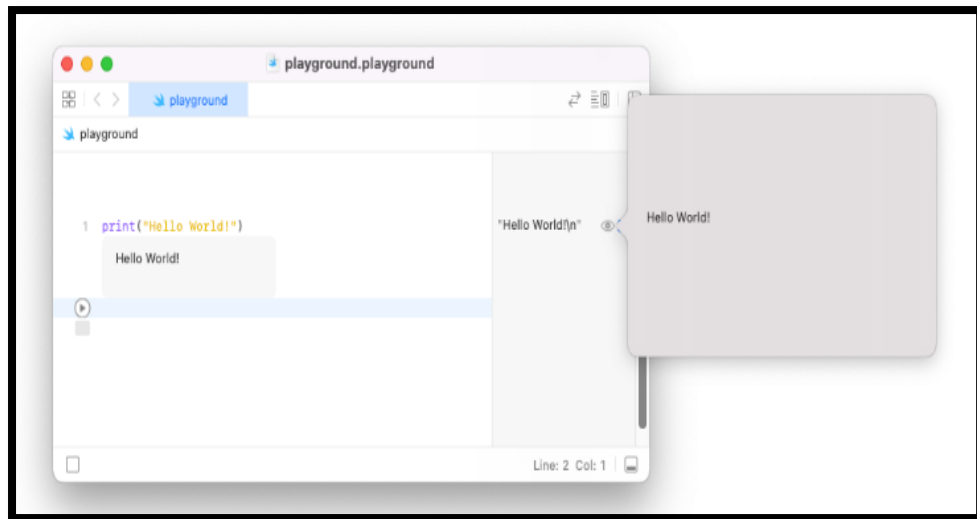
- Kelebihan

- Kinerja Tinggi: Swift dioptimalkan untuk kecepatan dan efisiensi, sehingga aplikasi berjalan lebih cepat.
- Sintaks Modern: Dengan sintaks yang lebih bersih dan mudah dipahami, Swift mengurangi kompleksitas dalam penulisan kode.
- Keamanan: Swift memiliki fitur pengelolaan memori otomatis dan menangani kesalahan, yang mengurangi kemungkinan terjadinya crash.
- Interoperabilitas: Dapat digunakan bersamaan dengan Objective-C, memungkinkan transisi yang lebih halus dari kode lama ke kode baru.
- Fitur Modern: Mendukung fitur-fitur canggih seperti generik, closures, dan protocol-oriented programming.
- Dukungan Komunitas: Memiliki komunitas yang aktif dan banyak sumber daya belajar yang tersedia, termasuk dokumentasi resmi dari Apple.

- Kekurangan

- Kurva Pembelajaran: Meskipun sintaksnya lebih sederhana, pengembang yang baru mulai mungkin masih perlu waktu untuk beradaptasi.
- Keterbatasan Platform: Hanya mendukung ekosistem Apple, sehingga tidak cocok untuk aplikasi lintas platform.
- Dukungan Pustaka yang Masih Terbatas: Beberapa pustaka dan framework yang sudah mapan dalam Objective-C mungkin belum sepenuhnya tersedia di Swift.
- Perkembangan yang Cepat: Dengan pembaruan yang sering, pengembang mungkin perlu terus-menerus memperbarui pengetahuan mereka tentang fitur baru dan perubahan dalam bahasa.

- Contoh Penggunaan Kode



## ❖ Objective-C

- Pengertian

Objective-C adalah bahasa pemrograman berorientasi objek yang dikembangkan pada awal 1980-an oleh Brad Cox dan Tom Love. Bahasa ini merupakan perluasan dari bahasa C dengan tambahan kemampuan pemrograman berorientasi objek (OOP) melalui penggunaan konsep-konsep seperti kelas, objek, dan pesan. Objective-C menjadi bahasa utama yang digunakan oleh Apple untuk pengembangan aplikasi di ekosistem mereka, seperti macOS dan iOS, sebelum akhirnya digantikan oleh Swift pada tahun 2014. Meskipun Swift saat ini lebih dominan, Objective-C tetap digunakan dalam banyak proyek warisan dan masih kompatibel dengan kode Swift.

Sebagai bahasa pemrograman yang dikompilasi, Objective-C memanfaatkan runtime dinamis, yang berarti banyak keputusan pemrograman dilakukan saat aplikasi sedang berjalan, bukan saat dikompilasi. Ini memberikan fleksibilitas yang tinggi dalam manipulasi objek dan alokasi memori. Selain itu, Objective-C menggunakan pesan (messaging) untuk berkomunikasi antar objek, yang lebih mirip dengan metode Smalltalk dibandingkan dengan bahasa-bahasa OOP lainnya seperti C++ yang lebih mengandalkan pemanggilan metode langsung.

Bahasa Objective-C juga mendukung penggunaan kerangka kerja atau framework besar seperti Cocoa dan Cocoa Touch, yang sangat esensial untuk pengembangan aplikasi di ekosistem Apple. Cocoa adalah kerangka kerja utama



untuk pengembangan aplikasi di macOS, sedangkan Cocoa Touch adalah untuk iOS. Kedua kerangka kerja ini menyediakan antarmuka pemrograman aplikasi (API) yang kaya, memungkinkan pengembang membangun aplikasi yang kompleks dengan lebih mudah.

Objective-C tetap dihargai karena kedewasaannya, stabilitasnya, dan keberadaannya di proyek-proyek besar dalam industri pengembangan perangkat lunak Apple. Banyak aplikasi dan pustaka yang sudah dibangun selama bertahun-tahun menggunakan Objective-C, sehingga bahasa ini masih memiliki relevansi, terutama dalam pemeliharaan dan pengembangan fitur baru untuk aplikasi yang sudah ada.

- Kelebihan

- Kompatibilitas dengan C: Karena Objective-C merupakan perluasan dari C, pengembang dapat menggunakan kode C di dalam program Objective-C, memungkinkan pengembang untuk memanfaatkan pustaka dan alat C yang ada.
- Runtime Dinamis: Memberikan fleksibilitas dalam pemrograman berorientasi objek, memungkinkan metode ditambahkan ke kelas selama waktu eksekusi.
- Kompatibilitas dengan Swift: Objective-C dapat digunakan bersama Swift dalam proyek yang sama, yang memungkinkan transisi yang lancar ke bahasa baru tanpa harus menulis ulang seluruh kode.
- Kerangka Kerja Kaya: Objective-C menawarkan akses ke Cocoa dan Cocoa Touch, kerangka kerja yang kaya yang menyediakan antarmuka untuk hampir semua kebutuhan pengembangan aplikasi di ekosistem Apple.
- Stabil dan Teruji Waktu: Objective-C telah ada selama beberapa dekade dan telah digunakan untuk membangun banyak aplikasi stabil yang berjalan di jutaan perangkat Apple.

- Kekurangan

- Sintaks yang Kompleks: Sintaks Objective-C dianggap lebih rumit dan sulit dipahami dibandingkan bahasa pemrograman modern seperti Swift. Banyak tanda kurung kurawal dan pesan yang terlihat tidak intuitif untuk pemula.

- Keamanan Lebih Rendah: Objective-C tidak memiliki fitur keamanan bawaan seperti Swift, seperti manajemen null safety yang lebih ketat, yang bisa menyebabkan lebih banyak kesalahan runtime.
- Kurva Pembelajaran Curam: Untuk pengembang baru, Objective-C bisa jadi sulit dipelajari karena kombinasi fitur C dan pendekatan uniknya terhadap OOP.
- Popularitas Menurun: Dengan diperkenalkannya Swift yang lebih modern dan mudah digunakan, popularitas Objective-C telah menurun drastis, meskipun masih penting untuk memelihara aplikasi lama.

- Contoh Penggunaan Kode

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

@interface SampleClass:NSObject
- (void)sampleMethod;
@end

@implementation SampleClass

- (void)sampleMethod {
    NSLog(@"Hello, World! \n");
}

@end

int main() {
    /* my first program in Objective-C */
    SampleClass *sampleClass = [[SampleClass alloc] init];
    [sampleClass sampleMethod];
    return 0;
}
```

Hasil:

```
demo[65832] Hello, World!
```

### 1.3. Desktop

#### 1.3.1. .Net

- Pengertian

.NET adalah platform pengembangan perangkat lunak yang dibuat oleh Microsoft, yang memungkinkan pengembang untuk membangun berbagai jenis aplikasi, termasuk aplikasi desktop, web, mobile, dan layanan cloud. .NET mendukung banyak bahasa pemrograman seperti C#, Visual Basic, dan F#. Untuk pengembangan aplikasi desktop, .NET menyediakan kerangka kerja yang kuat melalui Windows Forms (WinForms), Windows Presentation Foundation (WPF), dan Universal Windows Platform (UWP).

.NET Framework awalnya dirancang untuk Windows, namun dengan diperkenalkannya .NET Core (sekarang dikenal sebagai .NET 5 dan seterusnya), pengembang dapat membuat aplikasi yang berjalan lintas platform di Windows, Linux, dan macOS. Ini memberi fleksibilitas yang lebih besar bagi pengembang yang ingin menciptakan aplikasi yang dapat berjalan di berbagai sistem operasi. Pada platform desktop, .NET umumnya digunakan untuk membangun aplikasi dengan antarmuka pengguna grafis (GUI), seperti alat bisnis, perangkat lunak manajemen, hingga aplikasi hiburan.

Arsitektur .NET mendukung Garbage Collection (pengelolaan memori otomatis), yang membantu meningkatkan efisiensi dan stabilitas aplikasi. Selain itu, .NET menawarkan banyak pustaka dan API yang mempermudah pengembang dalam menangani berbagai kebutuhan umum, seperti akses data, jaringan, dan grafik. Windows Forms dan WPF adalah dua komponen utama yang digunakan dalam pengembangan aplikasi desktop menggunakan .NET. Windows Forms memberikan kemudahan dalam membuat antarmuka pengguna sederhana, sedangkan WPF mendukung aplikasi dengan desain antarmuka yang lebih kompleks, termasuk animasi dan elemen 3D.

Dengan evolusi .NET ke .NET 5, pengembangan aplikasi desktop menjadi lebih fleksibel dengan dukungan terhadap perangkat modern, peningkatan performa, dan integrasi yang lebih baik dengan layanan cloud serta DevOps. Dengan .NET, pengembang dapat membangun aplikasi yang mudah dikelola dan diperbarui dengan arsitektur modular serta pendekatan modern seperti dependency injection dan pemrograman asinkron.

- Kelebihan

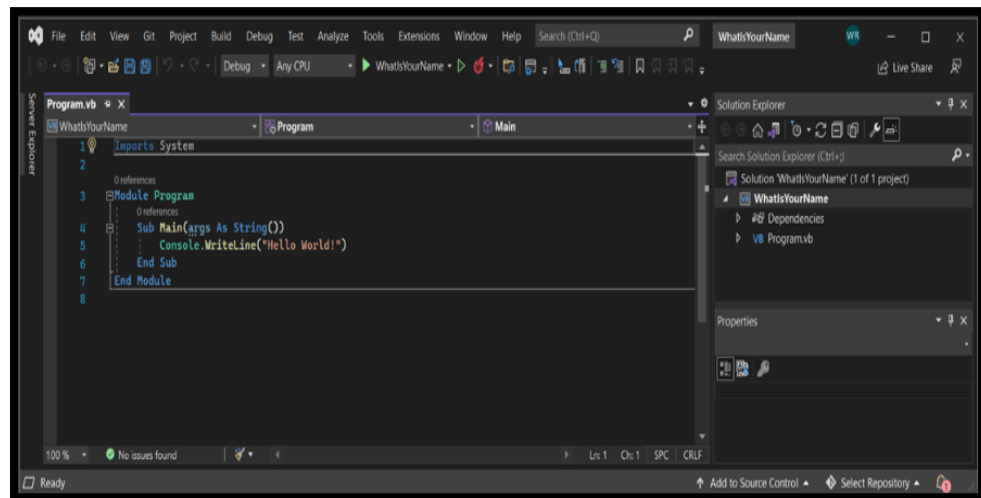
- Lintas Platform: Dengan .NET Core dan penerusnya, aplikasi desktop dapat dikembangkan untuk berjalan tidak hanya di Windows, tetapi juga di Linux dan macOS.
- Dukungan Bahasa Beragam: .NET mendukung banyak bahasa pemrograman seperti C#, F#, dan Visual Basic, memberi fleksibilitas dalam pemilihan bahasa.
- Komunitas Besar dan Dukungan Komersial: Karena dikembangkan oleh Microsoft dan digunakan secara luas, .NET memiliki komunitas besar yang aktif dan banyak sumber daya belajar serta dukungan komersial.
- Kinerja Tinggi: Dengan dukungan garbage collection, optimisasi memori, dan kompilasi just-in-time (JIT), aplikasi .NET dapat berjalan dengan sangat cepat dan efisien.
- Ekosistem Luas: .NET menyediakan berbagai pustaka dan API yang memungkinkan integrasi mudah dengan layanan cloud, database, dan alat lain yang dibutuhkan dalam pengembangan aplikasi desktop modern.
- Pemeliharaan Memori Otomatis: Fitur garbage collection otomatis menangani manajemen memori, sehingga pengembang dapat fokus pada logika bisnis aplikasi tanpa harus khawatir tentang kebocoran memori.
- Tooling Terbaik: Visual Studio, yang merupakan IDE andalan untuk .NET, menawarkan tooling yang sangat baik dengan fitur debugging canggih, pengelolaan dependensi, dan pengembangan berbasis DevOps.

- Kekurangan

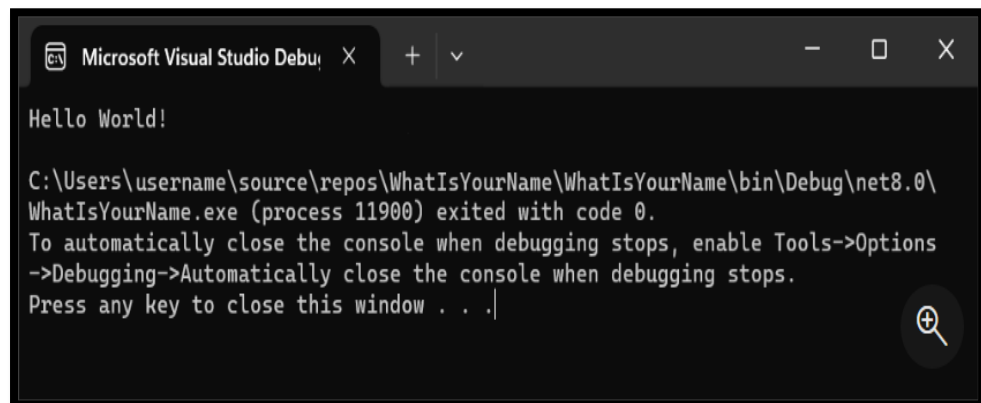
- Ukuran Aplikasi yang Besar: Aplikasi .NET desktop cenderung memiliki ukuran yang lebih besar dibandingkan dengan aplikasi yang dikembangkan menggunakan teknologi lain, terutama jika dibundel dengan runtime.
- Kurva Pembelajaran: Meskipun dokumentasinya sangat lengkap, pengembang baru mungkin membutuhkan waktu untuk menguasai seluruh ekosistem .NET, terutama dengan berbagai framework dan API yang tersedia.
- Tidak Cocok untuk Aplikasi Ringan: Karena fitur dan pustakanya yang kaya, .NET bisa berlebihan untuk aplikasi yang sangat sederhana atau memiliki sumber daya terbatas.

- Keterbatasan pada Platform Lain: Meskipun .NET Core mendukung lintas platform, beberapa fitur mungkin tidak berfungsi secara optimal atau memerlukan pengaturan tambahan di luar lingkungan Windows.
- WPF dan WinForms Terbatas pada Windows: Meskipun .NET Core mendukung lintas platform, WPF dan WinForms tetap terbatas pada Windows, yang berarti pengembangan aplikasi desktop lintas platform memerlukan pendekatan yang berbeda.

- Contoh Penggunaan Kode



Hasil:



### 1.3.2. C#

- Pengertian

C# adalah bahasa pemrograman yang dikembangkan oleh Microsoft pada tahun 2000 sebagai bagian dari inisiatif .NET Framework. Bahasa ini dirancang oleh tim yang dipimpin oleh Anders Hejlsberg dengan tujuan menciptakan bahasa yang sederhana, modern, berorientasi objek, dan berdaya guna tinggi. C# memiliki sintaks yang terstruktur dan mirip dengan bahasa C dan C++, namun

lebih mudah dipelajari karena menghilangkan kompleksitas rendah-level dari bahasa pendahulunya. C# sangat populer untuk pengembangan aplikasi desktop, terutama pada platform Windows, dengan dukungan penuh dari teknologi .NET seperti Windows Forms dan Windows Presentation Foundation (WPF). Dengan menggunakan C#, pengembang dapat membangun aplikasi desktop yang kaya fitur, efisien, dan memiliki antarmuka pengguna (UI) yang menarik.

C# berjalan di atas Common Language Runtime (CLR), yang merupakan inti dari platform .NET. CLR bertanggung jawab untuk melakukan manajemen memori otomatis, pengumpulan sampah (garbage collection), serta menyediakan fitur keamanan dan performa yang tinggi. Hal ini membuat C# menjadi bahasa yang aman dari kesalahan umum seperti kebocoran memori atau kesalahan alokasi memori yang sering terjadi pada bahasa seperti C dan C++. Sebagai bahasa yang berorientasi objek, C# mendukung konsep-konsep penting seperti enkapsulasi, pewarisan, dan polimorfisme. Ini memungkinkan pengembang untuk membuat aplikasi yang modular, mudah diperluas, dan lebih mudah dipelihara. Selain itu, C# mendukung pemrograman asinkron dan multi-threading, yang memungkinkan pengembangan aplikasi desktop yang responsif, terutama saat menangani tugas-tugas yang berat, seperti pengolahan data atau rendering UI.

Selain pengembangan desktop pada Windows, C# juga telah berkembang menjadi bahasa lintas platform dengan adanya .NET Core (sekarang .NET 5 dan versi di atasnya). Dengan ini, C# bisa digunakan untuk membangun aplikasi desktop yang berjalan di sistem operasi lain seperti macOS dan Linux, meskipun dukungan terbaik masih diberikan pada pengembangan di Windows. C# juga memiliki dukungan yang kuat dari Visual Studio, Integrated Development Environment (IDE) yang sangat terkenal dan banyak digunakan oleh pengembang untuk menulis, menguji, dan melakukan debugging aplikasi desktop C#. IDE ini menawarkan berbagai alat dan fitur canggih yang memudahkan proses pengembangan, termasuk IntelliSense (fitur auto-completion), debugger yang kuat, serta dukungan untuk pengembangan UI berbasis drag-and-drop menggunakan Windows Forms atau WPF.

Dalam pengembangan aplikasi desktop, C# digunakan untuk membangun aplikasi client-side yang dapat berjalan secara lokal di komputer pengguna. Aplikasi desktop C# sering digunakan dalam berbagai sektor, seperti bisnis, keuangan, dan pendidikan, di mana diperlukan aplikasi yang stabil, cepat, dan

dapat diandalkan untuk menangani data, berinteraksi dengan pengguna, dan mengelola sumber daya lokal. Secara keseluruhan, C# adalah bahasa yang sangat kuat untuk pengembangan aplikasi desktop di lingkungan Windows dan telah menjadi pilihan utama untuk pengembangan perangkat lunak bisnis, aplikasi produktivitas, dan alat-alat desktop lainnya. Dengan dukungan penuh dari ekosistem Microsoft, C# menawarkan berbagai fitur modern dan infrastruktur yang kaya, sehingga sangat cocok untuk membangun aplikasi desktop yang skalabel dan performa tinggi.

- Kelebihan

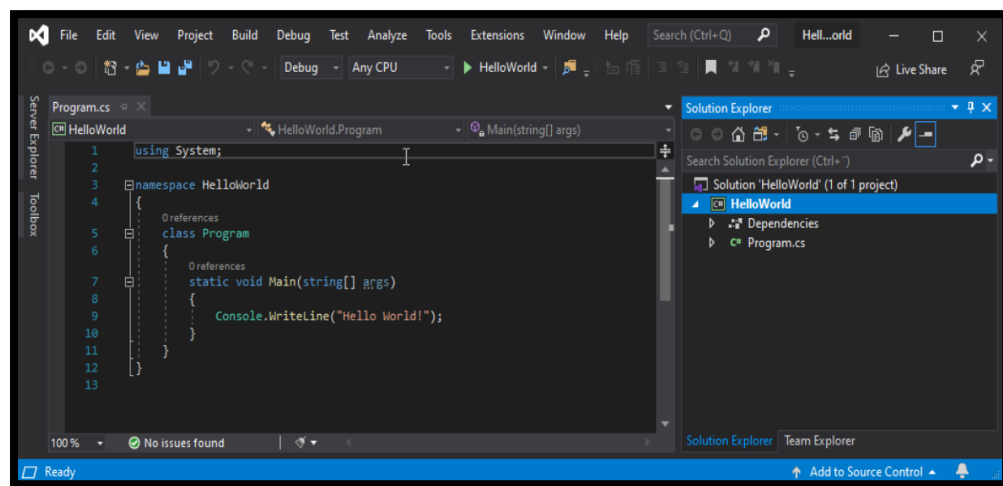
- Integrasi Kuat dengan Windows: C# memiliki integrasi yang sangat baik dengan sistem operasi Windows, memungkinkan pengembang memanfaatkan fitur-fitur Windows secara penuh.
- Berkembang di .NET: C# adalah bahasa utama di platform .NET, yang memberikan banyak dukungan untuk pengembangan aplikasi desktop dengan performa tinggi.
- Manajemen Memori Otomatis: Dengan garbage collection dan manajemen memori otomatis, pengembang tidak perlu khawatir tentang kebocoran memori.
- Framework Rich UI: C# mendukung berbagai framework untuk pengembangan antarmuka pengguna (UI) yang kaya, seperti Windows Forms dan WPF.
- Multi-threading: C# mendukung pengembangan aplikasi multi-threading yang dapat meningkatkan responsivitas dan kinerja aplikasi desktop.
- Penggunaan Umum di Industri: Sebagai bahasa yang populer dan didukung oleh Microsoft, C# sering digunakan dalam berbagai proyek skala besar dan enterprise.
- Lintas Platform: Melalui .NET Core (sekarang .NET 5+), C# mendukung pengembangan aplikasi desktop di berbagai platform, termasuk Windows, macOS, dan Linux.

- Kekurangan

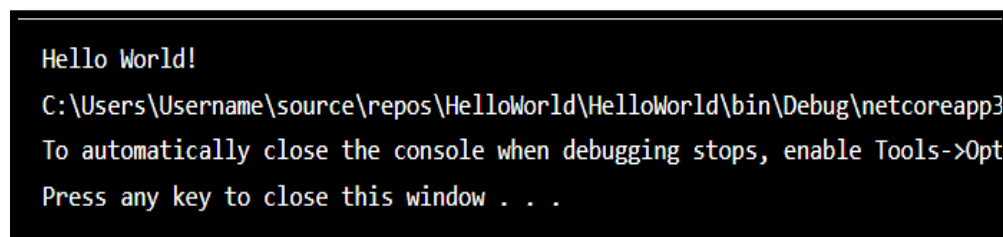
- Tergantung pada Ekosistem Microsoft: Meskipun C# dapat digunakan di platform lain melalui .NET Core, penggunaan utamanya tetap fokus pada ekosistem Microsoft dan Windows.

- Ukuran Aplikasi: Aplikasi desktop yang dikembangkan menggunakan C# dan .NET cenderung memiliki ukuran file yang lebih besar, terutama ketika harus menyertakan runtime .NET.
- Kompleksitas untuk Aplikasi Kecil: Untuk aplikasi desktop kecil, framework seperti WPF dan Windows Forms mungkin terasa terlalu kompleks dibandingkan dengan alternatif yang lebih ringan.
- Performa di Luar Windows: Meskipun lintas platform, performa aplikasi C# mungkin tidak seoptimal di luar lingkungan Windows, terutama jika membutuhkan integrasi mendalam dengan sistem operasi lain.

- Contoh Penggunaan Kode



Hasil:



### 1.3.3. Delphi

- Pengertian

Delphi adalah lingkungan pengembangan terpadu (IDE) dan bahasa pemrograman yang digunakan untuk membuat aplikasi desktop, mobile, web, dan konsol, terutama pada sistem operasi Windows. Dikembangkan oleh Borland pada tahun 1995 dan saat ini dikelola oleh Embarcadero Technologies, Delphi menggunakan bahasa Object Pascal, turunan dari Pascal yang mendukung pemrograman berorientasi objek. Delphi terkenal dengan kemampuan RAD



(Rapid Application Development), yang memungkinkan pengembang membangun aplikasi dengan cepat melalui antarmuka visual dan komponen yang dapat digunakan kembali.

Delphi memanfaatkan konsep komponen visual yang memungkinkan pengembang untuk menyusun antarmuka pengguna (UI) secara drag-and-drop, membuat pengembangan aplikasi desktop lebih cepat dan efisien. Dengan dukungan VCL (Visual Component Library) untuk aplikasi Windows dan FireMonkey (FMX) untuk aplikasi lintas platform, Delphi memberikan fleksibilitas dalam membangun aplikasi untuk berbagai sistem operasi, termasuk Windows, macOS, iOS, dan Android.

Sebagai IDE yang kuat, Delphi mendukung berbagai fitur modern seperti debugging, profiling, dan refactoring, yang memudahkan pengembangan dan pemeliharaan aplikasi. Delphi juga memungkinkan pemrograman berorientasi objek yang mendukung konsep-konsep seperti enkapsulasi, pewarisan, dan polimorfisme, sehingga memungkinkan pengembangan aplikasi yang modular dan mudah dikembangkan lebih lanjut.

Delphi sering digunakan untuk membangun aplikasi desktop bisnis yang kaya fitur, stabil, dan dapat menangani data dalam skala besar, serta aplikasi yang memerlukan integrasi dengan berbagai layanan sistem operasi. Kemampuan Delphi untuk menghasilkan kode mesin asli (native code) membuatnya efisien dan cepat dalam performa, terutama dalam menjalankan aplikasi desktop yang membutuhkan performa tinggi.

- Kelebihan

- RAD (Rapid Application Development): Delphi menawarkan alat pengembangan visual yang memungkinkan pengembang membangun aplikasi desktop dengan cepat melalui antarmuka drag-and-drop dan komponen yang siap digunakan.
- Native Compilation: Delphi menghasilkan kode mesin asli, yang memastikan aplikasi yang dibangun menggunakan Delphi memiliki performa tinggi dan efisien, terutama pada sistem operasi Windows.
- Cross-Platform Development: Dengan dukungan FireMonkey (FMX), pengembang dapat membuat aplikasi lintas platform yang dapat dijalankan pada Windows, macOS, iOS, dan Android.

- VCL (Visual Component Library): Perpustakaan komponen visual yang kaya untuk pengembangan aplikasi desktop Windows, memudahkan pengembang dalam membangun antarmuka pengguna yang profesional.
  - Stabil dan Terpercaya: Delphi memiliki sejarah panjang dalam pengembangan aplikasi desktop bisnis dan telah terbukti stabil serta mampu menangani aplikasi yang memerlukan keandalan tinggi.
  - Object Pascal: Bahasa pemrograman yang digunakan oleh Delphi, Object Pascal, memiliki sintaks yang mudah dipelajari, terutama bagi pengembang yang familiar dengan bahasa Pascal, tetapi tetap mendukung pemrograman berorientasi objek.
- Kekurangan
    - Kurva Pembelajaran untuk Pemula: Meskipun Delphi adalah alat yang kuat, pengembang pemula mungkin membutuhkan waktu untuk mempelajari antarmuka IDE dan pendekatan RAD yang digunakan.
    - Dukungan Lintas Platform Terbatas: Meskipun Delphi mendukung pengembangan lintas platform, dukungan terbaiknya masih untuk Windows. Pengembangan di platform lain seperti macOS dan mobile mungkin tidak sekuat pada Windows.
    - Biaya Lisensi: Delphi bukan perangkat lunak open-source dan membutuhkan biaya lisensi yang bisa mahal, terutama untuk versi profesional atau edisi yang lebih tinggi.
    - Komunitas Pengguna yang Lebih Kecil: Dibandingkan dengan bahasa pemrograman populer lainnya seperti C# atau Java, Delphi memiliki komunitas pengguna yang lebih kecil, sehingga mungkin lebih sulit menemukan sumber daya pembelajaran atau dukungan dari komunitas.

- Contoh Penggunaan Kode

```
program Project2;

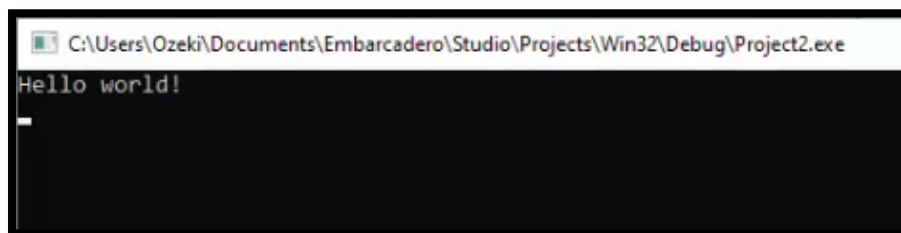
{$APPTYPE CONSOLE}

{$R *.res}

uses
  System.SysUtils;

begin
  try
    WriteLn('Hello world!');
    ReadLn;
  except
    on E: Exception do
      WriteLn(E.ClassName, ': ', E.Message);
    end;
  end.
end.
```

Hasil:



## **BAB II**

### **INSTALASI DAN UJI COBA SDK FLUTTER**

#### **2.1. Teori Dasar**

##### **2.1.1. Android Studio**

Android Studio adalah Integrated Development Environment (IDE) yang dikembangkan oleh Google untuk pengembangan aplikasi Android. IDE ini didasarkan pada IntelliJ IDEA dari JetBrains, yang menyediakan berbagai fitur canggih untuk memudahkan pengembang. Android Studio dirilis pertama kali pada tahun 2013 untuk menggantikan Eclipse sebagai IDE utama untuk pengembangan aplikasi Android. Salah satu keunggulan utama Android Studio adalah integrasinya dengan berbagai alat pengembangan Android dan kemampuannya untuk mendukung pengembangan aplikasi yang kompleks dengan lebih mudah dan efisien.

Android Studio dilengkapi dengan berbagai komponen utama, seperti Gradle Build System, Android Emulator, dan Layout Editor. Gradle memungkinkan pengembang untuk mengelola proses build aplikasi, mengatur dependensi, dan membuat konfigurasi build yang fleksibel. Android Emulator memungkinkan pengujian aplikasi di berbagai perangkat virtual, sementara Layout Editor memudahkan perancangan antarmuka pengguna secara visual. Fitur-fitur ini menjadikan Android Studio sebagai alat yang sangat bertenaga dan efisien bagi pengembang aplikasi Android.

Selain itu, Android Studio mendukung berbagai bahasa pemrograman, terutama Java, Kotlin, dan C++. Kotlin, yang diperkenalkan sebagai bahasa resmi untuk Android pada tahun 2017, menawarkan sintaksis yang lebih ringkas dan modern dibandingkan Java, serta memungkinkan interoperabilitas dengan Java. Bahasa pemrograman ini membantu pengembang membuat aplikasi dengan lebih sedikit bug dan lebih cepat beradaptasi dengan tren pengembangan aplikasi modern. Bagi aplikasi yang memerlukan performa tinggi, C++ juga dapat digunakan untuk interaksi langsung dengan hardware.

Proses pengembangan aplikasi Android di Android Studio meliputi beberapa tahapan penting, seperti desain tata letak, pengembangan logika aplikasi, pengujian, debugging, dan deployment. Dengan fitur-fitur seperti Instant Run, pengembang dapat melakukan perubahan kode secara langsung pada aplikasi yang sedang

berjalan tanpa perlu memulai ulang seluruh aplikasi, yang mempercepat proses pengembangan. Selain itu, alat debugging seperti Logcat dan Profiler Tools membantu pengembang menganalisis performa dan mengidentifikasi masalah dengan lebih efisien.

Secara keseluruhan, Android Studio merupakan pilihan utama bagi pengembang aplikasi Android karena menyediakan lingkungan yang komprehensif dan terintegrasi dengan baik. Dengan dukungan untuk berbagai perangkat Android dan kemampuan untuk menangani build yang kompleks, Android Studio memungkinkan pengembang untuk fokus pada pembuatan aplikasi berkualitas tinggi sambil memanfaatkan teknologi terbaru dari Google.

### **2.1.2. SDK Flutter**

Flutter adalah framework open-source yang dikembangkan oleh Google untuk membangun aplikasi lintas platform, memungkinkan pengembang membuat aplikasi untuk perangkat mobile, desktop, dan web dengan satu basis kode. Di pusat ekosistem Flutter terdapat Flutter SDK (Software Development Kit), yang menyediakan alat dan pustaka yang diperlukan untuk pengembangan. Dengan menggunakan bahasa pemrograman Dart, Flutter memudahkan pengembang dalam menciptakan antarmuka pengguna (UI) yang responsif dan estetis.

Salah satu fitur unggulan Flutter adalah hot reload, yang memungkinkan pengembang melihat perubahan kode secara langsung tanpa harus memulai ulang aplikasi. Ini sangat mempercepat proses pengembangan dan iterasi desain. Flutter berbasis pada konsep widget, di mana hampir setiap elemen UI merupakan widget yang dapat disesuaikan dan digabungkan. Dengan beragam widget yang tersedia, pengembang dapat dengan mudah menciptakan antarmuka yang kompleks dan interaktif.

Arsitektur Flutter mengedepankan pemisahan antara UI dan logika bisnis, mirip dengan arsitektur MVC (Model-View-Controller). Flutter menerapkan pemrograman reaktif, di mana UI diperbarui berdasarkan perubahan data. Hal ini memungkinkan pengembang untuk lebih mudah mengelola state aplikasi dan memastikan bahwa antarmuka pengguna selalu mencerminkan data terkini. Dengan kompilasi kode menjadi kode mesin native, Flutter juga menjamin kinerja tinggi di berbagai platform.

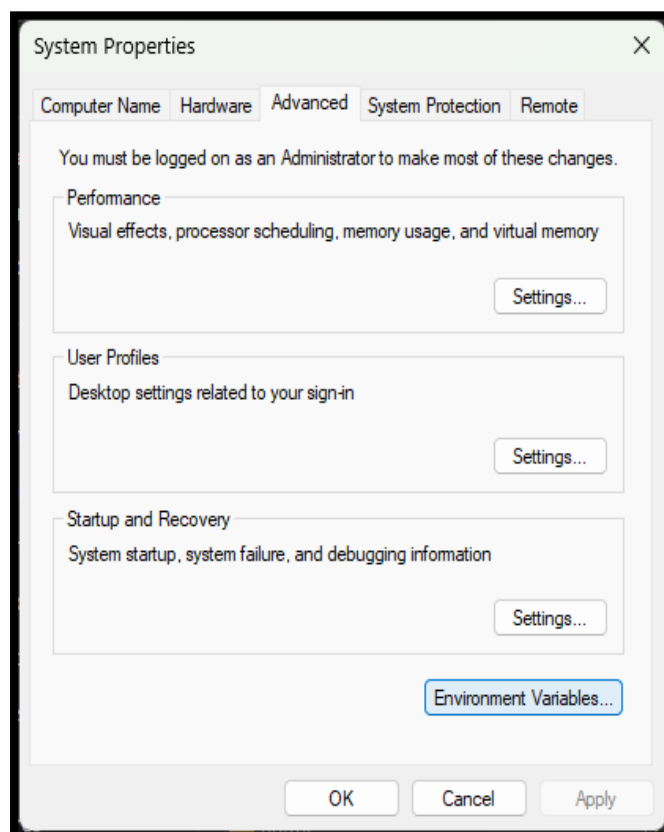
Kemampuan Flutter untuk mendukung pengembangan lintas platform adalah salah satu keunggulannya. Dengan satu basis kode, pengembang dapat menjalankan aplikasi di berbagai sistem operasi, seperti Android dan iOS, tanpa perlu menulis kode terpisah untuk masing-masing platform. Flutter menyediakan widget yang disesuaikan untuk platform tertentu, sehingga pengalaman pengguna tetap konsisten dan sesuai dengan desain masing-masing sistem operasi.

Ekosistem Flutter terus berkembang, didukung oleh banyaknya paket dan pustaka yang tersedia di repositori pub.dev. Pengembang dapat dengan mudah mengintegrasikan berbagai fungsionalitas tambahan ke dalam aplikasi mereka melalui paket-paket ini, seperti integrasi API dan manajemen state. Komunitas yang aktif juga menyediakan dokumentasi dan tutorial yang membantu pengembang dalam memanfaatkan Flutter secara efektif, menjadikannya pilihan menarik untuk pengembangan aplikasi modern.

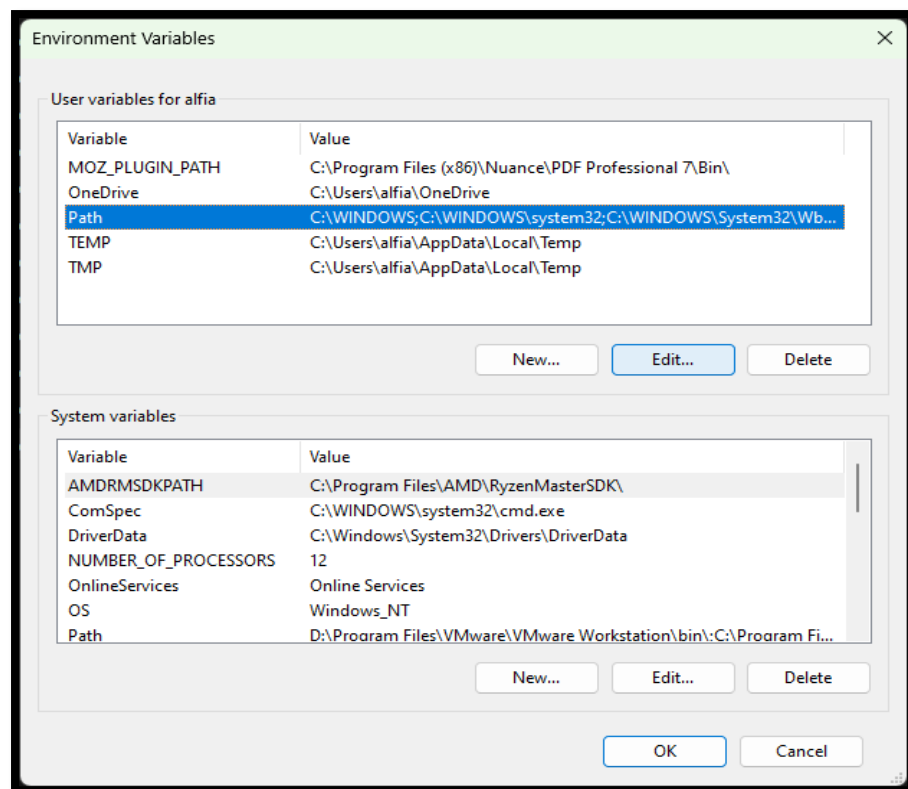
## 2.2. Praktikum

### 2.2.1. Instalasi SDK Flutter

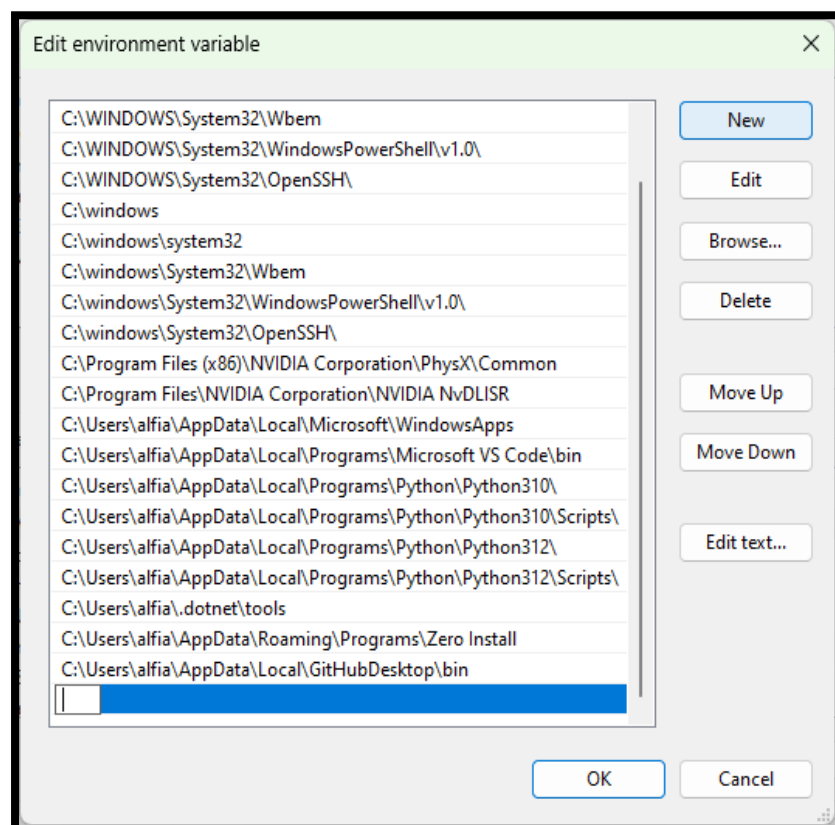
- Masuk menuju “Edit the system environment variables” dan klik bagian “Environment Variables”.

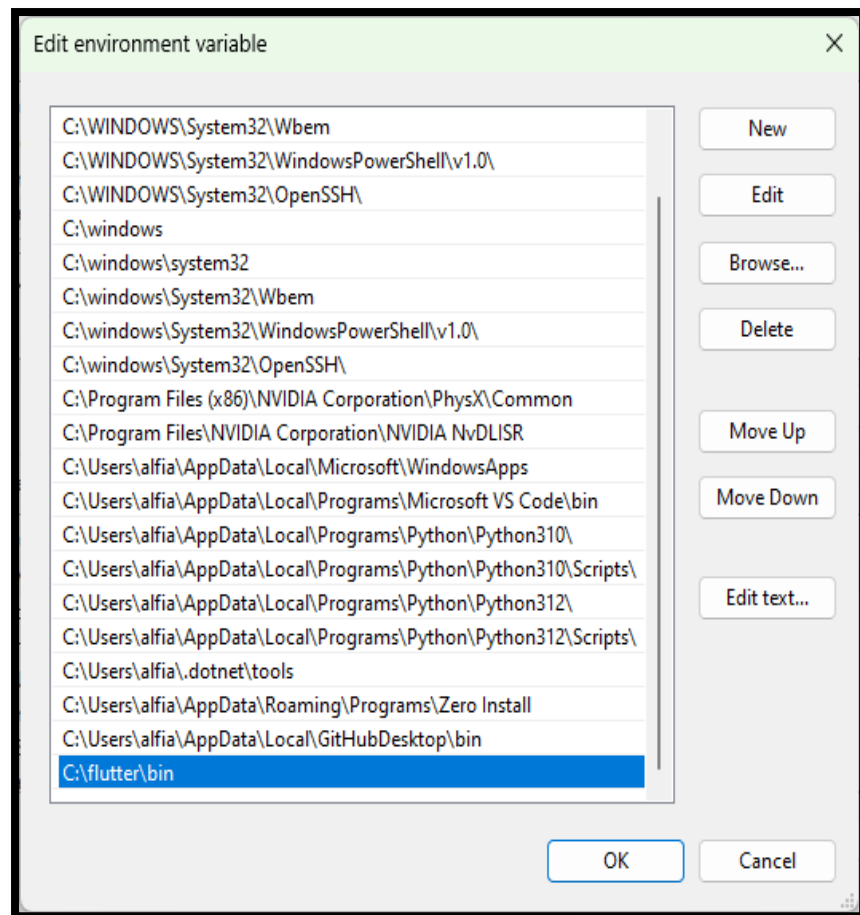


- Memilih “Path” pada bagian “User variables” dan klik bagian “Edit”.



- Klik bagian “New” and masukkan alamat directory dari folder “bin” yang berasal dari folder “flutter”, terakhir simpan dengan klik “OK”.





- Menjalankan “Command Prompt” sebagai “Administrator”, ketik “flutter”, dan tunggu hingga proses selesai.

```

Administrator: Command Prompt - flutter
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>flutter
Manage your Flutter app development.

Common commands:

  flutter create <output directory>
    Create a new Flutter project in the specified directory.

  flutter run [options]
    Run your Flutter application on an attached device or in an emulator.

Usage: flutter <command> [arguments]

Global options:
-h, --help           Print this usage information.
-v, --verbose        Noisy logging, including all shell commands executed.
                    If used with "--help", shows hidden options. If used with "flutter doctor", shows additional
                    diagnostic information. (Use "-vv" to force verbose logging in those cases.)
-d, --device-id      Target device id or name (prefixes allowed).
--version            Reports the version of this tool.
--enable-analytics   Enable telemetry reporting each time a flutter or dart command runs.
--disable-analytics  Disable telemetry reporting each time a flutter or dart command runs, until it is
                    re-enabled.
--suppress-analytics Suppress analytics reporting for the current CLI invocation.

Available commands:

Flutter SDK
bash-completion  Output command line shell completion setup scripts.
channel          List or switch Flutter channels.
config           Configure Flutter settings.
doctor           Show information about the installed tooling.
downgrade        Downgrade Flutter to the last active version for the current channel.
precache         Populate the Flutter tool's cache of binary artifacts.
upgrade          Upgrade your copy of Flutter.
  
```



```
Administrator: Command Prompt - flutter

Welcome to Flutter! - https://flutter.dev

The Flutter tool uses Google Analytics to anonymously report feature usage
statistics and basic crash reports. This data is used to help improve
Flutter tools over time.

Flutter tool analytics are not sent on the very first run. To disable
reporting, type 'flutter config --no-analytics'. To display the current
setting, type 'flutter config'. If you opt out of analytics, an opt-out
event will be sent, and then no further information will be sent by the
Flutter tool.

By downloading the Flutter SDK, you agree to the Google Terms of Service.
The Google Privacy Policy describes how data is handled in this service.

Moreover, Flutter includes the Dart SDK, which may send usage metrics and
crash reports to Google.

Read about data we send with crash reports:
https://flutter.dev/to/crash-reporting

See Google's privacy policy:
https://policies.google.com/privacy

To disable animations in this tool, use
'flutter config --no-cli-animations'.

The Flutter CLI developer tool uses Google Analytics to report usage and diagnostic
data along with package dependencies, and crash reporting to send basic crash
reports. This data is used to help improve the Dart platform, Flutter framework,
and related tools.

Telemetry is not sent on the very first run. To disable reporting of telemetry,
run this terminal command:
```

- Setelah proses selesai, kemudian ketik “flutter doctor” dan tunggu hingga proses selesai.

```
Administrator: Command Prompt - flutter - flutter doctor

flutter --disable-analytics

If you opt out of telemetry, an opt-out event will be sent, and then no further
information will be sent. This data is collected in accordance with the Google
Privacy Policy (https://policies.google.com/privacy).

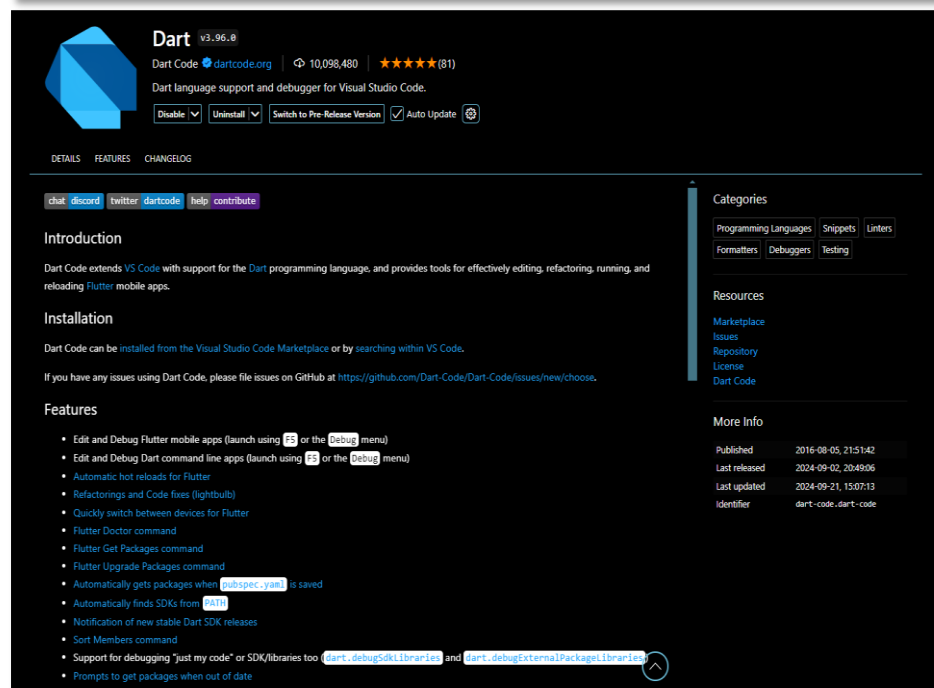
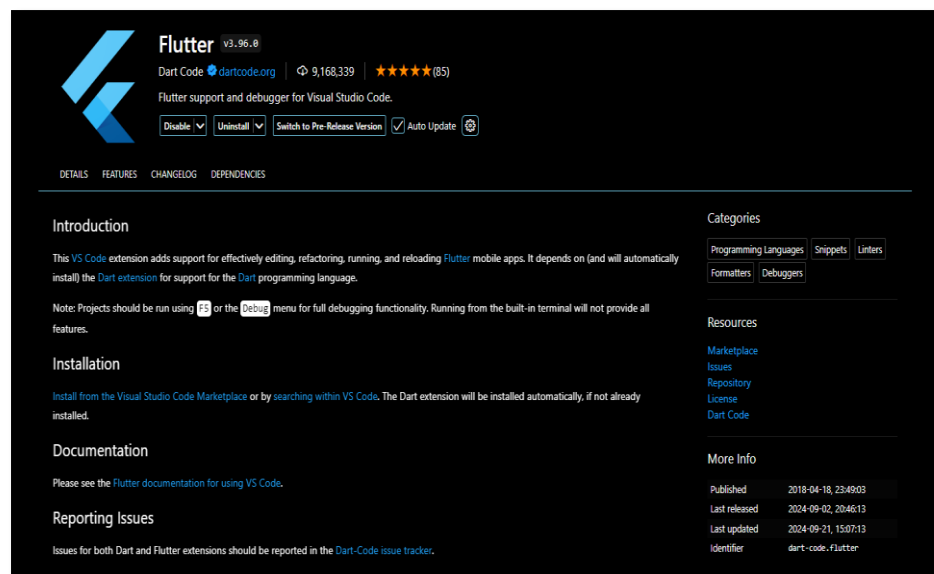
You have received two consent messages because the flutter tool is migrating to a new analytics system. Disabling
analytics collection will disable both the legacy and new analytics collection systems. You can disable analytics
reporting by running 'flutter --disable-analytics'

C:\Windows\System32>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 id-ID)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
    X cmdline-tools component is missing
      Run 'path/to/sdkmanager --install "cmdline-tools;latest"'
      See https://developer.android.com/studio/command-line for more details.
    X Android license status unknown.
      Run 'flutter doctor --android-licenses' to accept the SDK licenses.
      See https://flutter.dev/to/windows-android-setup for more details.
[✓] Chrome - develop for the web
[!] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.7.6)
    X Visual Studio is missing necessary components. Please re-run the Visual Studio installer for the "Desktop development with C++" workload,
      and include these components:
        MSVC v142 - VS 2019 C++ x64/x86 build tools
        - If there are multiple build tool versions available, install the latest
        C++ Make tools for Windows
        Windows 10 SDK
[✓] Android Studio (version 2024.1)
[✓] VS Code (version 1.90.1)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 2 categories.

C:\Windows\System32>
```

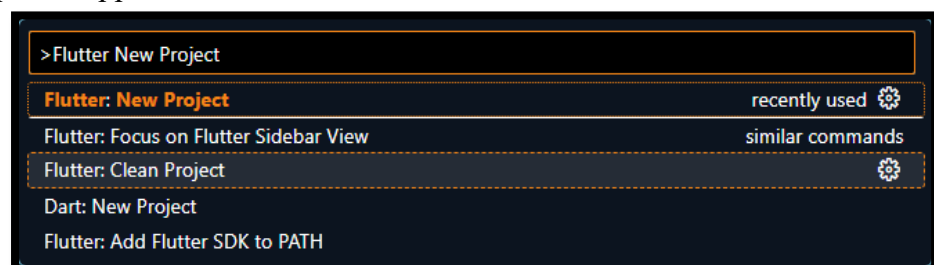
- Menjalankan “Visual Studio Code” dan install extensions “Flutter” serta

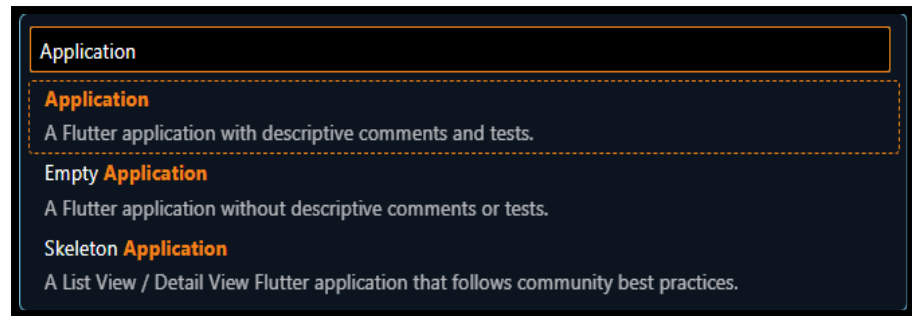


“Dart”.

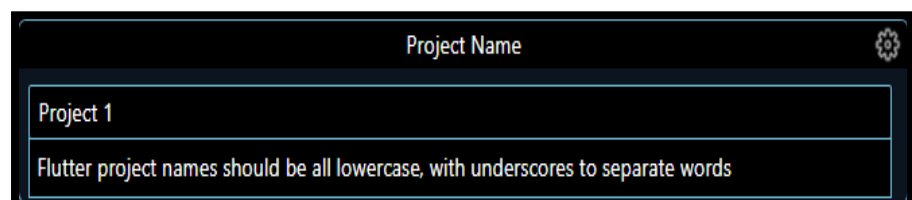
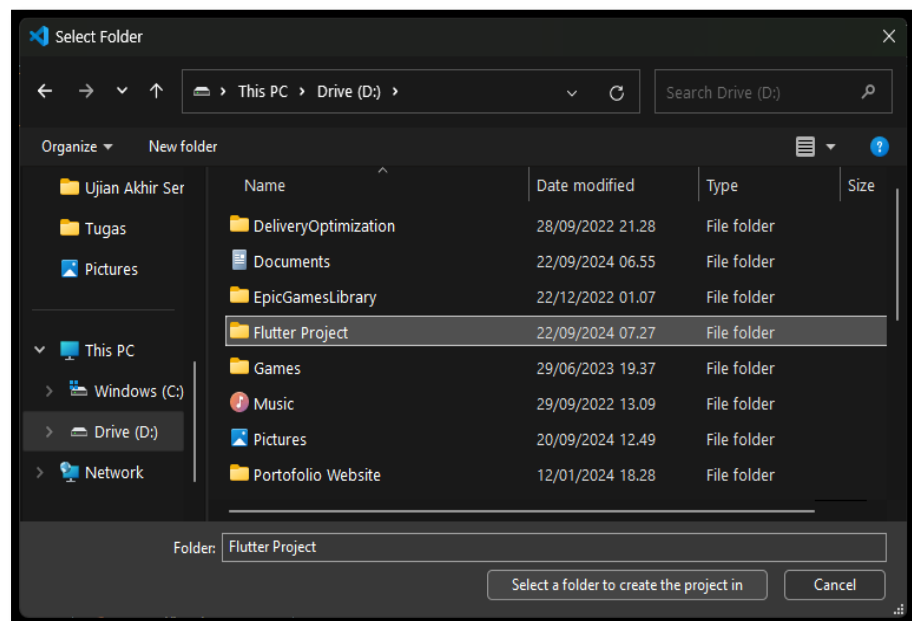
## 2.2.2. Uji Coba

- Membuat proyek pada “Visual Studio Code”, pilih “Flutter: New Project, dan pilih “Application”.

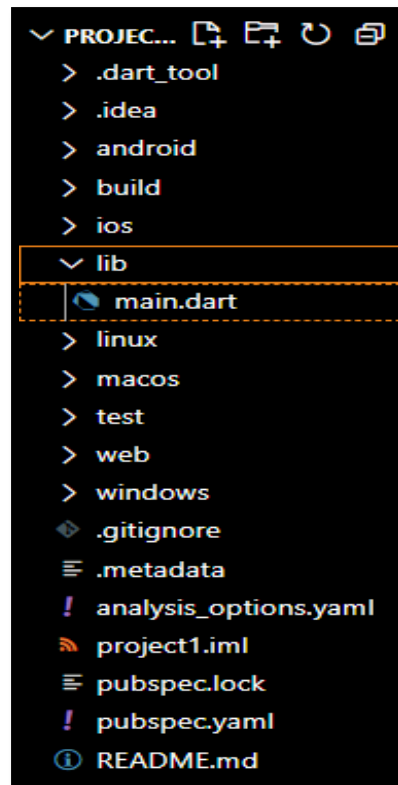




- Menentukan directory penyimpanan proyek akan disimpan dan memberikan nama folder proyek.



- Akan muncul file-file di dalam folder proyek seperti dalam gambar dan pilih “main.dart” pada folder “lib”.



- Berikut kode pemrograman di dalam file “main.dart”.

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

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

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: const
Color.fromARGB(255, 226, 8, 8)),
```

```

        useMaterial3: true,
      ),
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});

  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor:
Theme.of(context).colorScheme.inversePrimary,
        title: Text(widget.title),
      ),
      body: Center(

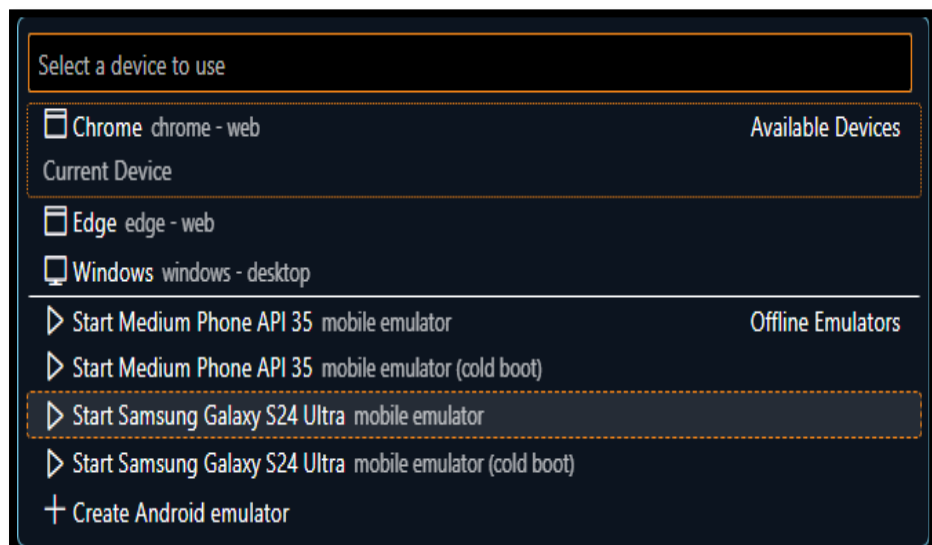
```

```

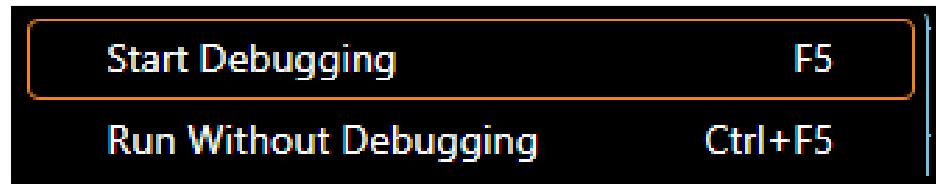
child: Column(
  mainAxisAlignment: MainAxisAlignment.center,
  children: <Widget>[
    const Text(
      'Hello World',
    ),
    Text(
      '$_counter',
      style: Theme.of(context).textTheme.headlineMedium,
    ),
  ],
),
floatingActionButton: FloatingActionButton(
  onPressed: _incrementCounter,
  tooltip: 'Increment',
  child: const Icon(Icons.add),
),
);
}
}

```

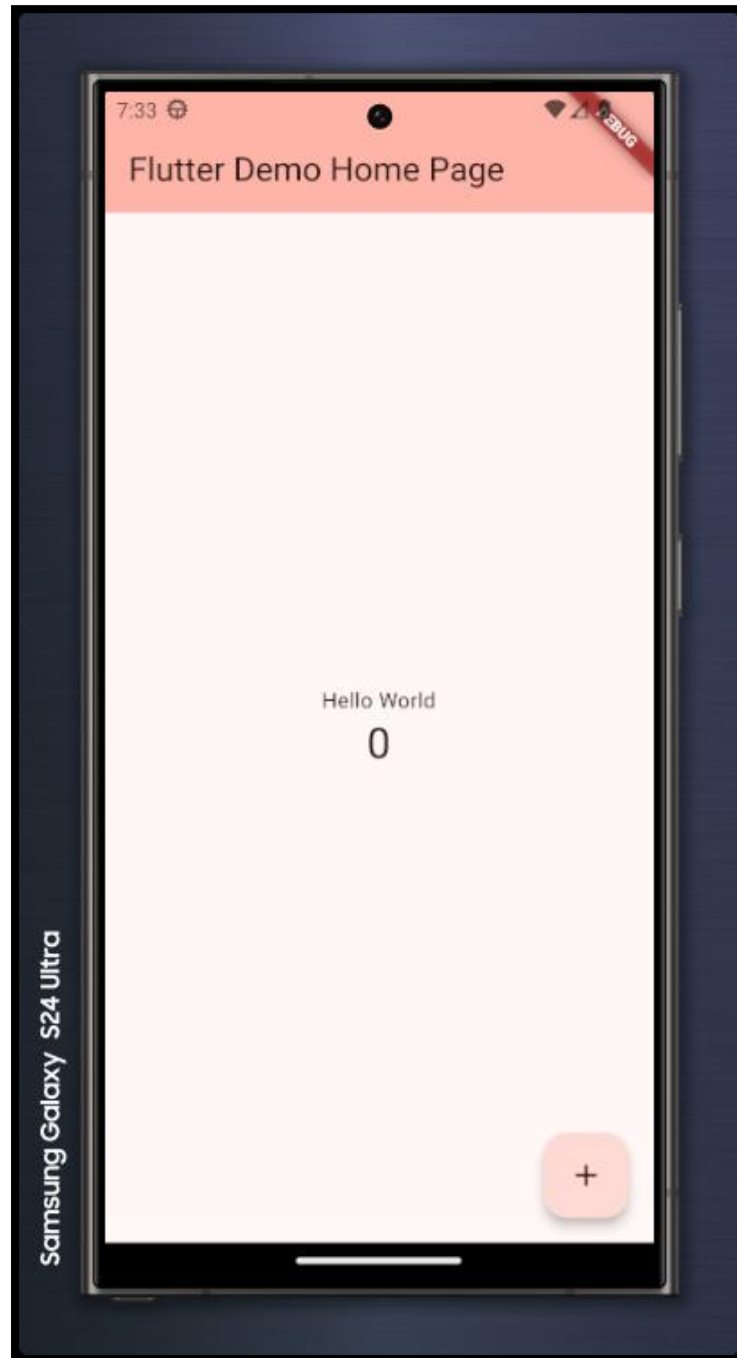
- Memilih device untuk melakukan debugging dan pilih “mobile emulator”.



- Jika device sudah dilakukan debugging sampai muncul, kemudian melakukan “Start Debugging” kode pemrograman untuk dikoneksikan dengan device.



- Berikut hasil dari kode pemrograman tersebut.



## REFERENSI

<https://primakara.ac.id/blog/info-teknologi/Membuat-web-dengan-python>  
<https://aws.amazon.com/id/what-is/python/>  
<https://www.digitalocean.com/community/tutorials/jsp-example-tutorial-for-beginners>  
<https://www.tutorialpedia.net/pengertian-jsp-java-server-pages/>  
<https://www.dicoding.com/blog/apa-itu-node-js/>  
<https://www.biznetgio.com/news/apa-itu-php>  
<https://id.jobstreet.com/id/career-advice/article/php-pengertian-contoh-kelebihan>  
<https://codingstudio.id/blog/asp-adalah/>  
<https://www.niagahoster.co.id/blog/kotlin-adalah/>  
<https://www.petanikode.com/belajar-dart/>  
<https://codingstudio.id/blog/mengenal-bahasa-pemrograman-swift/>  
<https://www.computerscience.org/resources/computer-programming-languages/objective-c/>  
<https://aws.amazon.com/id/what-is/net/>  
<https://codingstudio.id/blog/apa-itu-bahasa-pemrograman-c-sharp/>  
<https://www.debart.com/delphi-programming-language/>  
[https://youtu.be/xBTECTbYDwc?si=8NlqrWhK2Z0LA\\_TT](https://youtu.be/xBTECTbYDwc?si=8NlqrWhK2Z0LA_TT)  
<https://youtu.be/WLQchGLu8dk?si=4EiwVyeNOiirgtLA>