



# PERTEMUAN 1 : DASAR PEMROGRAMAN

---

Bahasa Pemrograman

Hemdani Rahendra Herlianto, S.Kom., M.T.I.

Teknik Informatika - S1

Fakultas Teknik

Universitas Pelita Bangsa



## *Study*

2013 - 2017 : Universitas Esa Unggul

2018 - 2020 : Universitas Bina Nusantara

## *Work Experience*

2012 - 2022 : Cinta Kasih Tzu Chi School (IT Head Div)

2018 - 2022 : PT.Quattro Ahli Gmilang (IT Project Manager)

2022 - ... : PT.eSTEP Global (IT consultant)

2022 - ... : PT.Putra Nulah Inovasi Bangsa (Project Development)

- *2017 - Google Educator for Education*

- *2018 - APTIKNAS Group*

- *2022 - Cloud Digital Leader Learning*

- *2022 - Super Mentor DCMF - Deall !*

- *2022 - AWS Cloud Practitioner Essentials Day*

Research Interests in **Internet of Things** and **Machine Learning**

## *Research :*

-Collaborative recommendation system in users of anime films

-Integration of ITIL V3, ISO 20000 & iso 27001: 2013 for it services and security management system

-IoT-Based Student Monitoring System for Smart School Applications

-Business Intelligence for Analysing Late Payment Fees for School

-Arduino Based Real-Time Face Recognition And Tracking System

# Hemdani Rahendra

# Herlianto, S.Kom., M.T.I.

05 Juni 1993

HP/WA

089516905924

Email

hemdani.rahendra@pelitabangsa.ac.id

Address

Jl.Kamal Raya Outer Ring Road No.20  
Cengkareng Timur Jakarta Barat



# Apa itu Bahasa pemrograman

Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer atau bahasa pemrograman komputer, adalah **instruksi standar untuk memerintah komputer**.

Bahasa pemrograman ini merupakan suatu **himpunan dari aturan sintaks dan semantik** yang dipakai untuk mendefinisikan program komputer. (Wikipedia)



# Program

Program: sederetan perintah-perintah yang harus dikerjakan oleh komputer untuk menyelesaikan masalah.

3 level bahasa pemrograman:

- Bahasa tingkat rendah
- Bahasa tingkat menengah
- Bahasa tingkat tinggi



# Bahasa tingkat rendah

- Bahasa mesinBerisi: kode kode mesin yg hanya dapat diinterpretasikan langsung oleh mesin komputer.  
Berupa kode numerik 0 dan 1
- Microcode: sekumpulan instruksi dalam bahasa mesin
  - (+):Eksekusicepat
  - (-):Sulitdipelajarimanusia



# Bahasa tingkat menengah

## Bahasa Assembly

Bahasa simbol dari bahasa mesin

Contoh: ADD, MUL, SUB, dll

## Macro instruksi: sekumpulan kode dalam bahasa assembly

(+) : Eksekusi cepat, masih dapat dipelajari daripada bahasa mesin,  
file kecil

(-) : Tetap sulit dipelajari, program sangat panjang



# Bahasa tingkat tinggi

## The 3rd Generation Programming Language

Lebih dekat dengan bahasa manusia

Memberi banyak fasilitas kemudahan dalam pembuatan program,  
mis.: variabel, tipe data, konstanta, struktur kontrol, loop, fungsi,  
prosedur, dll

Contoh: Pascal, Basic, C++, Java

(+) : Mudah dipelajari, mendekati permasalahan yang akan  
dipecahkan, kode program pendek

(-) : Eksekusi lambat



# Jenis Bahasa Pemrograman

- Bahasa tingkat rendah
  - Bahasa Mesin
  - Bahasa Assembly
- Bahasa tingkat tinggi
  - Berorientasi Objek
  - Prosedural

```
0000000 0000 0001 0001 1010 0010 0001 0004 0128
0000010 0000 import java.io.IOException;
0000020 0000
0000030 0000public class TomcatEmbedded {
0000040 0004
0000050 00e9    private static final String EMPTY = "";
0000060 00fc
0000070 0057    public static void main(String... args)
0000080 8888        throws Exception {
0000090 3b83        File baseFolder = new File(System.getProperty("user.dir"));
00000a0 d61f        File appsFolder = new File(baseFolder, child: "apps");
00000b0 8b06
00000c0 8a18
00000d0 a948
00000e0 3d86
00000f0 8888
0000100 0000
0000110 *
0000120 0000
0000130 0000
000013e
}
Tomcat tomcat = new Tomcat();
tomcat.setBaseDir(baseFolder.getAbsolutePath());
tomcat.setPort(8080);
tomcat.getHost().setAppBase(appsFolder.getAbsolutePath());
// Call the connector to create the default connector.
tomcat.getConnector();

tomcat.addWebapp(EMPTY, docBase: ".");
Wrapper wrapper = tomcat.addServlet(EMPTY, servletName: "hello", new HelloServlet());
wrapper.setLoadOnStartup(1);
wrapper.addMapping( s: "/*");

tomcat.start();
tomcat.getServer().await();
}
```

Indonesia

Sunda

Inggris

China

Jepang

Arab

Melayu

Italia

Thailand

Jerman

Pascal

C#

PHP

Java

Objective-C

Ruby

C

Python

Swift

R

# Ranah Bahasa Pemrograman

- Web Development



- 
- Mobile Development



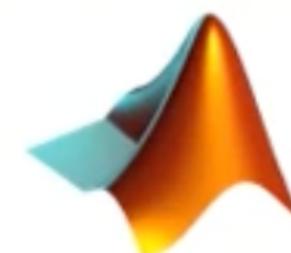
- 
- Game Development



- 
- Desktop Development



- 
- Data Analyst



- 
- Embedded System Development





# STRUKTUR DASAR BAHASA PEMROGRAMAN

## 1. Pernyataan (Statements):

- Pernyataan adalah instruksi-instruksi yang dieksekusi oleh program. Mereka bisa termasuk operasi matematika, penugasan nilai ke variabel, pemanggilan fungsi, dan banyak lagi.
- Contoh:

python

Copy code

```
a = 5 # Menugaskan nilai 5 ke variabel a
print("Hello, world!") # Mencetak teks ke layar
```

## 2. Variabel:

- Variabel digunakan untuk menyimpan data atau nilai. Mereka memberikan nama kepada nilai sehingga dapat diakses dan dimanipulasi dalam program.
- Contoh:

python

Copy code

```
x = 10 # x adalah variabel yang menyimpan nilai 10
nama = "John" # nama adalah variabel yang menyimpan string "John"
```



# STRUKTUR DASAR BAHASA PEMROGRAMAN

## 3. Tipe Data:

- Tipe data menggambarkan jenis nilai yang dapat disimpan dalam variabel. Contoh tipe data umum meliputi bilangan bulat (integer), pecahan (float), string, dan boolean.
- Contoh:

python

Copy code

```
umur = 25 # Integer
gaji = 3000.50 # Float
nama = "Alice" # String
benar = True # Boolean
```

## 4. Komentar:

- Komentar digunakan untuk memberikan penjelasan atau dokumentasi dalam kode. Komentar tidak dijalankan oleh program dan hanya untuk manusia.
- Contoh:

python

Copy code

```
# Ini adalah komentar
```



# STRUKTUR DASAR BAHASA PEMROGRAMAN

## 5. Struktur Kontrol:

- Struktur kontrol digunakan untuk mengatur alur eksekusi program. Ini termasuk pernyataan kondisional (if-else), perulangan (loop), dan fungsi.
- Contoh:

python

Copy code

```
if x > 0:  
    print("x adalah bilangan positif")  
for i in range(5):  
    print(i)
```

## 6. Fungsi:

- Fungsi adalah blok kode yang dapat dipanggil dengan nama tertentu. Mereka digunakan untuk mengelompokkan dan merencanakan tindakan tertentu.
- Contoh:

python

Copy code

```
def hitung_luas_persegi(sisi):  
    return sisi * sisi
```



# STRUKTUR DASAR BAHASA PEMROGRAMAN

## 7. Input dan Output:

- Program dapat berinteraksi dengan pengguna atau lingkungannya melalui input dan output. Input adalah data yang dimasukkan ke dalam program, sedangkan output adalah hasil keluaran dari program.
- Contoh:

python

Copy code

```
nama = input("Masukkan nama Anda: ")
print("Halo, " + nama + "!")
```

## 8. Komentar Blok:

- Beberapa bahasa pemrograman mendukung komentar blok yang digunakan untuk dokumentasi lebih besar dalam kode.
- Contoh (Python):

python

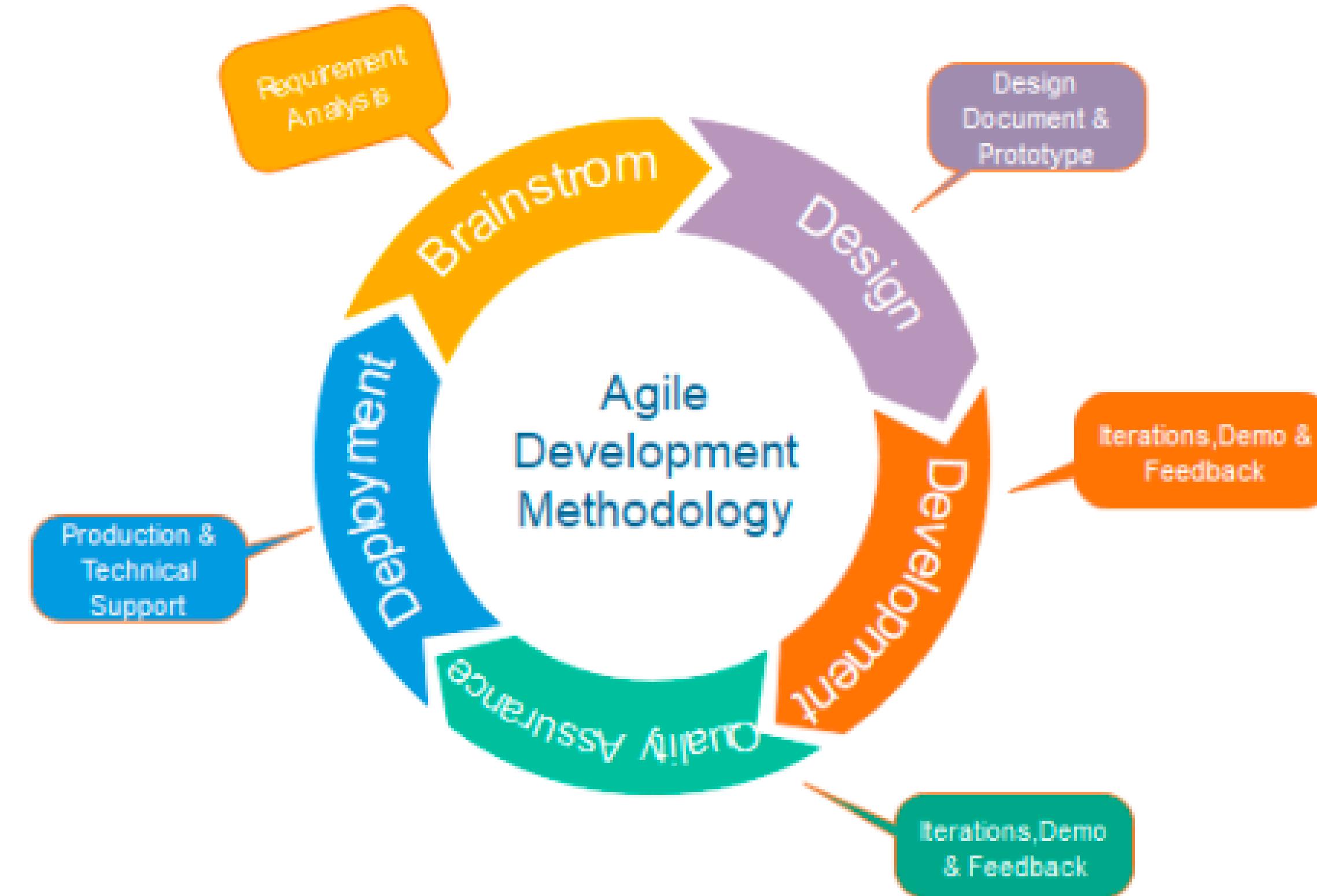
Copy code

```
"""
```

```
Ini adalah komentar blok yang dapat berisi dokumentasi lebih lanjut tentang
Misalnya, deskripsi singkat tentang apa yang program ini lakukan.
```

```
"""
```

# PENGEMBANGAN PERANGKAT LUNAK



*Fig. Agile Model*



# PENGEMBANGAN PERANGKAT LUNAK

- **Analisis Kebutuhan (Requirement Analysis):**

- Langkah pertama dalam pengembangan perangkat lunak adalah memahami kebutuhan pengguna dan masalah yang ingin dipecahkan.
- Ini melibatkan interaksi dengan pemangku kepentingan untuk mengidentifikasi persyaratan fungsional dan non-fungsional perangkat lunak.

- **Perancangan (Design):**

- Setelah persyaratan dianalisis, langkah berikutnya adalah merancang arsitektur perangkat lunak.
- Ini mencakup perencanaan struktur perangkat lunak, aliran kerja, dan antarmuka pengguna.
- Pemodelan seperti diagram aliran data, diagram kelas, dan diagram sekuen sering digunakan dalam tahap ini.

- **Pengembangan (Development):**

- Tahap ini melibatkan implementasi kode perangkat lunak berdasarkan desain yang telah dibuat.
- Programmers menulis kode sesuai dengan spesifikasi yang ada dan menggunakan bahasa pemrograman tertentu.

- **Pengujian (Testing):**

- Setelah pengembangan selesai, perangkat lunak harus diuji secara ekstensif untuk memastikan bahwa itu berfungsi dengan benar.
- Pengujian melibatkan identifikasi dan perbaikan bug atau masalah yang mungkin ditemukan dalam perangkat lunak.
- Berbagai jenis pengujian, termasuk pengujian fungsional, pengujian integrasi, dan pengujian kinerja, dapat dilakukan



# PENGEMBANGAN PERANGKAT LUNAK

- **Implementasi (Deployment):**

- Setelah perangkat lunak diuji dan disetujui, itu dapat diimplementasikan dalam lingkungan produksi.
- Ini melibatkan penyalinan perangkat lunak ke server atau infrastruktur yang digunakan oleh pengguna akhir.

- **Pemeliharaan (Maintenance):**

- Perangkat lunak yang diimplementasikan akan membutuhkan pemeliharaan terus-menerus. Ini mencakup perbaikan bug, peningkatan fitur, dan pembaruan keamanan.
- Pengembang perangkat lunak harus siap untuk merespons perubahan dan masalah yang muncul seiring waktu.

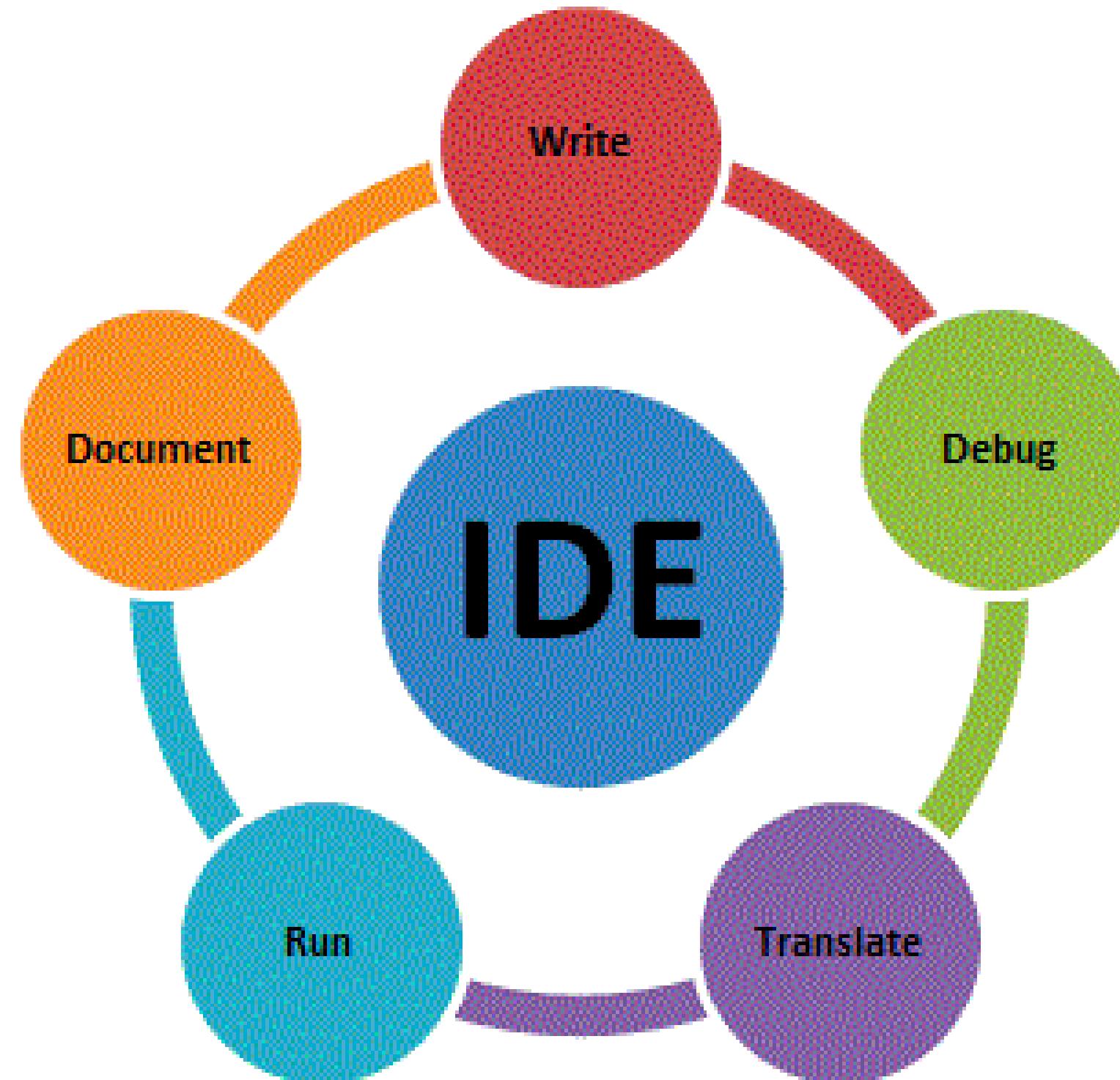
- **Dokumentasi (Documentation):**

- Selama seluruh proses pengembangan, penting untuk mendokumentasikan kode, desain, dan semua aspek perangkat lunak.
- Dokumentasi membantu pemeliharaan yang lebih mudah dan memungkinkan pengguna untuk memahami cara menggunakan perangkat lunak.

- **Manajemen Proyek (Project Management):**

- Manajemen proyek yang baik adalah kunci untuk pengembangan perangkat lunak yang sukses.
- Ini melibatkan perencanaan, pengendalian, dan pelacakan proyek, serta alokasi sumber daya yang efisien.

# INTEGRATED DEVELOPMENT ENVIRONMENT



Lingkungan pengembangan, juga dikenal sebagai **Integrated Development Environment (IDE)** atau lingkungan pengembangan terpadu, adalah perangkat lunak yang dirancang untuk membantu pengembang perangkat lunak dalam menulis, menguji, dan memelihara kode.





# INTEGRATED DEVELOPMENT ENVIRONMENT

## Editor Kode (Code Editor):

- komponen inti dalam IDE yang digunakan untuk menulis, mengedit, dan mengelola kode sumber.
- Fitur-fitur seperti penyorotan sintaksis, auto-completion, dan indentasi otomatis membuat penulisan kode lebih efisien.

The screenshot displays a dark-themed IDE interface. At the top, a menu bar includes File, Edit, Find, Navigate, Project, Help, MENU, TABS, and COLLABORATORS. Below the menu is a tab bar with several files: arrow.js, audio.js, camera-shake.js, elem.js, easing.js (which is selected), entity-pool.js, falling-leaves.js, and falling-leaves.js again. The main area is a TEXT EDITOR containing JavaScript code for easing functions. The code defines various easing functions like Linear, QuadraticIn, QuadraticOut, CubicIn, CubicOut, QuarticIn, QuarticOut, QuinticIn, QuinticOut, and QuinticInOut. The code editor shows syntax highlighting and line numbers from 1 to 55. On the left is a FILES sidebar listing project files such as audio, bob, bow, environment, items, scripts, and tween. At the bottom is a STATUS BAR with the text "Selected asset 'easing.js' (id: 91233)" and "Connected".

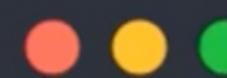
```
pc.extend(pc, function () {  
    var Linear = function (k) {  
        return k;  
    };  
  
    var QuadraticIn = function (k) {  
        return k * k;  
    };  
  
    var QuadraticOut = function (k) {  
        return k * (2 - k);  
    };  
  
    var QuadraticInOut = function (k) {  
        if ((k *= 2) < 1) {  
            return 0.5 * k * k;  
        }  
        return -0.5 * (-k * (k - 2) - 1);  
    };  
  
    var CubicIn = function (k) {  
        return k * k * k;  
    };  
  
    var CubicOut = function (k) {  
        return --k * k * k + 1;  
    };  
  
    var CubicInOut = function (k) {  
        if ((k *= 2) < 1) return 0.5 * k * k * k;  
        return 0.5 * ((k -= 2) * k * k + 2);  
    };  
  
    var QuarticIn = function (k) {  
        return k * k * k * k;  
    };  
  
    var QuarticOut = function (k) {  
        return 1 - ( --k * k * k * k );  
    };  
  
    var QuarticInOut = function (k) {  
        if ((k *= 2) < 1) return 0.5 * k * k * k * k;  
        return - 0.5 * ((k -= 2) * k * k * k - 2);  
    };  
  
    var QuinticIn = function (k) {  
        return k * k * k * k * k;  
    };  
  
    var QuinticOut = function (k) {  
        return --k * k * k * k * k + 1;  
    };  
  
    var QuinticInOut = function (k) {  
        if ((k *= 2) < 1) return 0.5 * k * k * k * k * k;  
        return - 0.5 * ((k -= 2) * k * k * k * k - 2);  
    };  
});
```



# INTEGRATED DEVELOPMENT ENVIRONMENT

## Syntax

- Dalam ilmu komputer, sintaksis bahasa komputer adalah **seperangkat aturan** yang mendefinisikan **kombinasi simbol** yang dianggap sebagai dokumen terstruktur dengan benar atau fragmen **dalam bahasa itu**.



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



```
public class HelloWorld {  
    public static void main(Strings[] args) {  
        System.out.print("Hello World");  
    }  
}
```



# INTEGRATED DEVELOPMENT ENVIRONMENT

## Manajemen Proyek (Project Management):

- Ini memungkinkan pengembang untuk mengatur semua file, folder, dan sumber daya proyek dalam satu tempat.
- Memungkinkan untuk membuat, mengedit, dan mengelola berbagai bagian dari proyek dengan mudah.

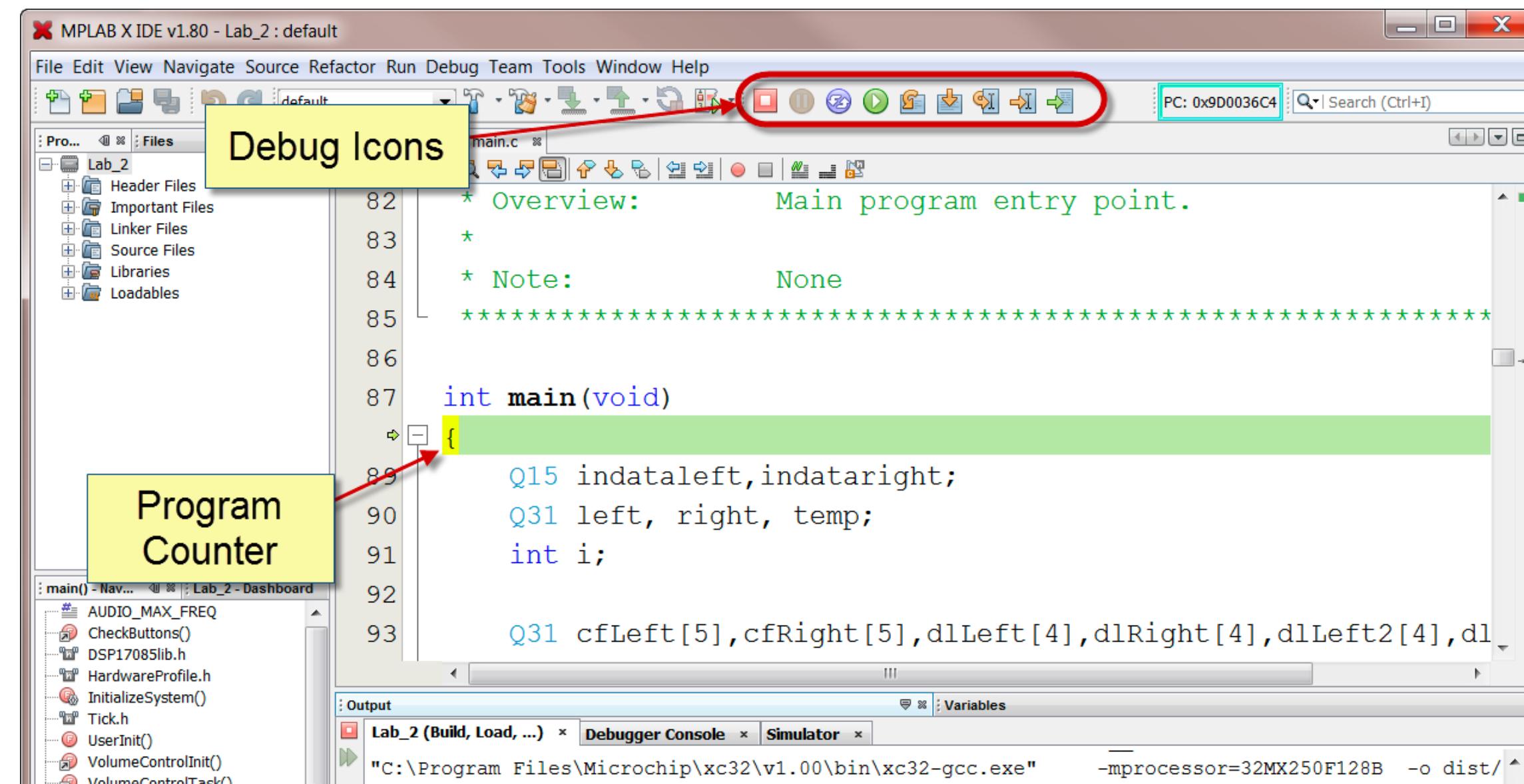
The screenshot shows the Asana interface with the title "Startup Hub". On the left, there is a sidebar with icons for Mode, List, Tools, List, Board, Columns, Comments, and File. The main area is divided into five sections:

- Company strategy**: Contains cards for "Goals" (4 cards), "Values" (2 boards, 1 card), "Strategic plan 2020.pdf" (Download - 31 KB), and "Lean canvas" (1 board, 21 cards).
- About the product**: Contains a large purple circular icon and cards for "Value proposition" (2 boards, 21 cards) and "Target Audience".
- Product**: Contains cards for "Key features" (1 board, 21 cards), "Product roadmap" (1 board, 21 cards), and "Ideas" (24 cards).
- Growth & marketing**: Contains cards for "Backlog" (1 board, 27 cards), "Upcoming experiments" (27 cards), and "Growth playbook" (2 boards, 27 cards).
- Team**: Shows profiles for "Paul Morris - CEO" (1 card), "Suraj Sahni - VP, Product" (1 card), "Kristy Williams - Product" (1 card), and "Sarah Li - Product Designer" (1 card). It also indicates 10 boards.

# INTEGRATED DEVELOPMENT ENVIRONMENT

## Debugger:

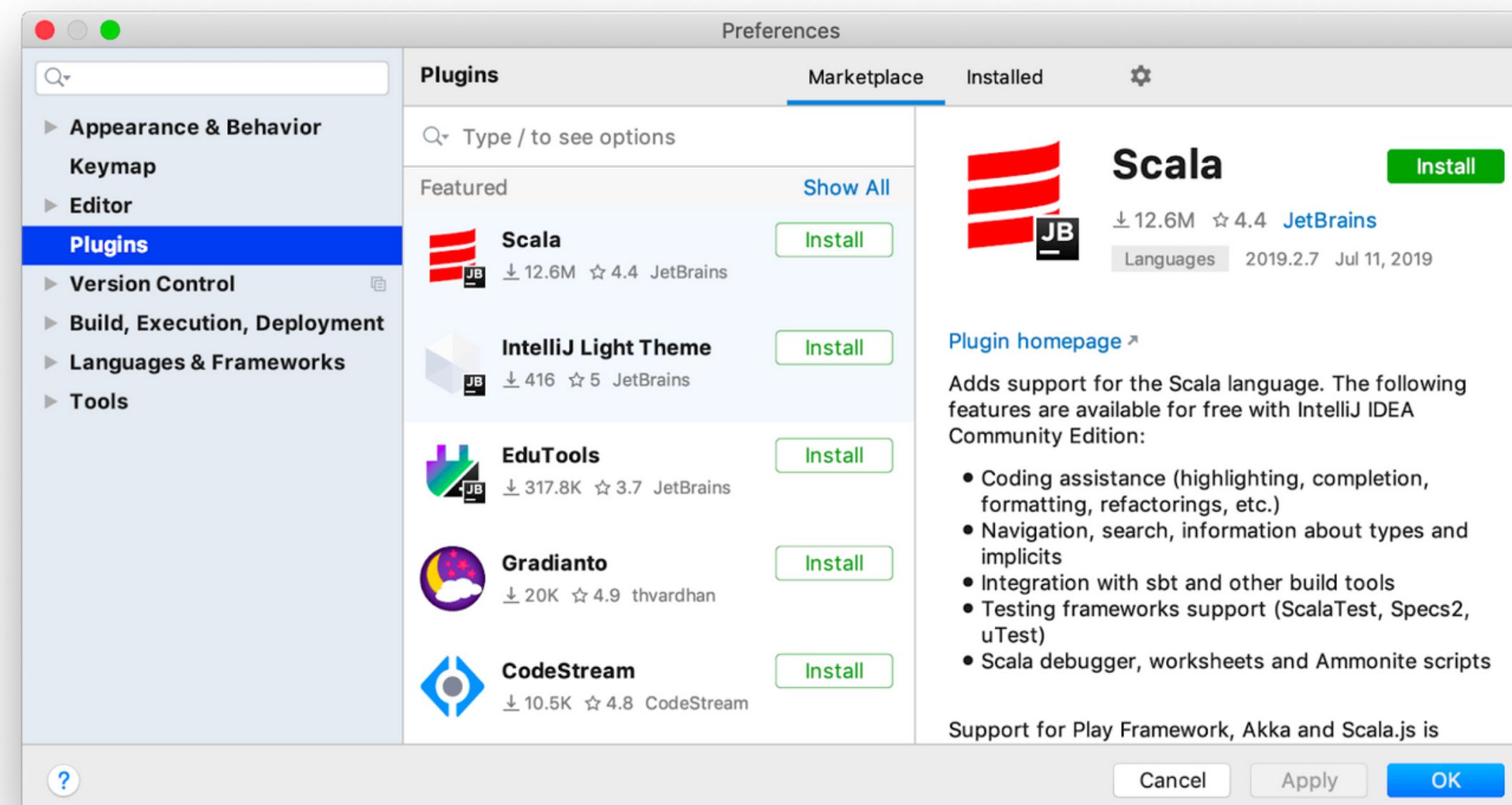
- Debugger adalah alat yang membantu pengembang dalam melacak dan memperbaiki bug dalam kode.
- Ini memungkinkan pengembang untuk mengeksekusi kode baris demi baris, melihat nilai variabel, dan memahami alur eksekusi program.



# INTEGRATED DEVELOPMENT ENVIRONMENT

## Plugin dan Ekstensi (Plugins and Extensions):

- IDE sering mendukung plugin atau ekstensi yang memungkinkan pengembang untuk menambahkan fungsionalitas tambahan sesuai dengan kebutuhan mereka.





# INTEGRATED DEVELOPMENT ENVIRONMENT

- **Integrasi Sistem Kontrol Versi (Version Control Integration):**
  - IDE sering terintegrasi dengan sistem kontrol versi seperti Git, SVN, atau Mercurial.
  - Ini memudahkan pengembang dalam melacak perubahan kode, berkolaborasi dengan tim, dan mengelola versi kode sumber.
- **Pengelolaan Dependensi (Dependency Management):**
  - IDE dapat menyediakan alat untuk mengelola dan mengunduh dependensi atau pustaka eksternal yang dibutuhkan oleh proyek.
  - Ini berguna terutama dalam pengembangan perangkat lunak yang menggunakan pustaka-pustaka pihak ketiga.
- **Pustaka dan Framework Terintegrasi:**
  - Beberapa IDE menyediakan pustaka dan framework terintegrasi yang dapat digunakan untuk mempercepat pengembangan dengan menyediakan kode contoh dan fungsi bawaan.
- **Alat Pemantauan Kinerja (Performance Monitoring Tools):**
  - Untuk aplikasi yang memerlukan pengoptimalan kinerja, IDE dapat menyertakan alat pemantauan kinerja untuk membantu identifikasi masalah kinerja.



# INTEGRATED DEVELOPMENT ENVIRONMENT

- **Console dan Output Debugging:**
  - IDE biasanya memiliki konsol atau panel output yang menampilkan pesan kesalahan, hasil operasi, atau informasi debug selama eksekusi program.
- **Integrasi Testing (Testing Integration):**
  - Untuk pengujian otomatis, IDE dapat terintegrasi dengan kerangka kerja pengujian dan menyediakan alat untuk menjalankan dan menganalisis hasil pengujian.
- **Dokumentasi Otomatis (Automated Documentation):**
  - Beberapa IDE dapat menghasilkan dokumentasi otomatis berdasarkan komentar dalam kode, memudahkan dalam dokumentasi proyek.
  -

# Kunci belajar Bahasa Pemrograman

- SUKAI...CINTAI...NIKMATI
- Pilih 1 Bahasa pemrograman
- Latihan...Latihan...Latihan
- KONSISTEN



# PRACTICE

[HTTPS://WWW.SOLEARN.COM/HOME](https://www.sololearn.com/home)

[HTTPS://WWW.W3SCHOOLS.COM](https://www.w3schools.com)



# Terimakasih

Hemdani Rahendra Herlianto, S.Kom., M.T.I

 +6289 516905924

 hemdani.rahendra@pelitabangsa.ac.id