



Pemrograman Berorientasi Objek

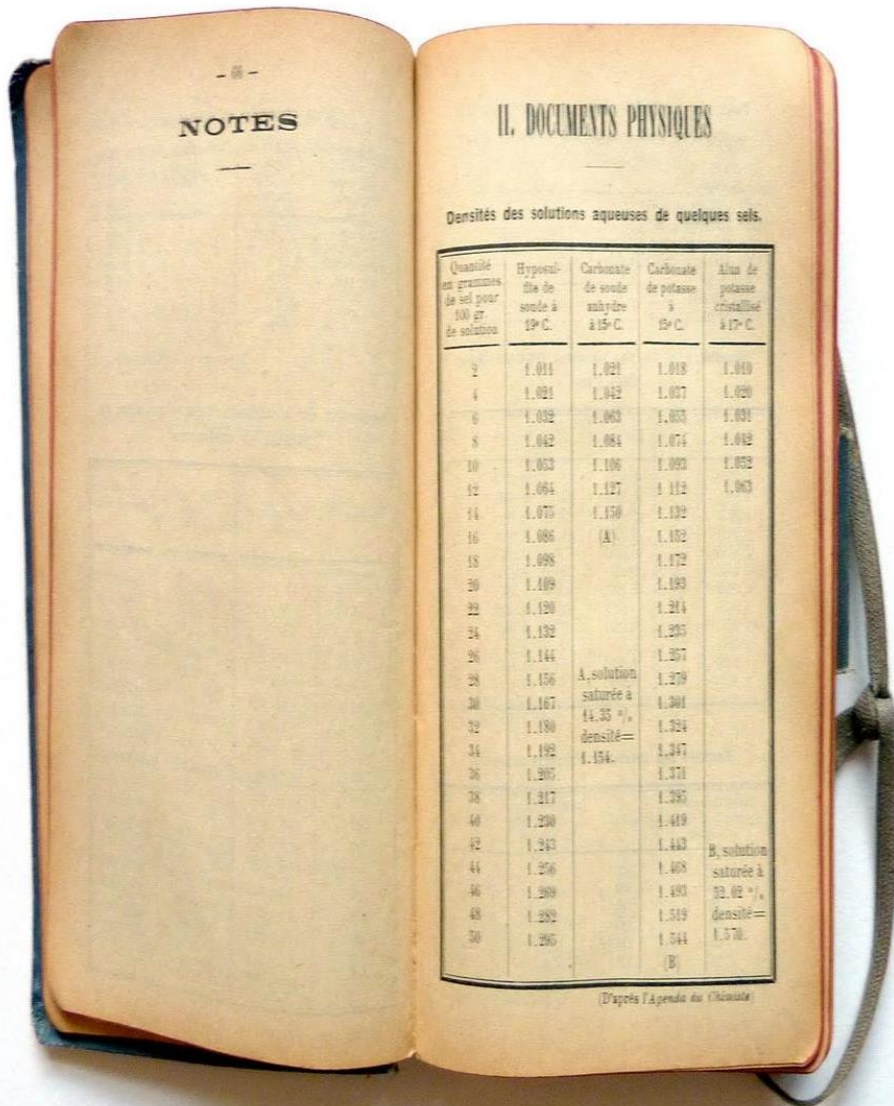


Pertemuan 1

Pengantar Pemrograman Objek

Pemateri : Chrismikha Hardyanto S.Kom., M.Kom.

KONTEN PERKULIAHAN

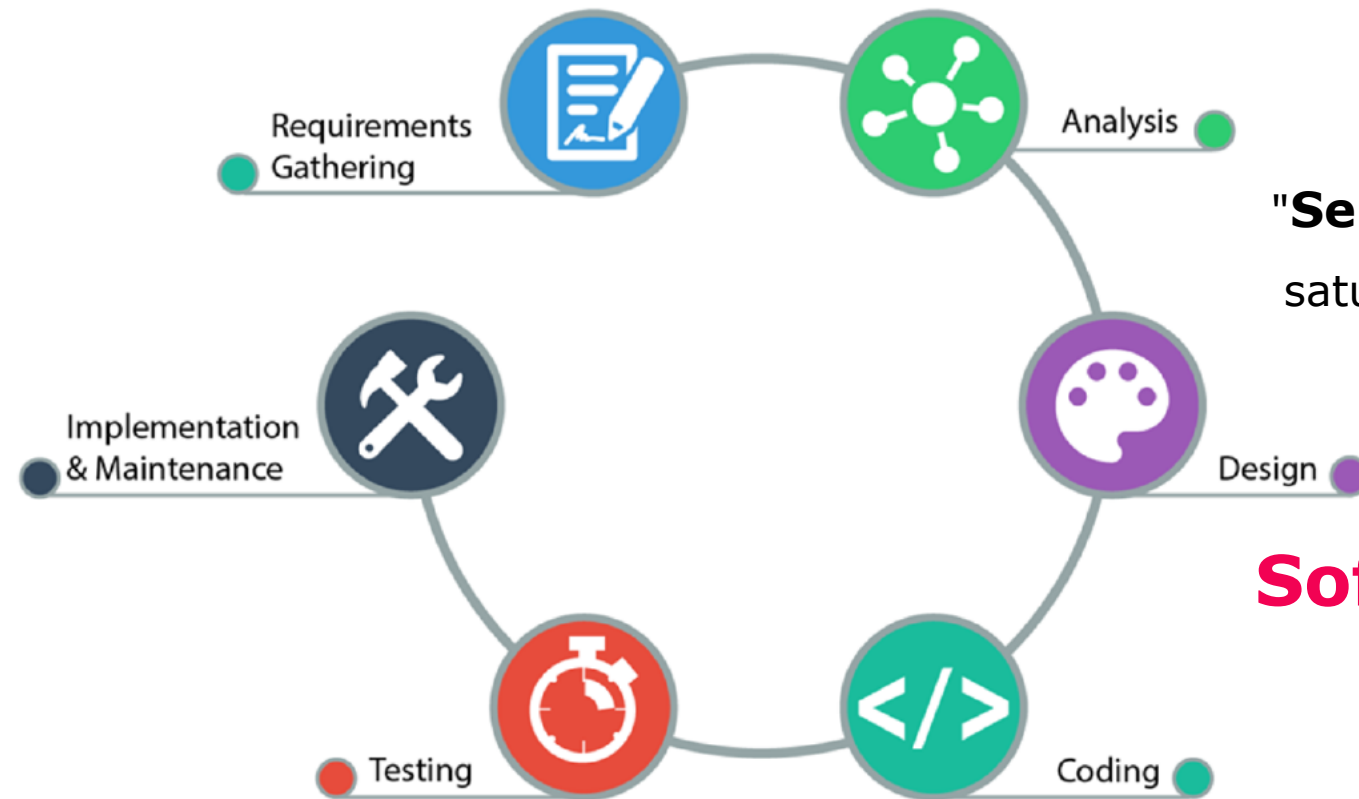


- Proses Pembangunan Perangkat Lunak / SDLC
- Pemrograman Prosedural VS Pemrograman Objek
- Konsep Dasar PBO
- Pengenalan JAVA
- Instalasi JDK dan Netbeans
- Program Pertama JAVA



Software Programming

Bagaimana Proses Pembangunan Perangkat Lunak ??



"**Sekumpulan kegiatan** dan **keterhubungannya** satu sama lain untuk mendukung proses pembangunan dari sebuah perangkat lunak"

Software Development Life Cycle (SDLC)

[Implementasi kedalam Bahasa Pemrograman]

Sumber gambar : [google.com](https://www.google.com)

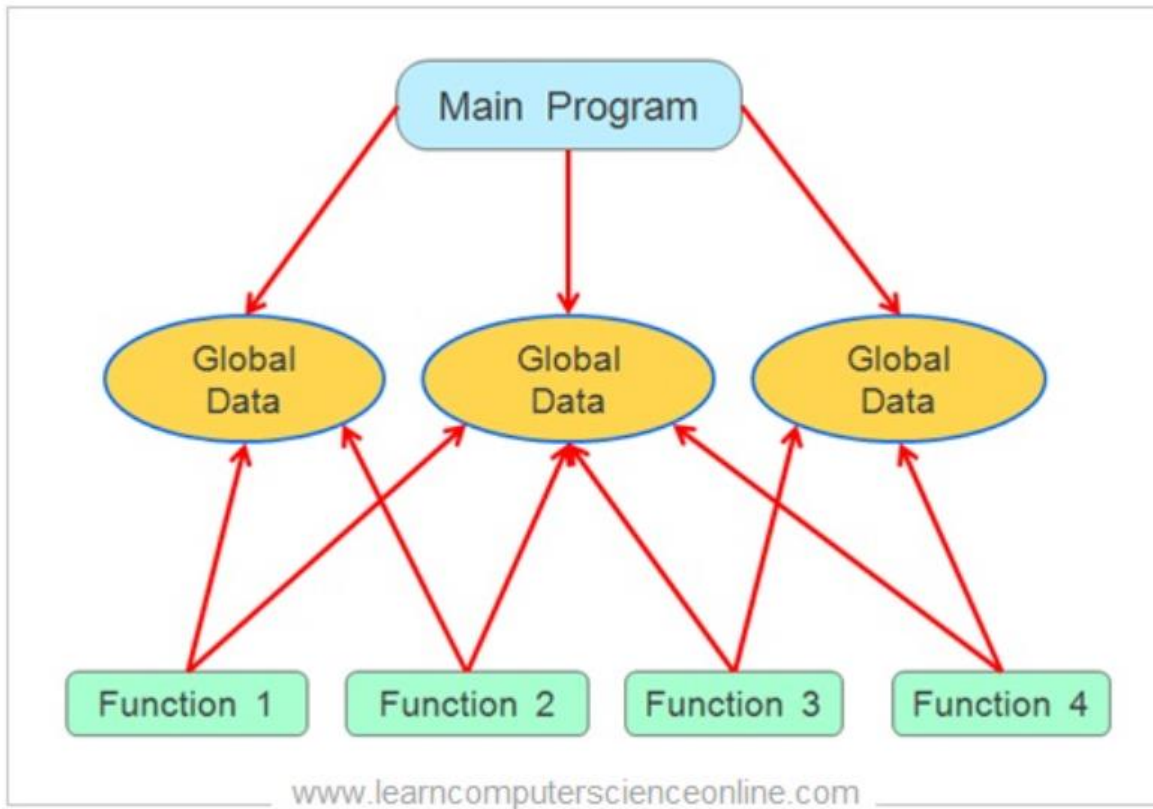
Pola Pikir / Pendekatan didalam pembangunan perangkat lunak dibagi menjadi **2** , yaitu :

- 1. Berorientasi Proses (Prosedural)**
- 2. Berorientasi Objek**



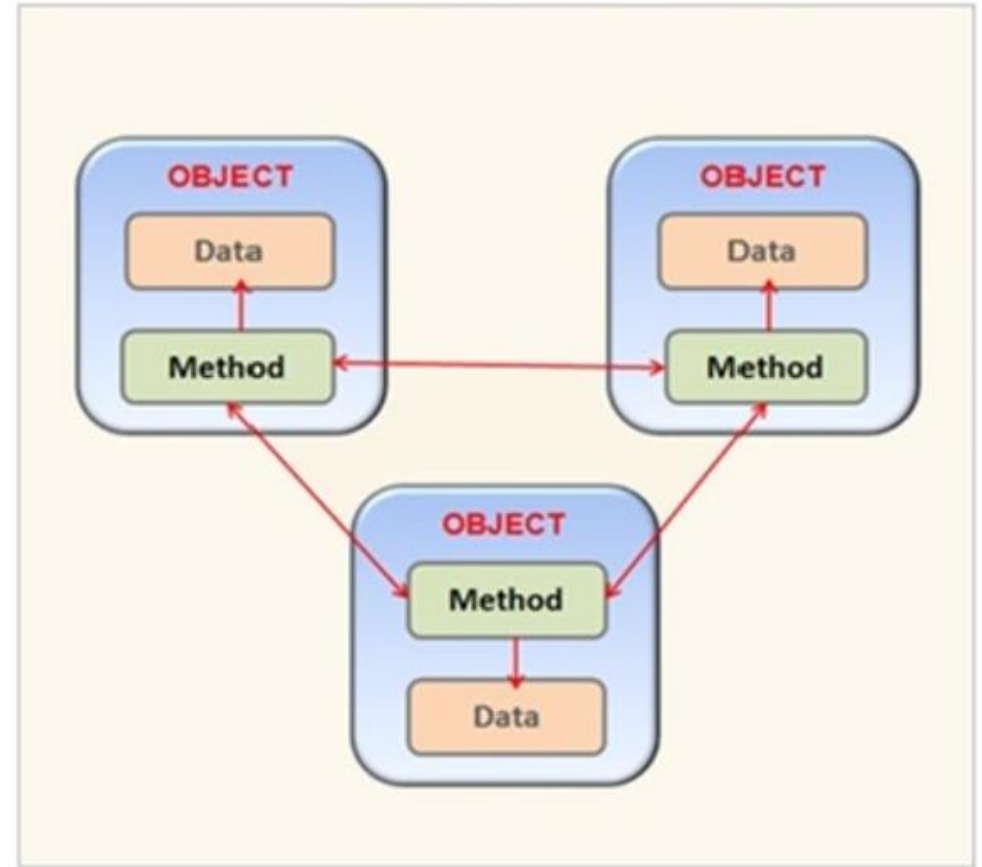
[Pola Pikir ini juga digunakan dalam **PEMROGRAMAN**]

Pendekatan Dalam Pemrograman



Pemrograman Prosedural :

Fokus terhadap "**proses**" yang digunakan

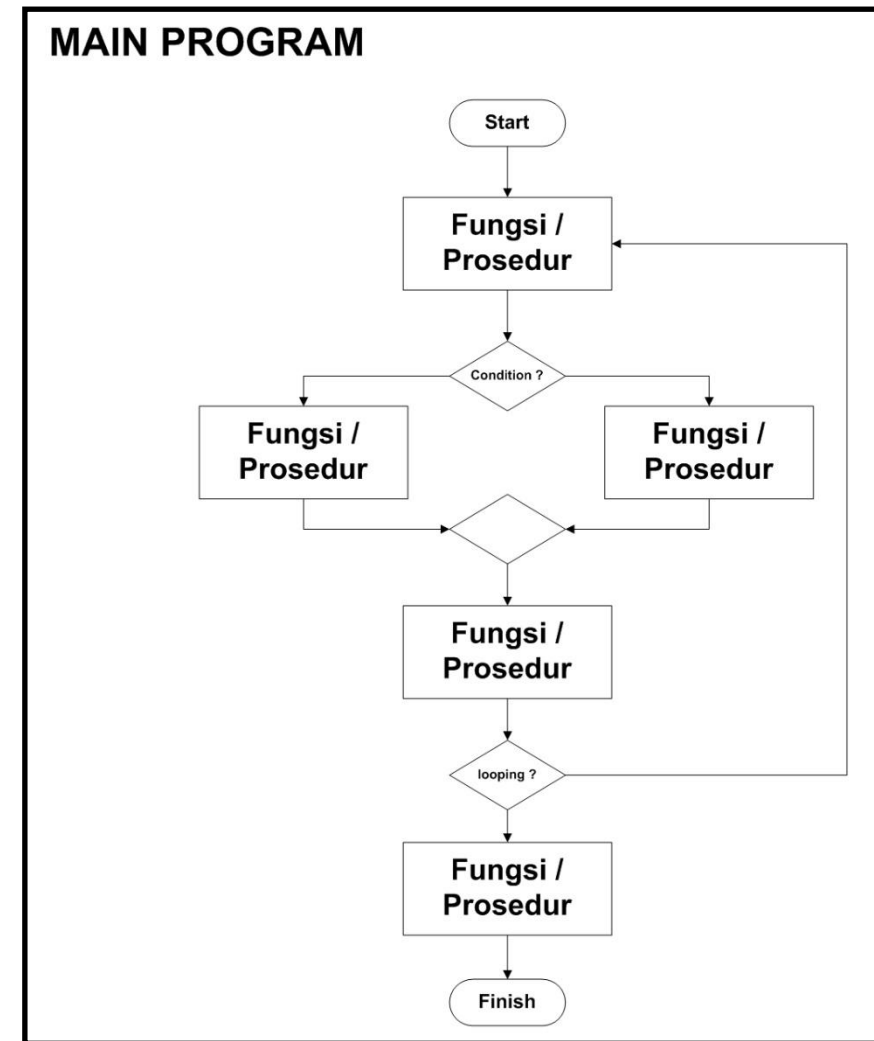


Pemrograman Objek :

Fokus terhadap **Objek** dan interaksinya

Pemrograman Prosedural (konsep)

1. Serangkaian tugas diselesaikan dalam bentuk **fungsi atau prosedur**.
2. Cara pandang → program adalah suatu **urutan instruksi/proses** yang terstruktur.
3. Adanya pemecahan program utama (yang kompleks) menjadi modul-modul kecil.

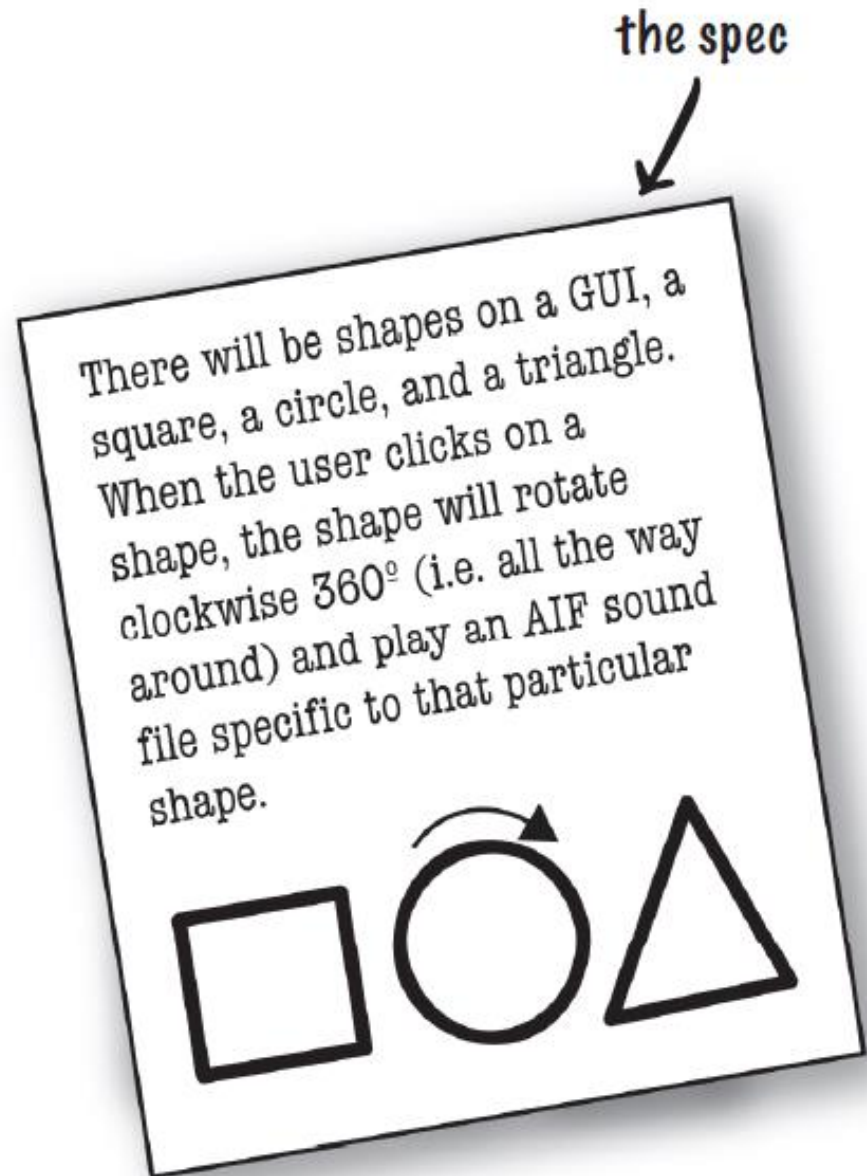


Pemrograman Objek ??

Konsep Pemrograman
Berorientasi Objek akan dibahas di
semester ini

Object**O**riented
Programming
VS
Procedural
Programming

Kenapa butuh PBO ?



Ada 2 orang bernama Larry dan Brad yang diberikan spec yang sama untuk merebutkan sebuah kursi pekerjaan dari bosnya . Dimisalkan :

Larry = **Berpikir Prosedural**

Brad = **Berpikir Berorientasi Objek**

Kenapa butuh PBO ?

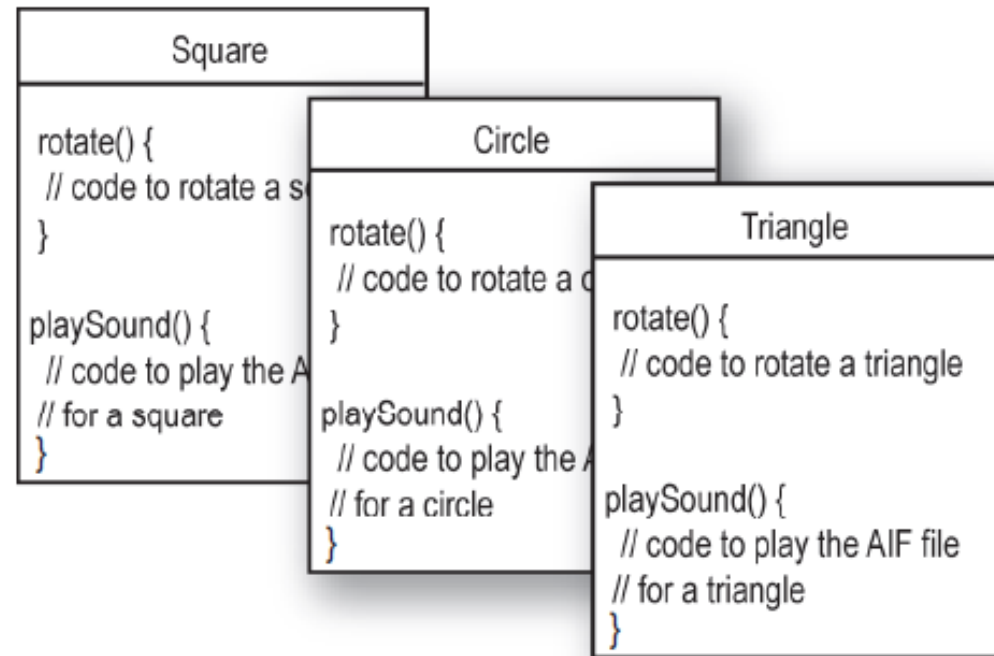
In Larry's cube

As he had done a gazillion times before, Larry set about writing his **Important Procedures**. He wrote **rotate** and **playSound** in no time.

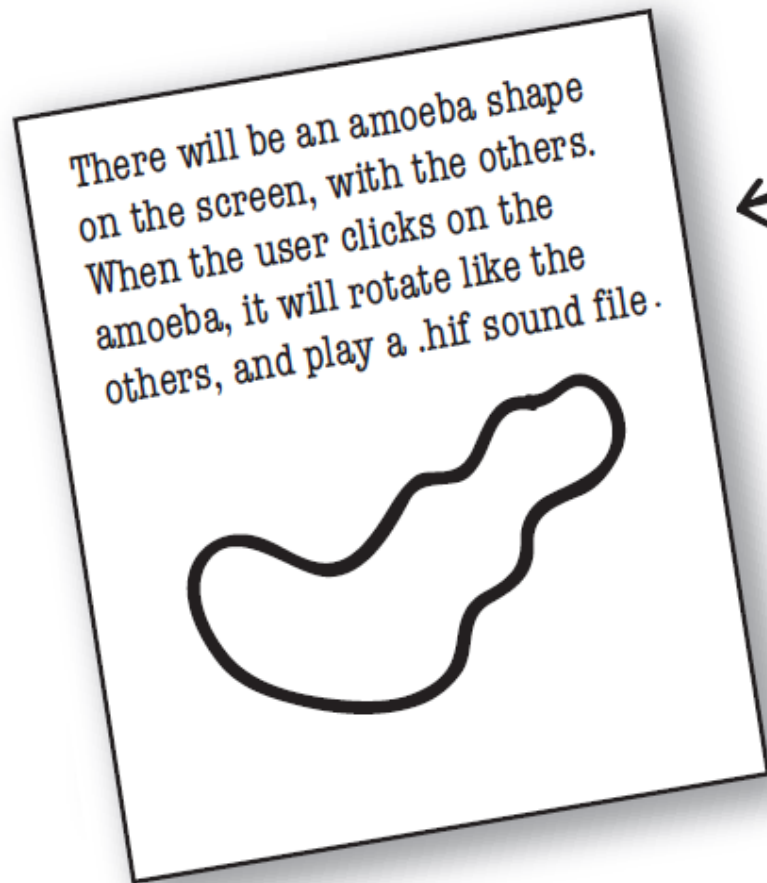
```
rotate(shapeNum) {  
    // make the shape rotate 360°  
}  
  
playSound(shapeNum) {  
    // use shapeNum to lookup which  
    // AIF sound to play, and play it  
}
```

At Brad's laptop at the cafe

Brad wrote a *class* for each of the three shapes



Kenapa butuh PBO ?



← what got added to the spec

Masalah muncul ketika ada **perubahan spesifikasi** pada program yang dikerjakan oleh Larry dan Brad

Kenapa butuh PBO ?

Back in Larry's cube

The rotate procedure would still work; the code used a lookup table to match a shapeNum to an actual shape graphic. But *playSound would have to change*. And what the heck is a .hif file?

```
playSound(shapeNum) {  
    // if the shape is not an amoeba,  
    // use shapeNum to lookup which  
    // AIF sound to play, and play it  
    // else  
    // play amoeba .hif sound  
}
```

It turned out not to be such a big deal, but *it still made him queasy to touch previously-tested code*. Of all people, *he* should know that no matter what the project manager says, *the spec always changes*.

At Brad's laptop at the beach

Brad smiled, sipped his margarita, and *wrote one new class*. Sometimes the thing he loved most about OO was that he didn't have to touch code he'd already tested and delivered. "Flexibility, extensibility,..." he mused, reflecting on the benefits of OO.

Amoeba
<pre>rotate() { // code to rotate an amoeba } playSound() { // code to play the new // .hif file for an amoeba }</pre>

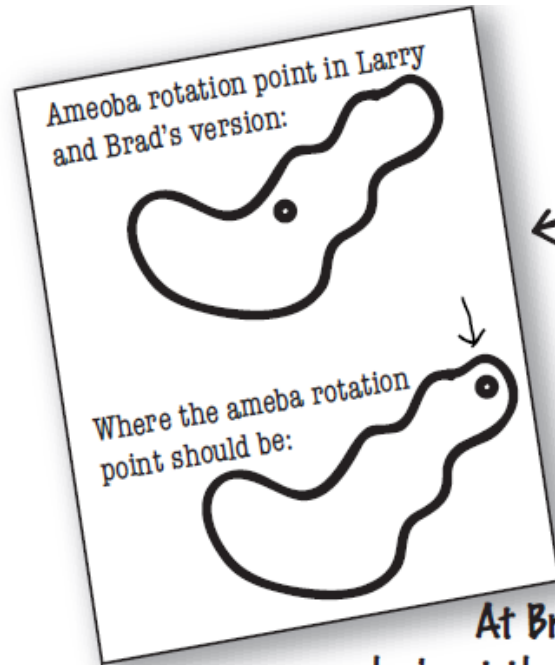
Kenapa butuh PBO ?



Back in Larry's cube

He figured he better add rotation point arguments to the rotate procedure. *A lot of code was affected.* Testing, recompiling, the whole nine yards all over again. Things that used to work, didn't.

```
rotate(shapeNum, xPt, yPt) {  
  // if the shape is not an amoeba,  
  // calculate the center point  
  // based on a rectangle,  
  // then rotate  
  // else  
  // use the xPt and yPt as  
  // the rotation point offset  
  // and then rotate  
}
```



What the spec conveniently forgot to mention

Muncul lagi **perubahan spesifikasi kebutuhan** pada program yang dikerjakan. Rotasi Amoeba Berbeda dengan 3 bentuk pertama

At Brad's laptop on his lawn chair at the Telluride Bluegrass Festival

Without missing a beat, Brad modified the rotate method, but only in the Amoeba class. *He never touched the tested, working, compiled code* for the other parts of the program. To give the Amoeba a rotation point, he added an attribute that all Amoebas would have. He modified, tested, and delivered (wirelessly) the revised program during a single Bela Fleck set.

Amoeba
int xPoint
int yPoint
rotate() { // code to rotate an amoeba // using amoeba's x and y }
playSound() { // code to play the new // .hif file for an amoeba }

Apa yang dapat kita **simpulkan** dari ilustrasi kasus sebelumnya ...

Untuk kasus dimana sering terjadi **perubahan (data / operasi) didalam program**, Paradigma PBO adalah solusi yang tepat untuk mempermudah programmer dalam menuliskan ataupun merawat kode program tersebut.

*keuntungan dari PBO tidak hanya ini, silakan Anda explore secara mandiri...

Jadi Apa itu  Object Oriented Programming ?

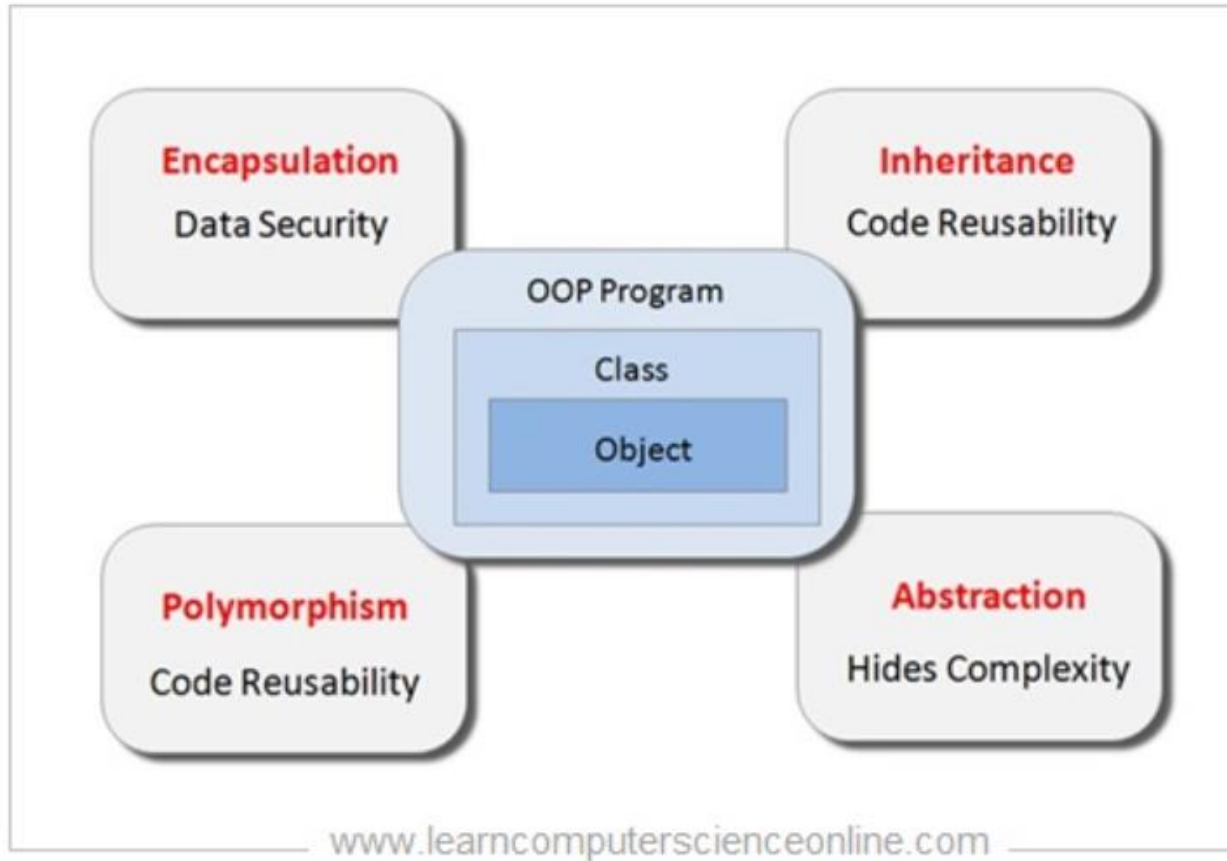
Konsep Dasar PBO

Definisi :

Suatu paradigma pemrograman yang menggunakan **kelas** & **objek** untuk membungkus **data** dan **operasi** yang digunakan sebagai dasar untuk menyelesaikan **masalah - masalah didalam pemrograman** dengan mengacu pada prinsip dasar PBO

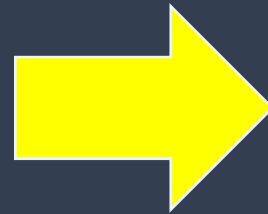
4 Prinsip Dasar PBO

Principles Of OOP



1. **Encapsulation**
(Pembungkusan)
2. **Inheritance**
(Pewarisan)
3. **Polymorphism**
(Banyak Bentuk)
4. **Abstraction**
(Bentuk Abstrak)

Bagaimana Kita Dapat membuat program dengan pendekatan Objek ?



Memilih Bahasa pemrograman yang mendukung PBO

*Saat ini sudah banyak sekali Bahasa pemrograman yang sudah mendukung PBO

Pengenalan Bahasa JAVA

Apa itu bahasa JAVA ?

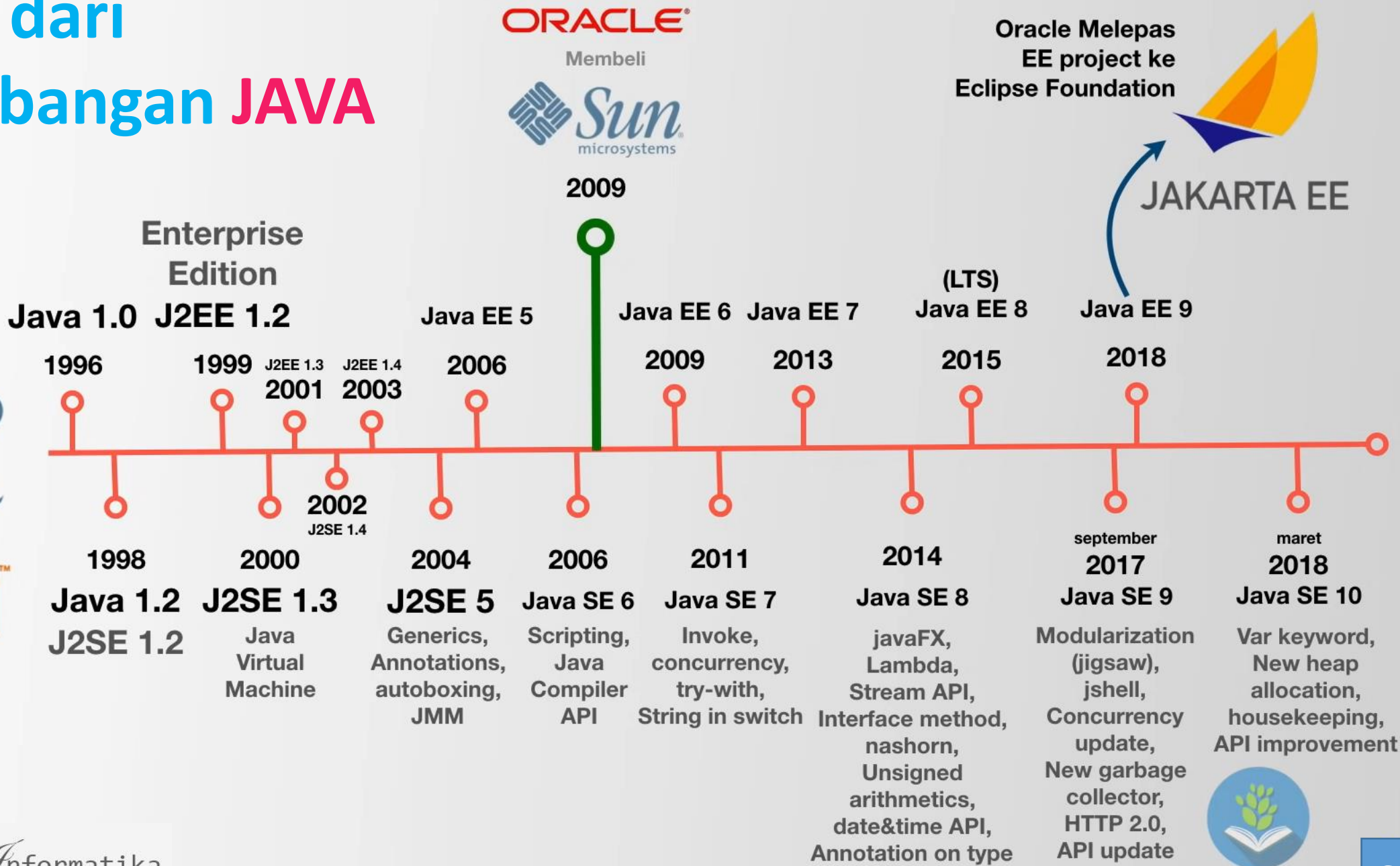
Suatu bahasa **pemrograman tingkat tinggi** (*high level language*) dan **berbasis kelas** (*class based*) yang digunakan untuk mendukung **pemrograman berorientasi objek**.

[sumber : wikipedia]



[James gosling – 1991 – Sun Microsystems]

Sejarah dari Perkembangan **JAVA**



[sumber : youtube-kelas terbuka channel]

Teknologi JAVA

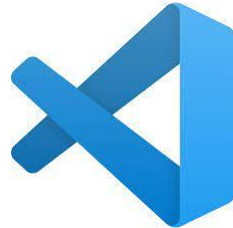
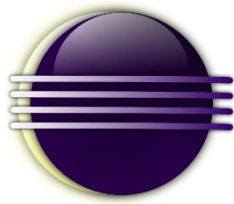
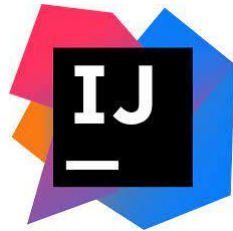


- ☐ **Berorientasi Objek**
- ☐ **Bahasanya Sederhana**
- ☐ **Portable / Multiplatform**
- ☐ **Robustness (error handling)**
- ☐ **Arsitekturnya Kuat**
- ☐ **API dan Dokumentasi yang lengkap (Javadoc)**

**Setelah memahami konsep dasar dari
JAVA , selanjutnya kita persiapkan tools
untuk membuat program dengan JAVA**

Persiapan Tools

1. Spesifikasi minimum dari komputer untuk menjalankan JAVA yaitu **Intel Pentium III 800 MHz** atau **spek yang setara**, **RAM 512 MB**, dan **Free Space 800 MB**.
2. Sistem operasi **windows XP/7/8/10** atau **Ubuntu Linux minimal versi 12.04** atau **Mac OS X minimal versi 10.8.3**
3. **Java SE Development Kit** (JDK dapat diunduh gratis pada website oracle)
4. **IDE atau Text Editor** seperti netbeans, IntelliJ IDEA, Eclipse, Visual Studio Code, Sublime, dan sebagainya.



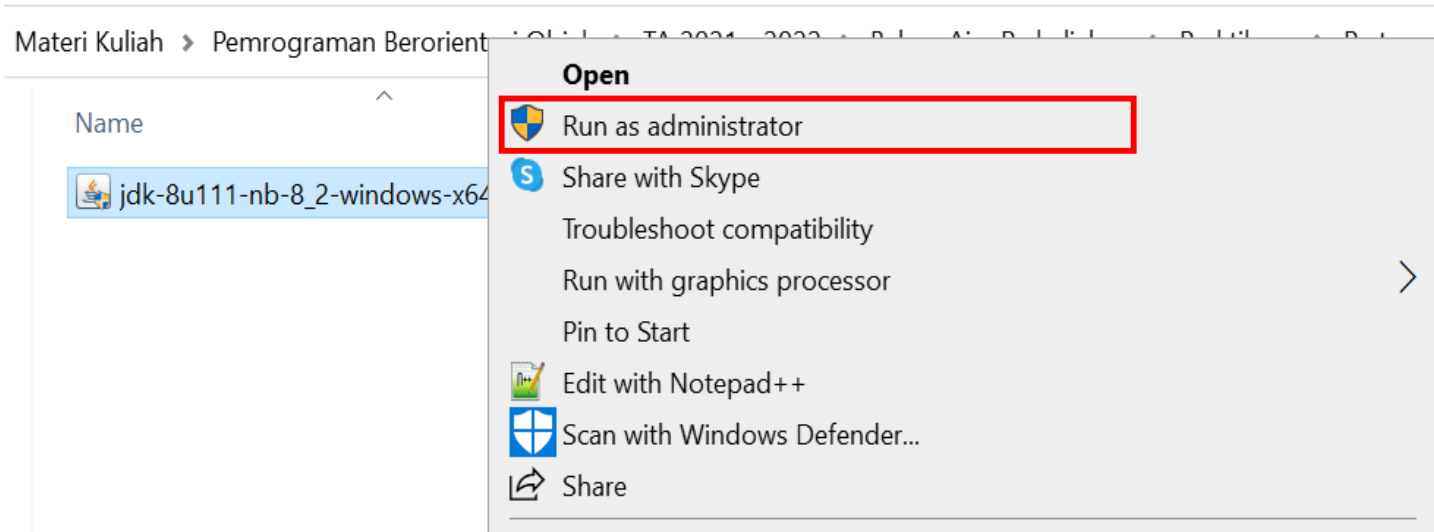
Instalasi JDK dan Netbeans

Files for jdk-8u111-nb-8_2

Name	Last modified	Size
Go to parent directory		
jdk-8u111-nb-8_2-linux-i586.sh	16-Apr-2021 18:38	286.7M
jdk-8u111-nb-8_2-linux-x64.sh	16-Apr-2021 18:41	282.6M
jdk-8u111-nb-8_2-macosx-x64.dmg	16-Apr-2021 18:44	343.0M
jdk-8u111-nb-8_2-windows-i586.exe	16-Apr-2021 18:46	317.2M
jdk-8u111-nb-8_2-windows-x64.exe	16-Apr-2021 18:49	326.0M
jdk-8u111-nb-8_2_archive.torrent	16-Apr-2021 18:50	32.7K
jdk-8u111-nb-8_2_files.xml	16-Apr-2021 18:50	2.6K
jdk-8u111-nb-8_2_meta.sqlite	16-Apr-2021 18:50	20.0K
jdk-8u111-nb-8_2_meta.xml	16-Apr-2021 18:39	1.3K

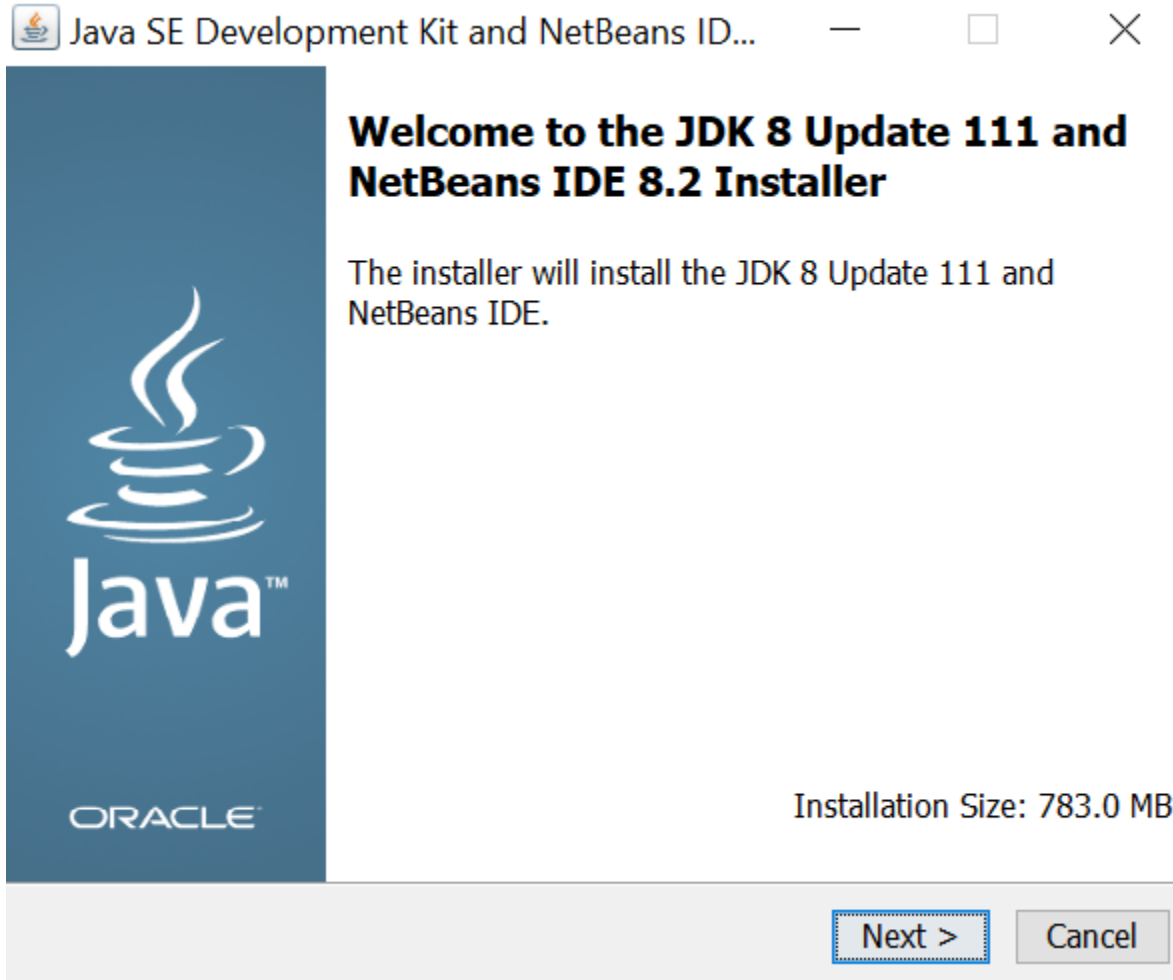
- ❑ **Langkah 1 : Unduh file Java Development Kit (JDK) dan IDE Netbeans.**
- ❑ Untuk kebutuhan praktikum, kita akan menggunakan bundle software **JDK 8u111** dan **Netbeans 8.2** yang disediakan secara gratis oleh oracle . File dapat diunduh gratis pada link berikut (sesuaikan dengan OS Anda) :

Instalasi JDK dan Netbeans (2)



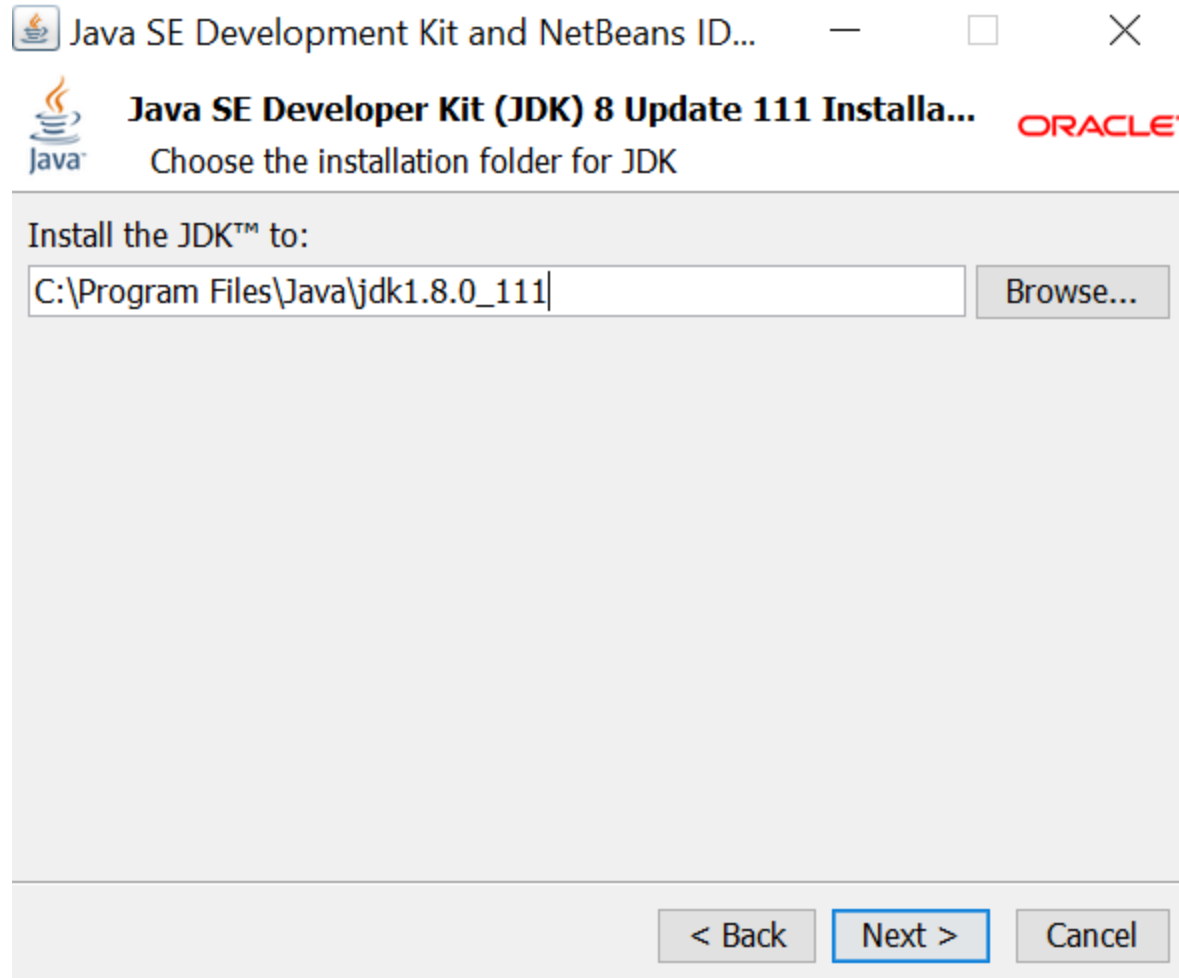
- ❑ **Langkah 2 : Lakukan proses instalasi pada bundle file yang telah Anda download**
- ❑ Untuk melakukan instalasi , Klik kanan dan pilih **“runs as administrator”**

Instalasi JDK dan Netbeans (3)



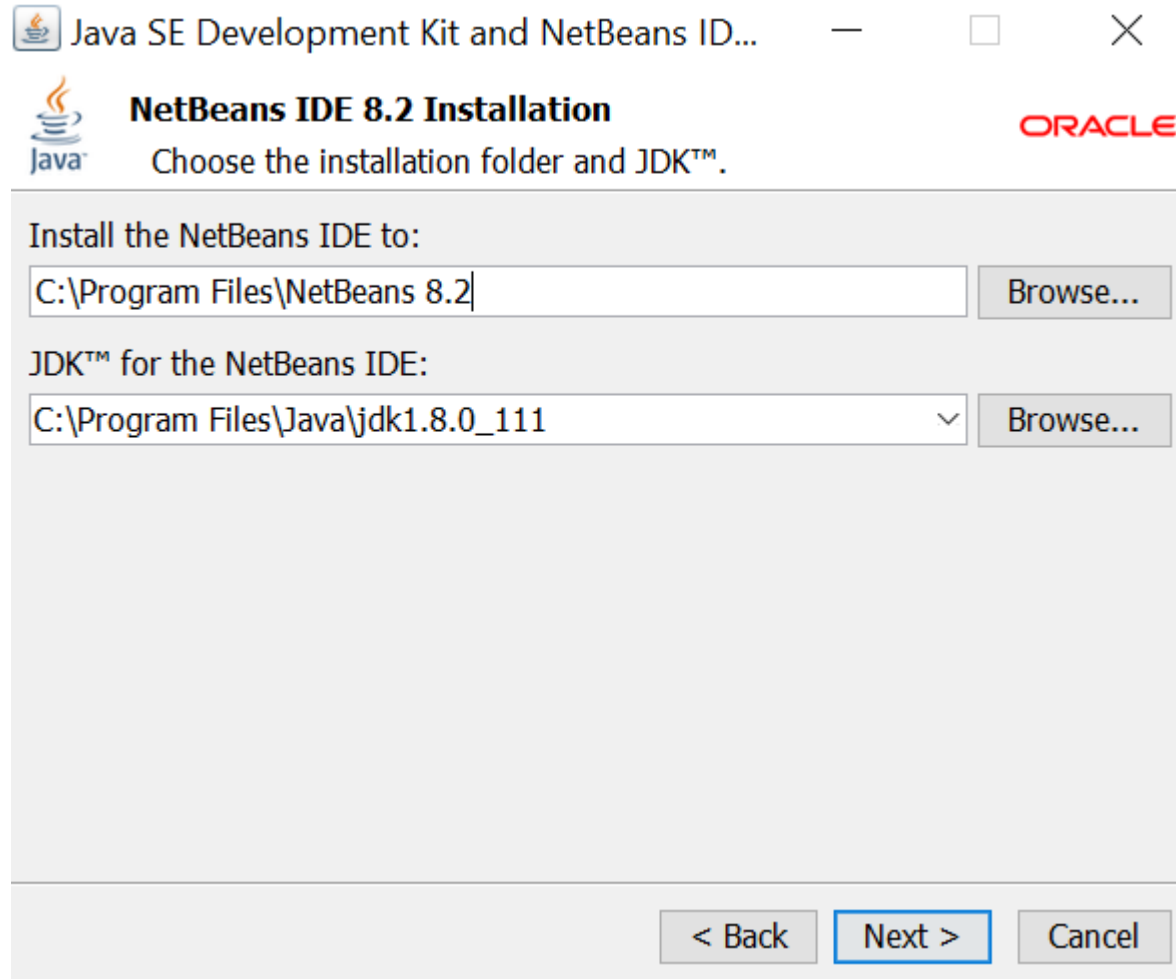
- ❑ **Langkah 3 : Pada jendela installation wizard, klik tombol "next"**

Instalasi JDK dan Netbeans (4)



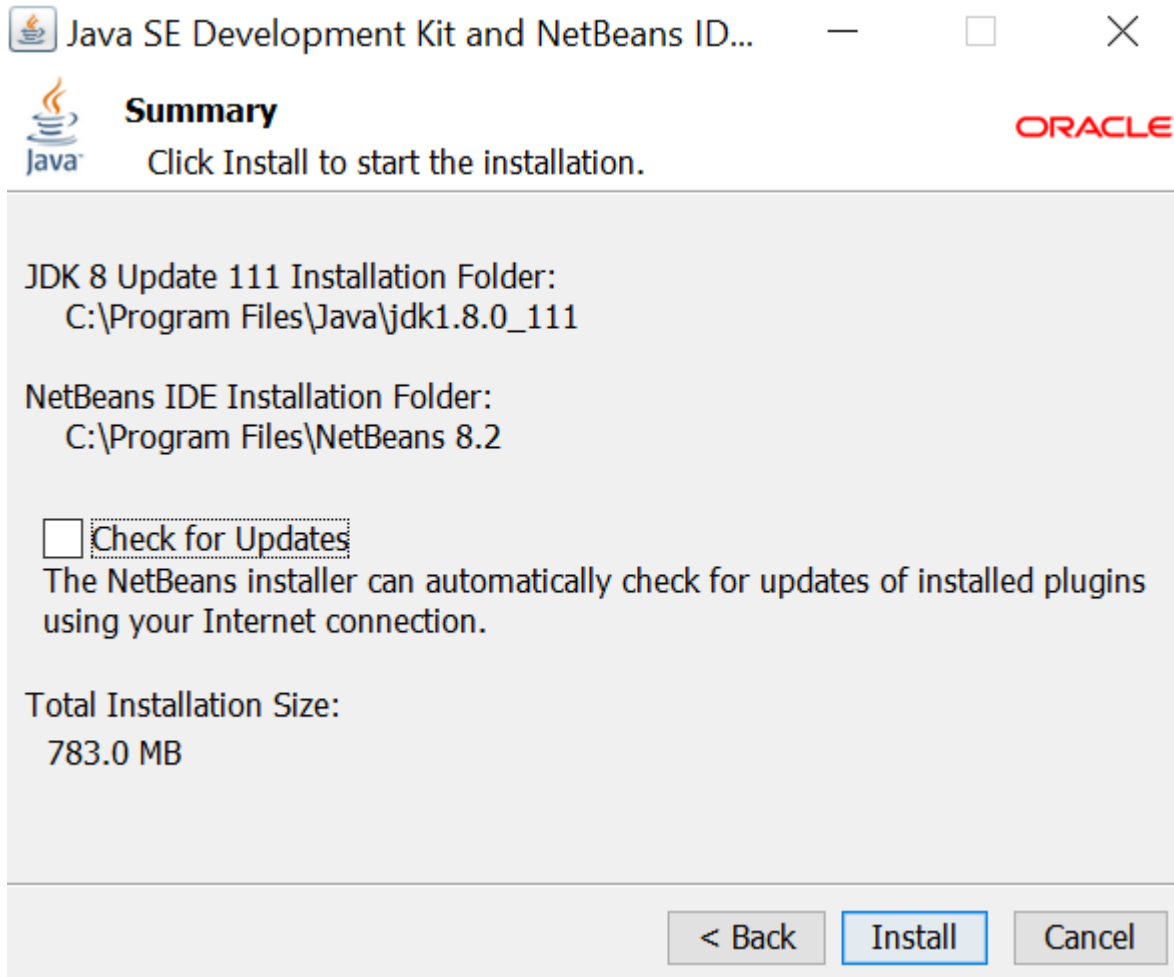
- ❑ **Langkah 4 : Pada jendela instalasi JDK, pilih direktori untuk menginstal JDK lalu klik "next"**

Instalasi JDK dan Netbeans (5)



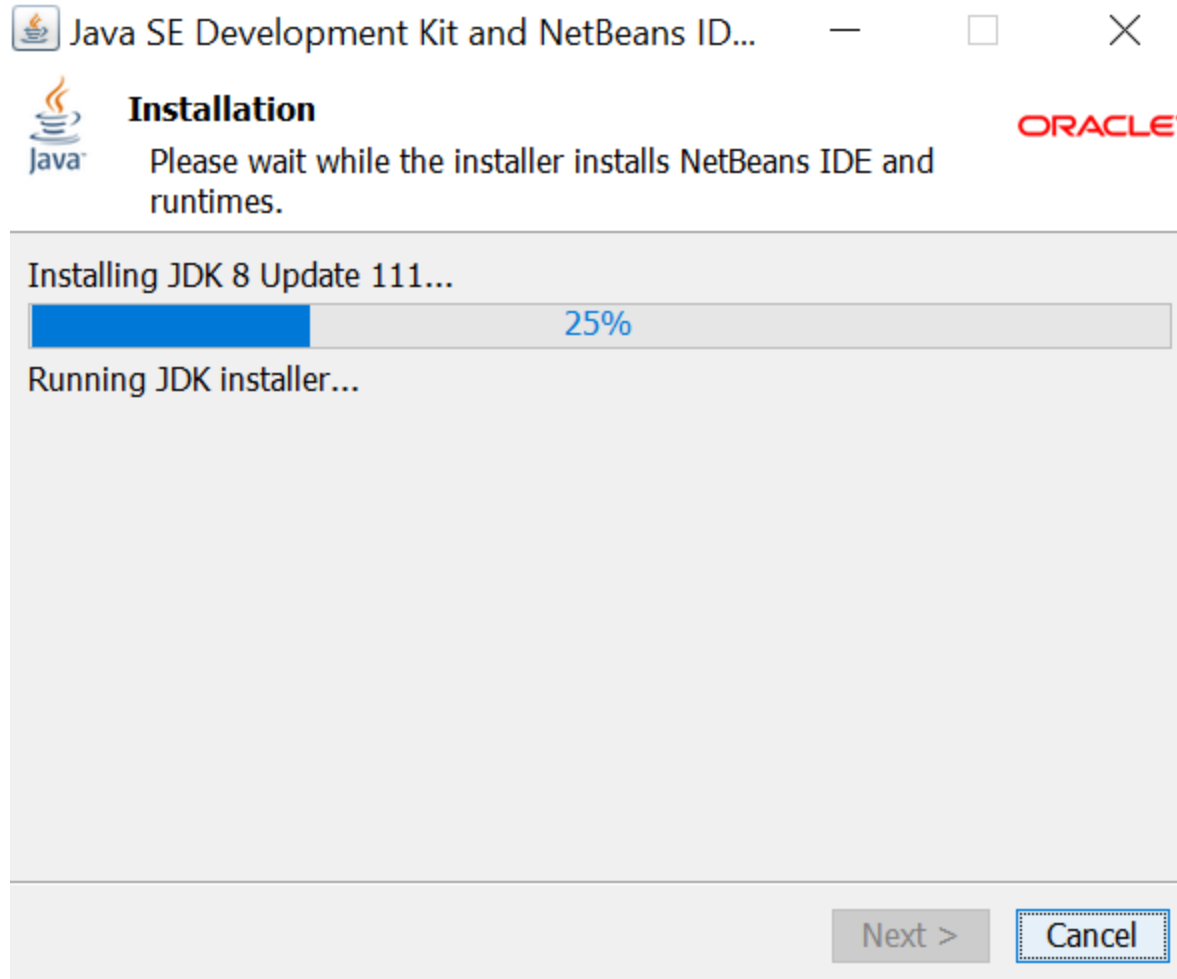
- ❑ **Langkah 5 : Pada jendela instalasi Netbeans, pilih direktori untuk menginstal aplikasi NetBeans.**
- ❑ Selajutnya **pilih direktori** dari JDK yang akan digunakan pada IDE. Jika sudah lalu klik **“next”**

Instalasi JDK dan Netbeans (6)



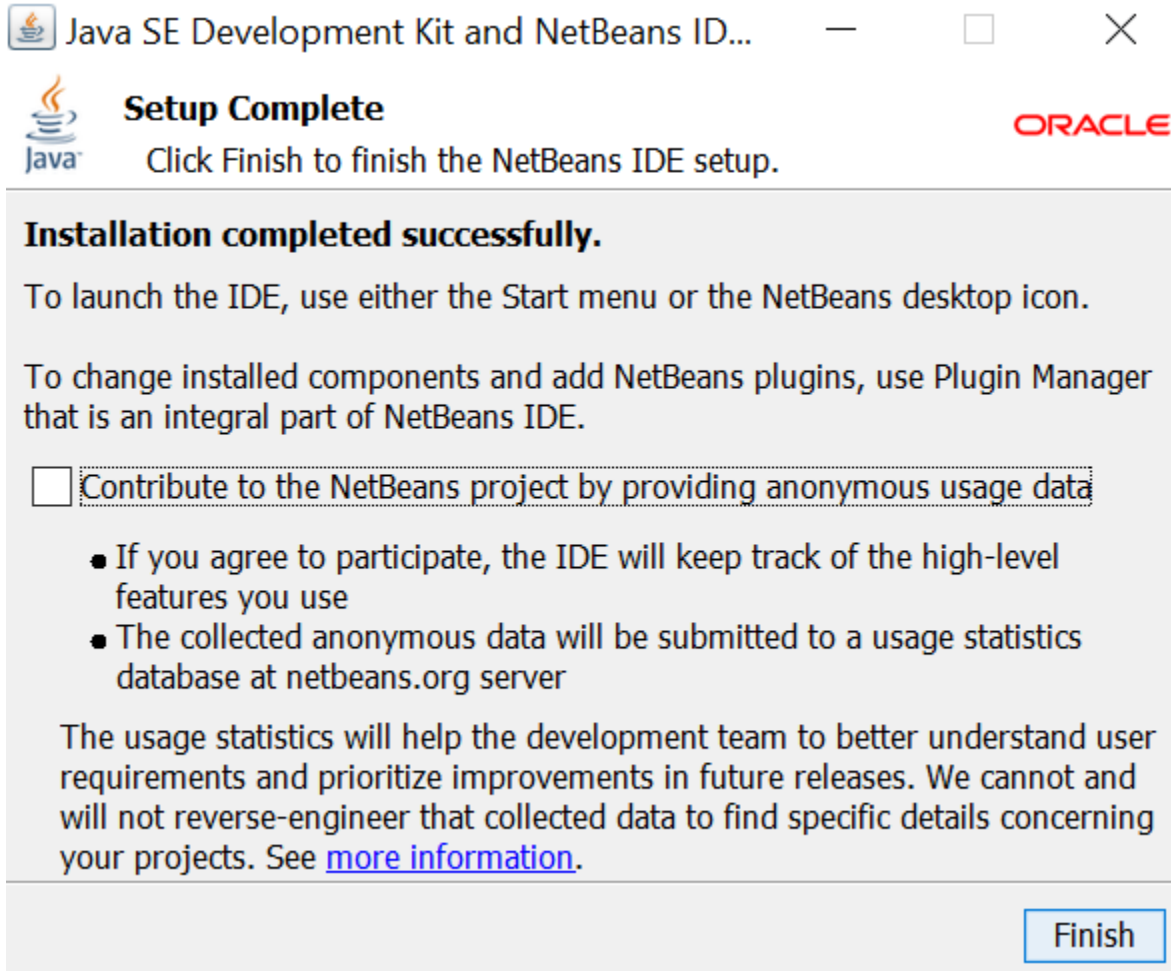
- ❑ **Langkah 6 : Pada jendela summary, pastikan bahwa lokasi dari direktori yang akan digunakan untuk instalasi sudah benar.**
- ❑ **Selanjutnya matikan pilihan “check for Update”**
- ❑ **Klik tombol “Install”, untuk memulai proses instalasi**

Instalasi JDK dan Netbeans (7)



- ❑ **Langkah 7 : Tunggu sampe proses instalasi JDK dan Netbeans selesai.**
- ❑ Jika instalasi sudah selesai , secara otomatis tampilan akan berpindah ke jendela **"setup complete"**

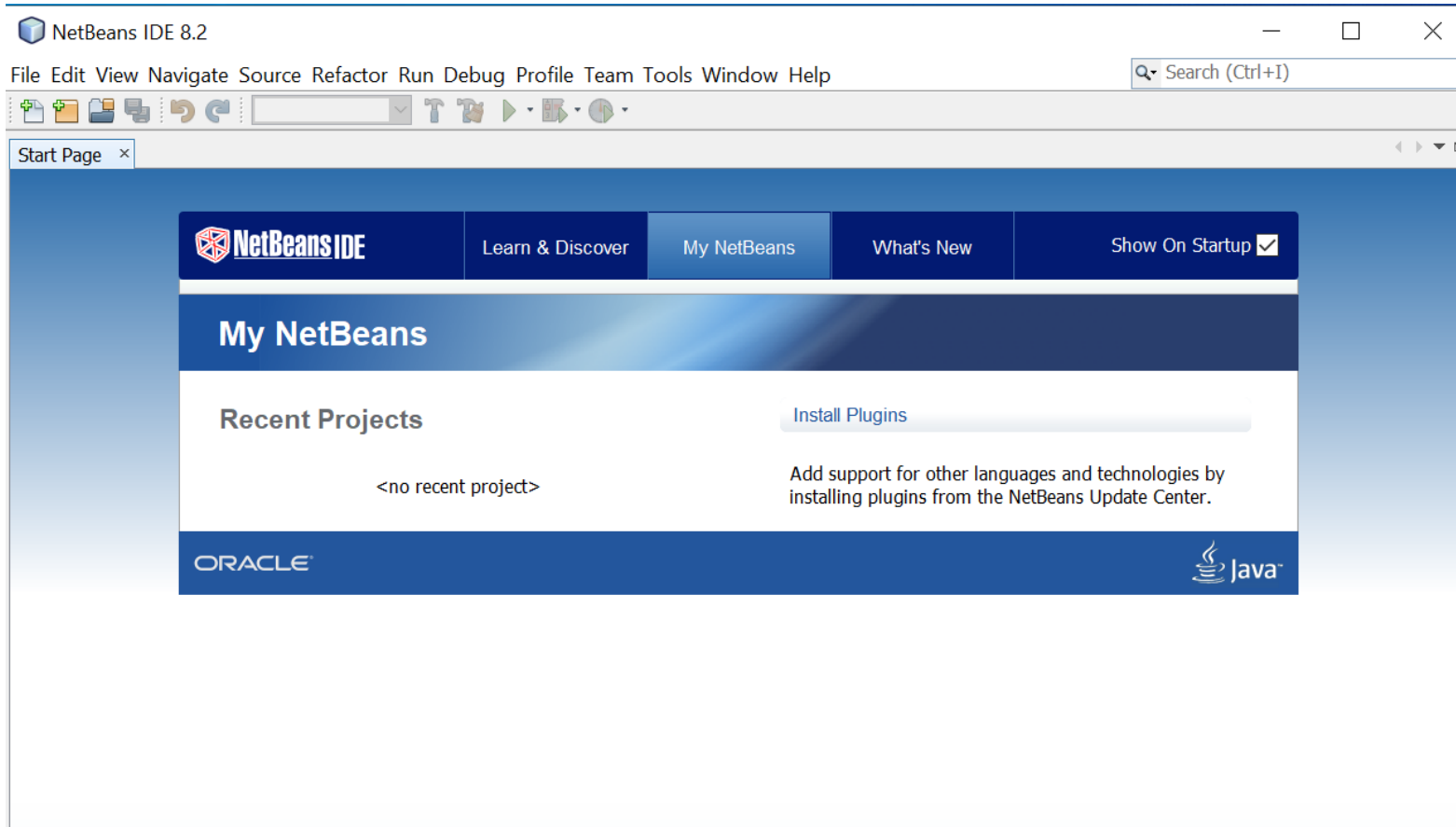
Instalasi JDK dan Netbeans (8)



- ❑ **Langkah 8 : Pada jendela setup complete. Unchecklist pada pilihan contribute to NetBeans... (optional)**
- ❑ **Pilih tombol finish** untuk menyelesaikan proses instalasi JDK dan NetBeans pada komputer Anda.

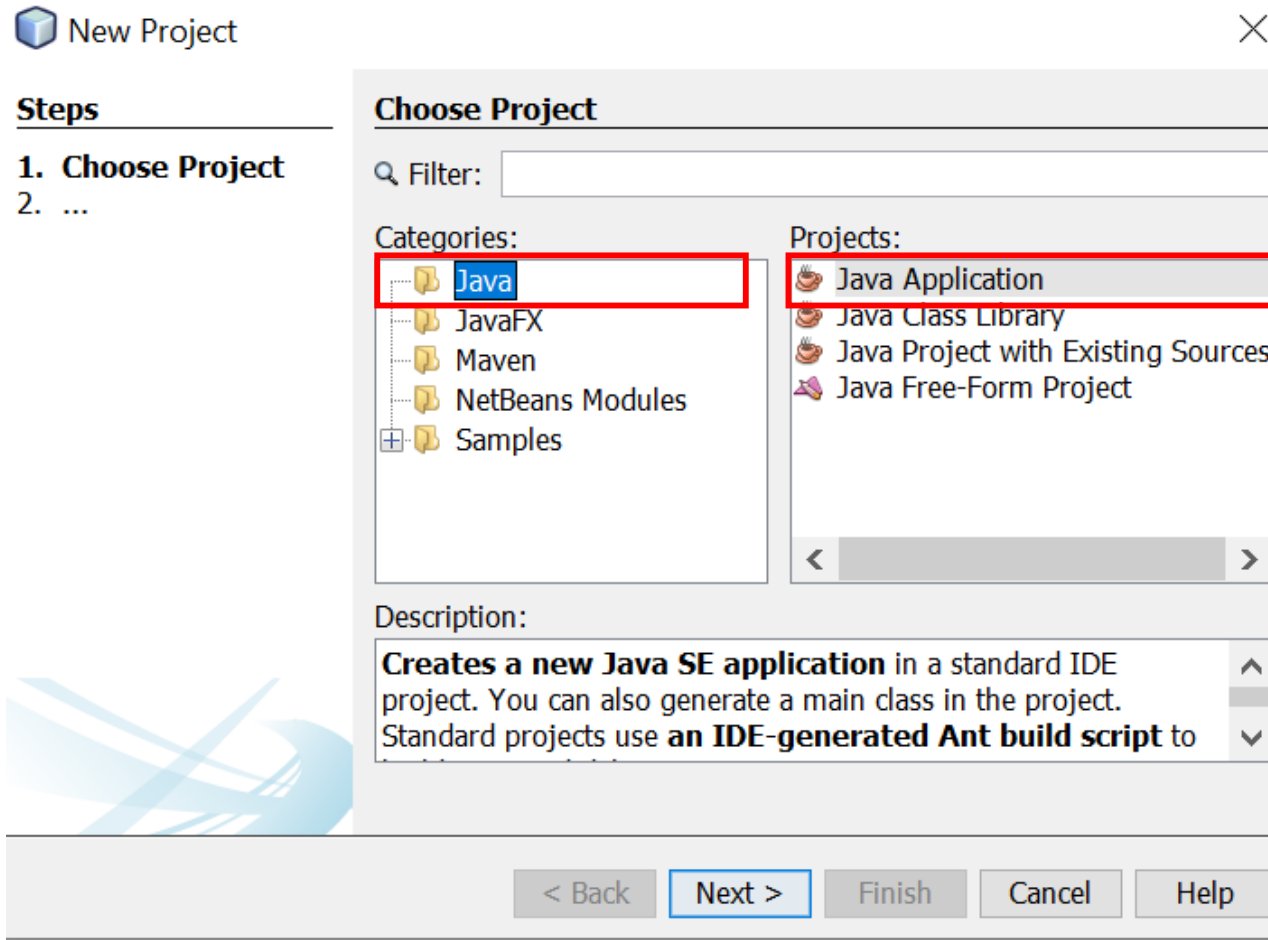
Anda sudah dapat membuat project pertama dengan JAVA

Membuat Project Baru Pada NetBeans



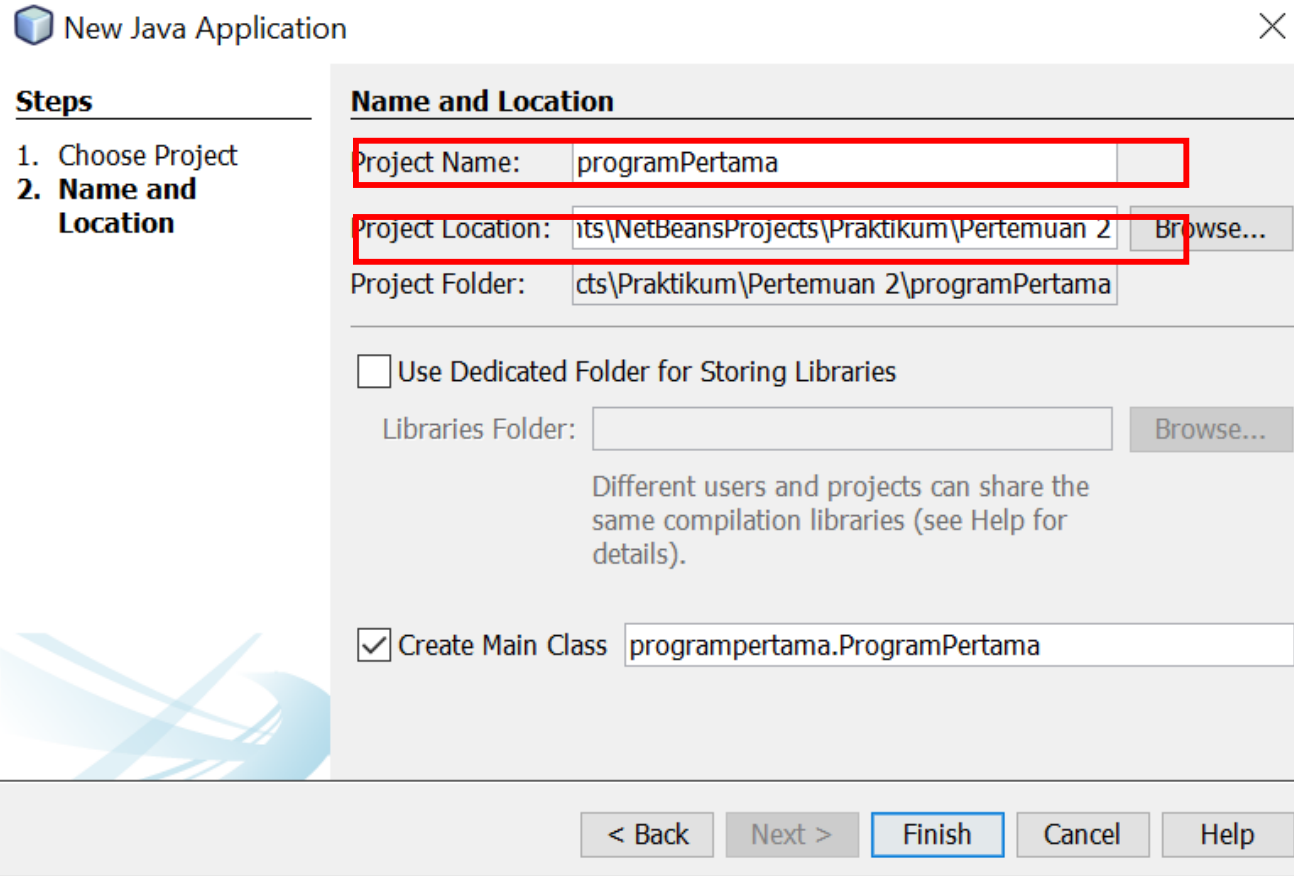
- ❑ **Langkah 1 : Buka aplikasi NetBeans Anda (klik 2 kali pada icon NetBeans)**
- ❑ Tampilan awal dari NetBeans ketika dibuka adalah seperti tampilan berikut.

Membuat Project Baru Pada NetBeans



- ❑ **Langkah 2 : Untuk membuat project baru, klik file -> new project**
- ❑ **Pada jendela new project, pilih **Java** pada “categories” lalu pilih “java Application”.**
- ❑ **Selanjutnya klik tombol “next”.**

Membuat Project Baru Pada NetBeans



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location: Browse...

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

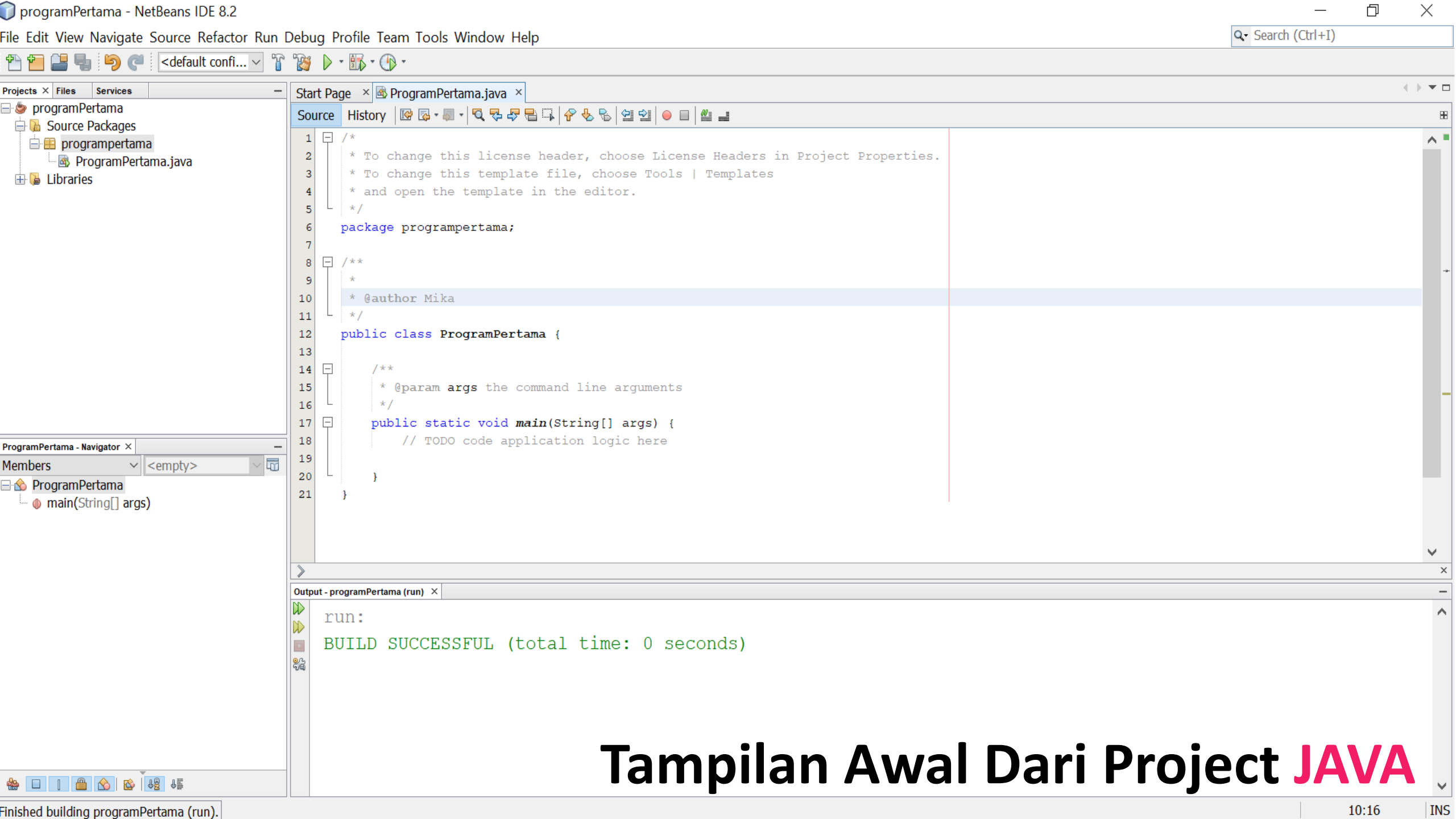
Libraries Folder: Browse...

Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class

< Back Next > **Finish** Cancel Help

- ❑ **Langkah 3** : pada field project name, isilah dengan nama project Anda. Misalkan **"programPertama"**
- ❑ **Tentukan lokasi** dimana project Anda akan disimpan.
- ❑ Selanjutnya klik **tombol "finish"**.



Tampilan Awal Dari Project **JAVA**

Mari mulai untuk menuliskan kode JAVA pertama Anda pada NetBeans

Program Pertama Dengan JAVA

- ❑ Untuk Program Pertama, Mari kita tampilkan kalimat "Hello Dunia" dengan JAVA
- ❑ Untuk mencetak sesuatu ke layar pada JAVA, gunakanlah perintah berikut :

```
System.out.println(" "); atau System.out.print(" ");
```

```
<Hello World/>
```


Program Pertama Dengan JAVA

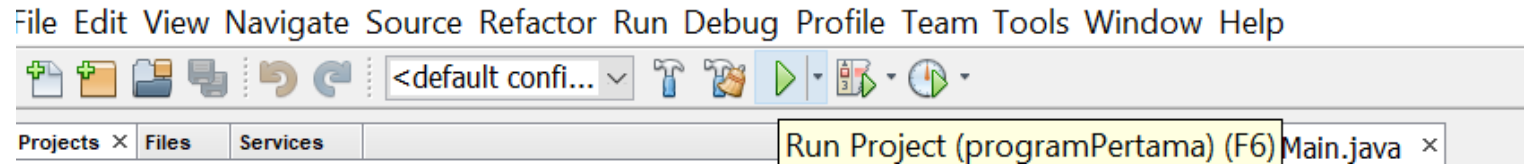
```
Source History [Icons]
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package HaloDunia;
7
8  /**
9   *
10   * @author Mika
11   */
12  public class Main {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          System.out.println("Hello Dunia, Saya Sedang Kuliah PBO");
20      }
21  }
```

*pada JAVA seluruh kode program yang akan dieksekusi dituliskan pada method ini. Untuk pertemuan berikutnya kelas yg didalamnya ada method ini dinamakan kelas tester atau kelas main

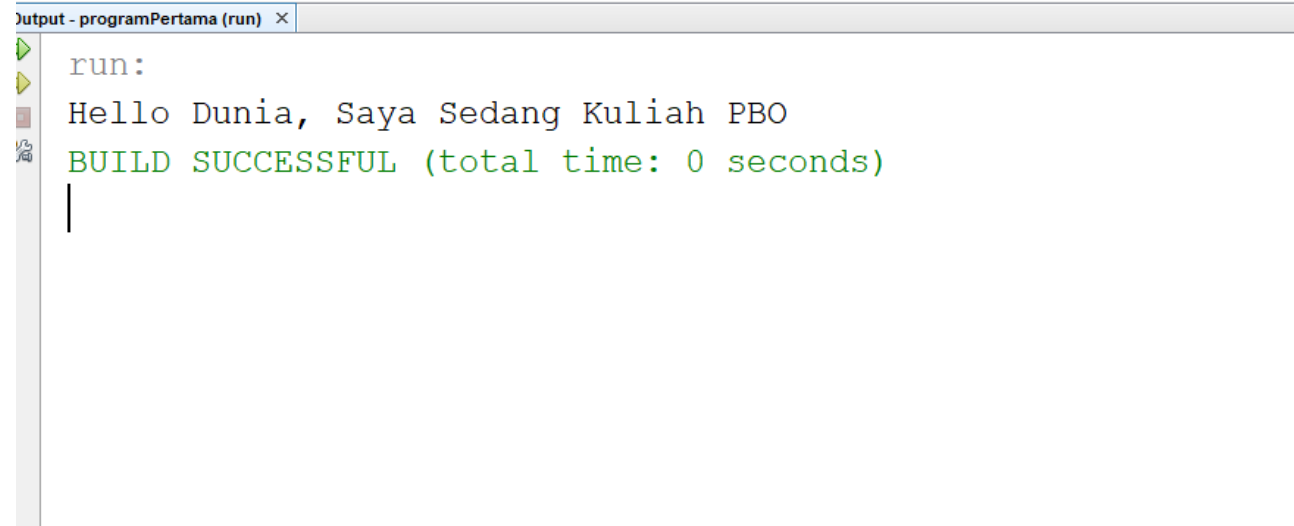
Catatan : hati – hati, Keywords di JAVA sifatnya **Case Sensitif**.

Program Pertama Dengan JAVA

- ❑ Setelah seluruh kode program sudah kita tuliskan, selanjutnya kita RUN untuk melihat apakah kode tersebut dapat dicompile tanpa muncul error.



- ❑ Tampilan program pertama JAVA yang telah berhasil di run



Latihan Praktikum **Pertemuan 1**

Latihan Mandiri

- ❑ **Buatlah sebuah project dengan nama Latihan1 untuk menampilkan Biodata Anda dengan JAVA. Contoh tampilan sebagai berikut (Info diri Anda boleh lebih dari contoh) :**

```
Output - Latihan1 (run) ×
run:
Biodata Diri
-----
NAMA           : Chrismikha Hardyanto
ALAMAT         : Margahayu
TEMPAT LAHIR   : Denpasar
TANGGAL LAHIR  : 1 Januari 1993
-----
HOBI           : Membaca Novel
MAKES          : PIZZA
MIKE           : ES LEMON
BUILD SUCCESSFUL (total time: 0 seconds)
```

Terima Kasih