

## PERTEMUAN 1

### PENGANTAR PEMROGRAMAN BAHASA JAVA

#### A. TUJUAN PEMBELAJARAN

Mahasiswa mampu memahami sejarah bahasa java, memahami Terminologi bahasa java, mengerti JDK, JVM, JRE, dapat membuat program java pertama.

#### B. URAIAN MATERI

##### 1. Pengantar bahasa Java

JAVA dikembangkan oleh Sun Microsystems Inc pada tahun 1991, kemudian Java diakuisisi oleh Oracle Corporation dan dikembangkan oleh James Gosling dan Patrick Naughton. Bahasa java adalah bahasa pemrograman yang sederhana. Menulis, mengkompilasi, dan men-debug program, sangat mudah dan bahasa java membantu untuk membuat program modular dan kode yang dapat digunakan kembali.

##### a. Terminologi Bahasa Java

Sebelum kita mulai belajar Java, mari kita kenali istilah-istilah umum java. *Java Virtual Machine* (JVM) atau Mesin Virtual Java Ini umumnya disebut sebagai JVM. Sebelum kita membahas tentang JVM mari kita lihat tahapan-tahapan eksekusi program. Tahapannya adalah sebagai berikut: kita menulis programnya, kemudian kita compile programnya dan terakhir kita menjalankan programnya.

- 1) Penulisan program java yang dilakukan oleh programmer .
- 2) Kompilasi program dilakukan oleh kompiler javac, javac adalah kompiler java utama yang termasuk dalam java development kit (JDK). Dibutuhkan program java sebagai input dan menghasilkan bytecode java sebagai output.
- 3) Pada fase ketiga, JVM mengeksekusi *bytecode* yang dihasilkan oleh compiler. Ini disebut fase menjalankan program (Run Program).

Sehingga fungsi utama JVM adalah untuk mengeksekusi bytecode yang dihasilkan oleh compiler. Setiap sistem operasi memiliki JVM yang berbeda, namun output yang mereka hasilkan setelah eksekusi *bytecode*

sama di semua sistem operasi. Itu sebabnya kami menyebut java sebagai bahasa platform independen.

**b. Bytecode**

Seperti penjelasan di atas, kompiler javac dari JDK mengkompilasi kode sumber java menjadi *bytecode* sehingga dapat dieksekusi oleh JVM. *Bytecode* disimpan dalam file .class oleh compiler.

**2. Java Development Kit (JDK)**

Saat menjelaskan JVM dan bytecode, istilah umumnya adalah JDK. Mari kita bahas tentang itu. Seperti namanya, ini adalah kit pengembangan java lengkap yang mencakup JRE (Java Runtime Environment), kompiler dan berbagai alat seperti JavaDoc, Java debugger dll.

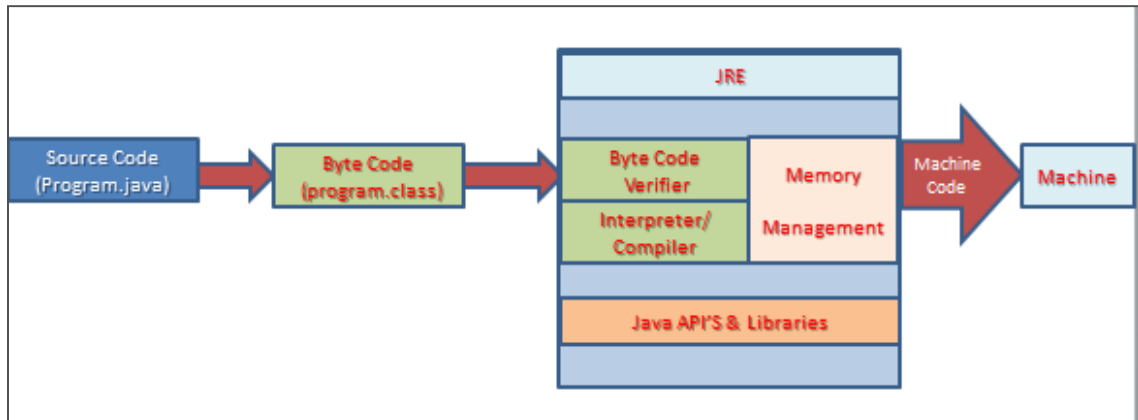
Untuk membuat, mengkompilasi, dan menjalankan program Java, Anda perlu menginstal JDK di komputer Anda.

**3. Java Runtime Environment(JRE)**

Java Run-time Environment (JRE) adalah bagian dari Java Development Kit (JDK). Merupakan distribusi perangkat lunak yang dapat di download JRE secara gratis yang memiliki Java Class Library, alat khusus, dan JVM yang berdiri sendiri.

Ini adalah lingkungan paling umum yang tersedia di perangkat untuk menjalankan program java. Kode sumber Java dikompilasi dan diubah menjadi bytecode Java. Jika Anda ingin menjalankan bytecode ini di platform apa pun, Anda harus download JRE.

JRE memuat kelas, memverifikasi akses ke memori, dan mengambil sumber daya sistem. JRE bertindak sebagai lapisan di atas sistem operasi.



**Gambar 1. 1** Ilustrasi Java Run Time Environment

#### 4. Fitur fitur utama Bahasa JAVA

##### a. Java adalah bahasa platform independen

Compiler (javac) mengonversi kode sumber (file.java) ke *bytecode* (file.class). Seperti disebutkan di atas, JVM mengeksekusi bytecode yang dihasilkan oleh compiler. *Bytecode* /Kode byte ini dapat berjalan di platform apa pun seperti Windows, Linux, Mac OS dll. Yang berarti program yang dikompilasi di windows dapat berjalan di Linux dan sebaliknya. Setiap sistem operasi memiliki JVM yang berbeda, namun output yang mereka hasilkan setelah eksekusi *bytecode* sama di semua sistem operasi. Itu sebabnya java disebut sebagai bahasa platform independen.

##### b. Java adalah bahasa Berorientasi Objek (OOP)

Pemrograman berorientasi objek adalah cara mengatur program sebagai kumpulan objek, yang masing-masing mewakili turunan dari kelas.

4 konsep utama pemrograman Berorientasi Objek adalah:

- 1) Abstraksi
- 2) Enkapsulasi
- 3) Pewarisan
- 4) Polimorfisme

##### c. Simple /Sederhana

Java dianggap sebagai salah satu bahasa yang sederhana karena tidak memiliki fitur yang kompleks seperti Operator overloading, Multiple inheritance, pointer dan alokasi memori Explicit.

d. *Robust* /Bahasa yang Kuat

Kuat berarti dapat diandalkan. Bahasa pemrograman Java dikembangkan dengan cara yang memberikan banyak penekanan pada pemeriksaan awal untuk kemungkinan kesalahan, itu sebabnya kompilernya mampu mendeteksi kesalahan yang tidak mudah dideteksi dalam bahasa pemrograman lain. Fitur utama java yang membuatnya kuat adalah pengumpulan sampah, Penanganan Pengecualian, dan alokasi memori.

e. *Secure* /Aman

Bahasa java tidak memiliki operasi pointer , pada pemrograman dengan mode operasi pointer merupakan hal yang luar biasa untuk optimalisasi dan pembuatan program yang handal serta efisien. Namun operasi pointer dapat menjadi bumerang , karena pointer merupakan sarana luar biasa untuk pengaksesan tak diotorisasi. Dengan peniadaan operasi pointer, Java dapat menjadi bahasa yang lebih aman sehingga tidak dapat mengakses array terikat di java. Itu sebabnya beberapa kelemahan keamanan seperti tumpukan atau buffer overflow tidak mungkin dieksploitas di Java.

f. Bahasa Java terdistribusi

Dengan menggunakan bahasa pemrograman java kita dapat membuat aplikasi terdistribusi. RMI (Remote Method Invocation) dan EJB (Enterprise Java Beans) digunakan untuk membuat aplikasi terdistribusi di java. Dengan kata sederhana: Program java dapat didistribusikan di lebih dari satu sistem yang terhubung satu sama lain menggunakan koneksi internet. Objek pada satu JVM (mesin virtual java) dapat menjalankan prosedur pada JVM jarak jauh.

g. Multithreading

Bahasa Java mendukung multithreading. Multithreading adalah fitur Java yang memungkinkan eksekusi bersamaan dari dua atau lebih bagian program untuk pemanfaatan CPU secara maksimal.

#### h. Portabel

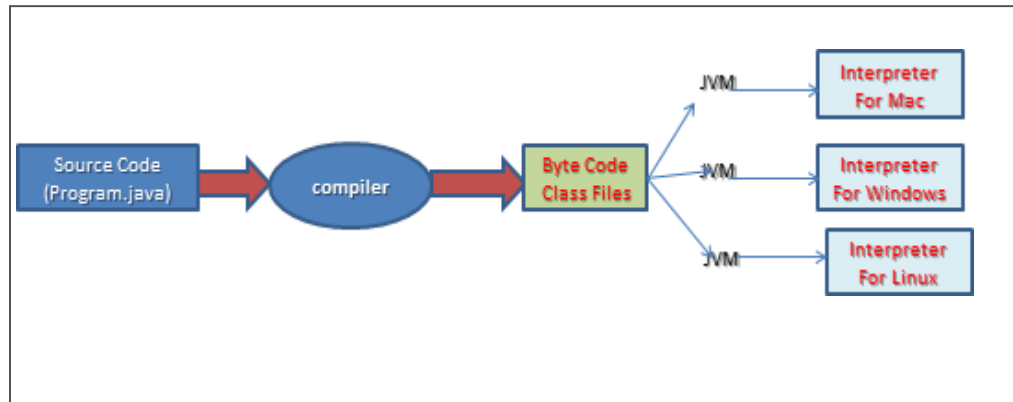
Seperti dibahas di atas, kode java yang ditulis di satu mesin dapat berjalan di mesin lain. Kode byte platform independen dapat dibawa ke platform apa pun untuk dieksekusi yang membuat kode java portabel.

### 5. Compiler dan Interpreter Java

Compiler adalah suatu program yang menerjemahkan bahasa program (source code) kedalam bahasa objek (obyek code) atau istilah pada java disebut *byte code*.

Sedangkan Interpreter adalah Perangkat lunak yang mampu mengeksekusi code program (yang ditulis oleh programmer) lalu menterjemahkannya ke dalam bahasa mesin atau byte code, sehingga mesin melakukan instruksi yang diminta oleh programmer pada java di dieksekusi oleh JVM

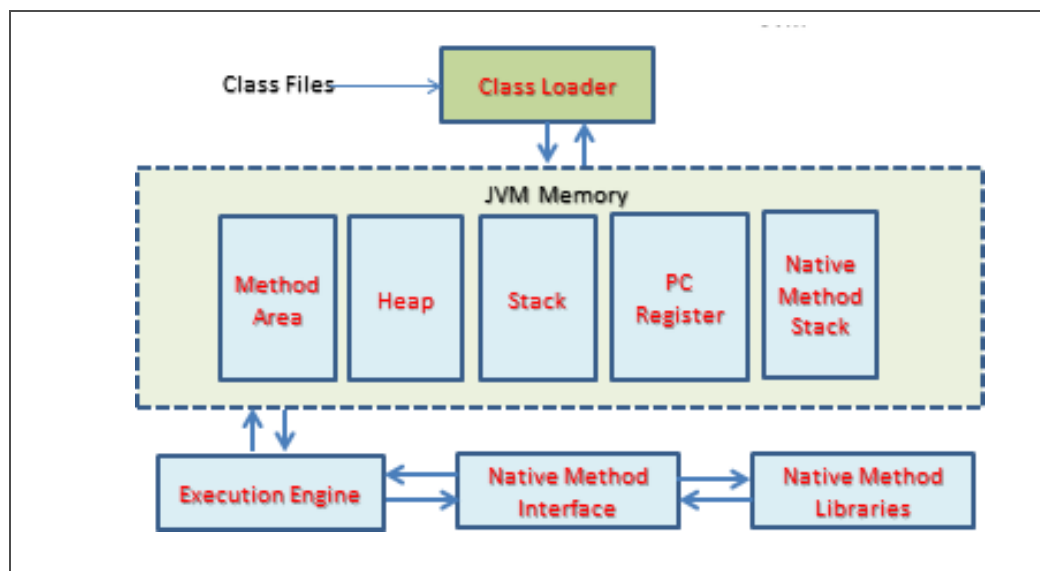
Program yang ditulis dalam bahasa tingkat tinggi (*High Level Language*) tidak dapat dijalankan di mesin mana pun secara langsung. Perlu diterjemahkan terlebih dahulu ke dalam bahasa mesin tertentu. Kompiler javac melakukan hal ini, dibutuhkan program java (file .java yang berisi kode sumber) dan menerjemahkannya ke dalam kode mesin (disebut sebagai byte code atau file .class). Java Virtual Machine (JVM) atau mesin virtual Java mengeksekusi *byte code* tersebut untuk menghasilkan output yang sama di setiap sistem operasi berbeda. Artinya, kode byte yang dihasilkan pada Windows dapat dijalankan di Mac OS dan sebaliknya. Itu sebabnya java sebagai bahasaplatform independen. Hal yang sama dapat dilihat pada gambar di bawah ini:



**Gambar 1. 2** compiler dan interpreter Java

Kesimpulan :

Java Virtual machine (JVM) adalah mesin virtual yang berjalan pada komputer /laptop dan mengeksekusi byte code . JVM tidak memahami kode sumber Java yang ditulis oleh programmer, oleh karena itu kita perlu memiliki Compiler javac yang mengkompilasi file \*.java untuk mendapatkan file \*.class yang berisi Code byte yang dipahami oleh JVM. JVM membuat java portabel (tulis sekali, jalankan di mana saja). Setiap sistem operasi memiliki JVM yang berbeda, namun output yang mereka hasilkan setelah eksekusi byte code sama di semua sistem operasi.



**Gambar 1. 3** Class Loader

Cara kerja JVM:

**Class Loader:** Class loader membaca file .class dan menyimpan kode byte di area metode.

**Area Metode:** Hanya ada satu area metode dalam JVM yang dibagikan di antara semua kelas. Ini menyimpan informasi tingkat kelas dari setiap file .class.

**Heap:** Heap adalah bagian dari memori JVM tempat objek dialokasikan. JVM membuat objek Kelas untuk setiap file .class.

**Stack:** Stack juga merupakan bagian dari memori JVM tetapi tidak seperti Heap, ini digunakan untuk menyimpan variabel sementara.

**Register PC:** Ini melacak instruksi mana yang telah dieksekusi dan mana yang akan dieksekusi. Karena instruksi dieksekusi oleh utas, setiap utas memiliki register PC yang terpisah.

**Native Method stack :** Metode asli dapat mengakses area data runtime dari mesin virtual.

**Native Method interface:** Ini memungkinkan kode java untuk memanggil atau dipanggil oleh aplikasi asli. Aplikasi asli adalah program yang khusus untuk perangkat keras dan OS suatu sistem.

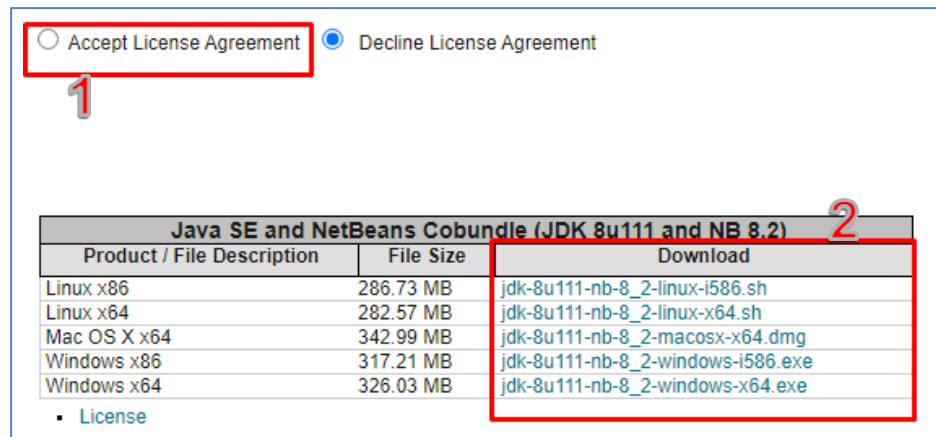
**Garbage collection:** Sebuah instance kelas secara eksplisit dibuat oleh kode java dan setelah digunakan secara otomatis dihancurkan oleh pengumpulan sampah untuk manajemen memori.

## 6. Instalasi JDK dengan Netbeans 8.2

Sebelum kita membuat aplikasi dengan java maka anda harus menginstal software Bahasa pemrograman java. java dapat anda dapatkan dengan cara download di alamat link berikut :

<https://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html> . JDK ini mencakup bundel Java SE dari NetBeans IDE, yang merupakan lingkungan pengembangan terintegrasi untuk mengembangkan aplikasi pada platform Java.

Setelah masuk ke halaman web resminya kita harus memilih *accept License Agreement* [1], Kemudian pilih JDK sesuai dengan sistem operasi yang digunakan [2].



The screenshot shows the NetBeans installation page. At the top, there are two radio buttons: "Accept License Agreement" (selected) and "Decline License Agreement". Below this is a table titled "Java SE and NetBeans Cobundle (JDK 8u111 and NB 8.2)". The table has three columns: "Product / File Description", "File Size", and "Download". The rows list the download links for Linux x86, Linux x64, Mac OS X x64, Windows x86, and Windows x64. A red box highlights the "Accept License Agreement" button and the "Download" column, with a red number "1" next to the button and a red number "2" next to the "Download" column header.

Product / File Description	File Size	Download
Linux x86	286.73 MB	<a href="#">jdk-8u111-nb-8_2-linux-i586.sh</a>
Linux x64	282.57 MB	<a href="#">jdk-8u111-nb-8_2-linux-x64.sh</a>
Mac OS X x64	342.99 MB	<a href="#">jdk-8u111-nb-8_2-macosx-x64.dmg</a>
Windows x86	317.21 MB	<a href="#">jdk-8u111-nb-8_2-windows-i586.exe</a>
Windows x64	326.03 MB	<a href="#">jdk-8u111-nb-8_2-windows-x64.exe</a>

Below the table, there is a link for "License".

**Gambar 1. 4** Instalasi Netbean

Selanjutnya ikut langkah langkah instalasi, hingga selesai.

## 7. Program java pertama

Dalam pertemuan ini, kita akan melihat bagaimana menulis, mengkompilasi dan menjalankan program java dengan Netbeans . kita akan bahas sintaks java, dan beberapa cara untuk menjalankan program java.

```
public class FirstJavaProgram {
    public static void main(String[] args) {
        System.out.println("Ini adalah program java pertama
saya");
    } //End of main
} //End of FirstJavaProgram Class
```

Penjelasan program yang telah kita tulis di atas adalah :

```
public class FirstJavaProgram {
```



Ini adalah baris pertama dari program java . Setiap aplikasi java harus memiliki setidaknya satu definisi kelas yang terdiri dari kata kunci kelas diikuti dengan nama kelas. kata kunci code , itu berarti tidak boleh diubah, kita harus menggunakannya apa adanya. Untuk nama kelas bisa apa saja.

```
public static void main(String[] args) {
```

baris berikutnya dalam program ini seperti diatas adalah

**public:** Ini membuat metode utama menjadi publik yang berarti bahwa kita dapat memanggil metode dari luar kelas.

**static:** Kita tidak perlu membuat objek untuk menjalankan metode statis. Mereka bisa berjalan sendiri.

**void:** Tidak mengembalikan nilai apa pun.

**main:** Ini adalah nama metode. metode titik awal masuk dari mana JVM dapat menjalankan program kita.

**(String[] args):** Digunakan untuk argumen baris perintah yang diteruskan sebagai string. akan dibahas pada pertemuan terpisah.

```
System.out.println("This is my first program in java");
```

Metode ini mencetak konten di dalam tanda kutip ganda ke dalam layar dan menyisipkan baris baru setelahnya.

## 8. Aplikasi yang dijalankan dengan bahasa Java

Aplikasi yang digunakan perangkat di mana Java saat dijalankan di antaranya adalah sebagai berikut:

1. Aplikasi Desktop seperti acrobat reader, media player, antivirus, dll.
2. Aplikasi Web seperti irctc.co.in, javatpoint.com, dll.
3. Aplikasi untuk Perusahaan seperti aplikasi perbankan.
4. Aplikasi Mobile
5. *Embedded System*
6. *Smart Card*
7. Robotika
8. Game , dll.

## 9. Jenis jenis Aplikasi Java

Berdasarkan klasifikasi jenis ada 4 jenis aplikasi yang dapat dibuat menggunakan pemrograman Java:

### a. Aplikasi *Stand Alone*

Aplikasi *Stand Alone* juga dikenal sebagai aplikasi desktop atau aplikasi berbasis *windows*. Ini adalah perangkat lunak tradisional yang perlu kita instal di setiap mesin. Contoh aplikasi standalone adalah Media player, antivirus, dll. AWT dan Swing digunakan di Java untuk membuat aplikasi standalone.

### b. Aplikasi berbasis Web

Aplikasi yang berjalan di sisi server dan membuat halaman dinamis atau disebut aplikasi berbasis web. Saat ini, teknologi Servlet, JSP, Struts, Spring, Hibernate, JSF, dll. digunakan untuk membuat aplikasi berbasis web.

### c. Aplikasi *Enterprise*

Aplikasi yang sifatnya terdistribusi, seperti aplikasi perbankan, dll disebut aplikasi *enterprise*. Ini memiliki keunggulan seperti keamanan tingkat tinggi, penyeimbangan beban, dan pengelompokan. pada Java, EJB digunakan untuk membuat aplikasi *Enterprise*

### d. Aplikasi Mobile

Aplikasi yang dibuat untuk perangkat *Mobile* disebut aplikasi seluler. Saat ini, seperti Android dan Java ME digunakan untuk membuat aplikasi seluler.

## 10. Platform pada Java

Berdasarkan platform ada 4 platform pada Bahasa Java:

### a. Java SE (Java Standard Edition)

Platform ini adalah platform inti pemrograman Java. termasuk pemrograman API Java seperti `java.lang`, `java.io`, `java.net`, `java.util`, `java.sql`, `java.math` dll. termasuk topik inti seperti OOP, *String*, *Regex*, *Exception*, *Class*, *Multithreading*, I/O Stream, Jaringan, AWT, *Swing* dll.

b. Java EE (Edisi Java Enterprise)

Platform Ini adalah platform *enterprise* terutama digunakan untuk mengembangkan aplikasi berbasis web dan Aplikasi *Enterprise* . dibangun di atas platform Java SE. mencakup topik seperti Servlet, JSP, *Web Service*, EJB, JPA, dll.

c. Java ME (Java Micro Edition)

Platform ini adalah platform micro yang didedikasikan untuk aplikasi berbasis mobile seperti android.

d. JavaFX

Platform ini adalah open source untuk aplikasi desktop, mobil , dan *embedded* system yang dibangun di atas Java. platform ini adalah upaya kolaboratif oleh banyak individu dan perusahaan dengan tujuan menghasilkan toolkit yang modern, efisien, dan berfitur lengkap

### C. LATIHAN / TUGAS

1. Jelaskan apa itu Jre ? Apa kegunaannya!
2. Jelaskan apa itu JDK?
3. Jelaskan apa itu Java Virtual Machine?
4. Jelaskan tahap membuat proyek aplikasi baru dan class baru di netbeans!
5. Berdasarkan jenis , aplikasi apa saja yang dapat dibuat dengan menggunakan pemrograman Java?
6. Berdasarkan platform , platform apa saja pada Java?

### D. REFERENSI

Horstmann Cay S., (2011). *Big Java 4<sup>th</sup> Edition* ,san jose university , united state Of America. RRD jefferson city publishing.

Deitel Paul , Deitel Harvey, (2012) Java how to program eighth edition, pearson education, Boston Massachusetts , USA, publishing as prentice hall.

Rose Cristhoper, (2017), Java Succinctly Part 2, Morrisville, NC 27560, USA, Syncfusion, Inc.

Downey Allen B. , Mayfield Chris, (2017), Think Java, Needham, Massachusetts, USA, Green Tea Press

Hayes Helen, (2021), BeginnersBook.com, <https://beginnersbook.com/java-tutorial-for-beginners-with-examples/>, di akses pada tanggal 21 November 2021.

Sonoo Jaiswal , (2021) JavaTpoint offers college campus training , <https://www.javatpoint.com/java-tutorial>, diakses pada tanggal 1 Desember 2021