

# PENGENALAN JAVA

---

## OBJEKTIF :

1. Mahasiswa mampu belajar pengenalan Java seperti mengetahui cara kerja Java dan software yang dibutuhkan.
  2. Mahasiswa mampu menginstall JDK 14 dan IDE IntelliJ versi Community.
- 

## 1.1 SEJARAH JAVA



Pada tahun 1991, Sun Microsystems menawarkan *project* kepada tim yang dipimpin oleh James Gosling, Patrick Naughton, dan Mike Sheridan untuk membuat perangkat konsumen seperti *cable tv box*. Dikarenakan perangkat tersebut tidak mempunyai banyak memori, bahasa yang digunakan harus berukuran kecil dan sesuai dengan arsitekturnya. Sehingga, akhirnya mereka memutuskan untuk membuat bahasa baru. Mereka menamakan *project* tersebut dengan nama "*Green Project*".



Pada tahun 1994, terbentuklah sebuah nama untuk bahasa pemrograman yang mereka buat. Bahasa tersebut bernama OAK. Namun, ternyata nama OAK sudah ada yang menggunakan. Sehingga, nama OAK diubah namanya menjadi Java. Nama Java terinspirasi dari secangkir kopi yang diminum berasal dari Pulau Jawa, Indonesia. Akhirnya dibuatlah bahasa pemrograman Java dengan logo secangkir kopi.



Setelah itu nama Java mulai terdengar, sehingga berbagai industri dan universitas pun banyak yang melirikinya. Java merupakan bahasa pemrograman yang lebih sederhana dan dapat mengakomodasikan hampir seluruh fitur penting bahasa pemrograman lain.

## 1.2 EDISI JAVA

Sebagian besar bahasa pemrograman modern berdiri di atas pustaka-pustaka kelas yang telah ada untuk mendukung fungsionalitas. Pada bahasa pemrograman Java, kelompok-kelompok kelas yang berkaitan erat dimasukkan di satu paket. Paket tersebut bermacam-macam sesuai dengan edisi Java.

Java merupakan bahasa pemrograman yang dapat dijalankan di berbagai *platform*, seperti internet, *consumer electronic products*, dan *computer applications*. Java memiliki beberapa edisi, diantaranya yaitu :

- Java *Standard Edition* (JSE)
- Java *Enterprise Edition* (JEE)
- Java *Micro Edition* (JME)
- Java *Card*

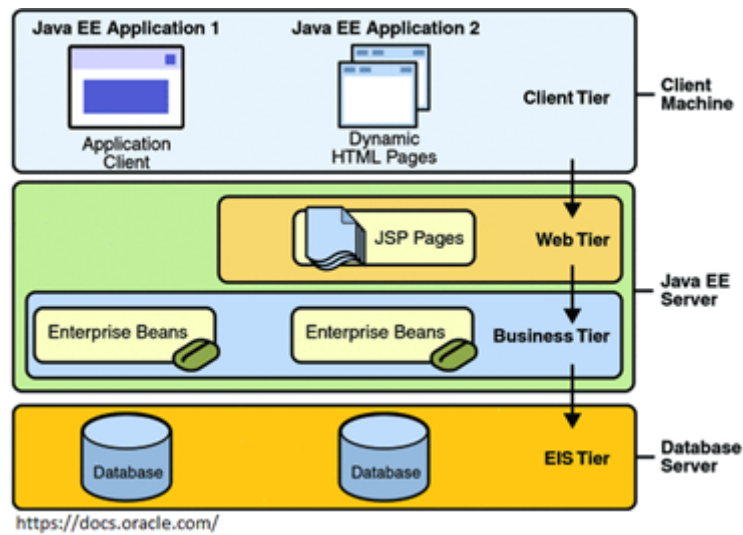
### 1.2.1 JAVA STANDARD EDITION (JSE)

Java *Standard Edition* atau JSE, menyediakan lingkungan pengembangan yang kaya akan fitur, stabil, aman, dan *cross-platform*. JSE termasuk sebagai paket-paket dasar bahasa Java. Edisi ini mendukung konektivitas basis data, rancangan antarmuka pemakai, *input/output*, dan pemrograman jaringan. Berikut *link* untuk *download* JSE <https://www.oracle.com/java/technologies/javase-downloads.html>.



### 1.2.2 JAVA ENTERPRISE EDITION (JEE)

Java *Enterprise Edition* atau JEE, menyediakan *tools* untuk membangun dan menjalankan *multitier applications*. JEE berisi paket-paket yang ada di JSE dan ditambah paket-paket untuk mendukung pengembangan *Enterprise JavaBeans*, *JavaBeans*, *Java Servlets*, *JavaServer Pages*, XML, dan kendali transaksi fleksibel. Berikut *link* untuk *download* JEE <https://www.oracle.com/java/technologies/javasee-8-sdk-downloads.html>.



### 1.2.3 JAVA MICRO EDITION (JME)

Java *Micro Edition* atau JME, digunakan untuk berbagai *consumer electronic product*, seperti *pager*, *smart card*, *cell phone*, *handheld PDA*, dan *set-top box*. JME unggul dalam portabilitas dan menggunakan sekumpulan paket yang lebih kecil. JME berisi *subset* paket JSE ditambah paket spesifik *Micro Edition* `javax.microedition.io`. Aplikasi JME dapat diskala agar dapat bekerja dengan JSE dan JEE. Berikut ini *link* untuk *download* JME <https://www.oracle.com/java/technologies/sdk-downloads.html>.



### 1.2.4 JAVA CARD

Java *Card* merupakan bagian dari platform Java yang ditujukan untuk *smart card*. Untuk keperluan tersebut, Java menyediakan package `javax.smartcardio`. Penggunaan teknologi ini, dapat mengembangkan aplikasi *smart card* untuk berbagai keperluan menggunakan *platform* Java. Java Card menawarkan *platform* eksekusi yang aman dan dapat dioperasikan yang dapat menyimpan dan memperbarui beberapa aplikasi pada satu perangkat dengan sumber daya terbatas, sambil tetap mempertahankan tingkat sertifikasi tertinggi dan kompatibilitas dengan standar. Berikut *link* untuk *download* Java Card Development Kit <https://www.oracle.com/java/technologies/javacard-sdk-downloads.html>.



## 1.3 JDK DAN KONFIGURASI PATH

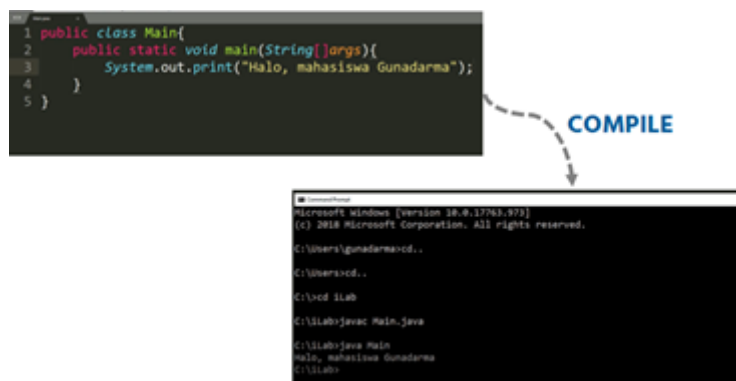
JDK atau *Java Development Kit* digunakan untuk mengembangkan dan menciptakan program Java. Komponen ini memberikan semua *tools*, executables, binaries yang diperlukan untuk *debugging* dan mengeksekusi sebuah program Java. Berikut adalah daftar komponen utama JDK:

- Kompilator (javac)
- *Interpreter* program Java (java)
- *Applet viewer* (appletviewer)
- *Debugger* (jdb)
- *Class file disassembler* (javap)
- *Header and stub file generator* (javah)
- *Documentation generator* (javadoc)
- *Applet demo*
- Kode sumber Java API

## 1.4 CARA KERJA



Pada Java, program yang akan tersimpan akan memiliki nama *file* dengan *extension* .java. *File* .java merupakan *source code* yang dapat dimengerti oleh manusia. Agar program dapat dibaca oleh mesin, *file* .java harus di-*compile* terlebih dahulu. Dari hasil *compile*, maka akan terbentuk *file* baru dengan *extension* .class. *File* tersebut merupakan *byte code* yang dapat dimengerti oleh mesin. Nama *file* pada *source code* akan sama dengan nama *file byte code*, hanya *extension*-nya saja yang berbeda. Sehingga, setelah *compile* dapat dilakukan *run* program dan akan menghasilkan *output* sesuai dengan program tersebut.



Kita dapat melakukan *compile* menggunakan *command line interface* seperti Command Prompt atau CMD. Untuk melakukan *compile* pada CMD, dapat menggunakan perintah `javac nama_file.java`, nama *file* java diikuti dengan *extension* dari *file* java. Setelah itu, gunakan perintah `java nama_file`. Maka akan muncul *output* sesuai dengan program tersebut.

Melakukan *compile* menggunakan CMD tidak efisien jika program yang dimiliki cukup besar dan banyak. Sehingga lebih disarankan menggunakan IDE atau *Integrated Development Environment*.

## 1.5 IDE (INTEGRATED DEVELOPMENT ENVIRONMENT)

IDE atau *Integrated Development Environment*, Aplikasi perangkat lunak yang memberikan fasilitas secara lengkap untuk proses pengembangan *software*. IDE digunakan untuk mengembangkan *game*, *software*, *hardware*, yang menawarkan integrasi mulai dari *debugging*, hingga kompilasi (*compile*).

IDE dibuat oleh beberapa organisasi atau perusahaan besar, seperti Visual Studio dari Microsoft, Xcode dari Apple, dan Android Studio dari Google. Terdapat juga IDE yang dibuat dari organisasi lain seperti IntelliJ dari JetBrains, Eclipse dari Eclipse, dan NetBeans dari Oracle.



## 1.6 KEYWORD dan IDENTIFIER

### 1.6.1 KEYWORD

*Keyword* adalah kata khusus yang sudah disediakan oleh Java untuk tujuan tertentu. *Keyword* hanya dapat digunakan untuk fungsi tertentu dan tidak dapat digunakan sebagai *identifier* nama suatu *variable*, *class*, dan *method*. Berikut ini merupakan daftar *keyword* pada Java :

<b>abstract</b>	<b>continue</b>	<b>for</b>	<b>new</b>	<b>switch</b>
<b>assert***</b>	<b>default</b>	<b>goto*</b>	<b>package</b>	<b>synchronized</b>
<b>boolean</b>	<b>do</b>	<b>if</b>	<b>private</b>	<b>this</b>
<b>break</b>	<b>double</b>	<b>implements</b>	<b>protected</b>	<b>throw</b>
<b>byte</b>	<b>else</b>	<b>import</b>	<b>public</b>	<b>throws</b>
<b>case</b>	<b>enum****</b>	<b>instanceof</b>	<b>return</b>	<b>transient</b>
<b>catch</b>	<b>extends</b>	<b>int</b>	<b>short</b>	<b>try</b>
<b>char</b>	<b>final</b>	<b>interface</b>	<b>static</b>	<b>void</b>
<b>class</b>	<b>finally</b>	<b>long</b>	<b>strictfp**</b>	<b>volatile</b>
<b>const*</b>	<b>float</b>	<b>native</b>	<b>super</b>	<b>while</b>

\* not used  
\*\* added in 1.2  
\*\*\* added in 1.4  
\*\*\*\* added in 5.0

<https://www.javaguides.net/>

### 1.6.2 IDENTIFIER

*Identifier* adalah suatu pengenalan atau pengidentifikasi yang kita deklarasikan agar *compiler* dapat mengenalinya. *Identifier* digunakan untuk penamaan variabel, metode, kelas, paket, dan *interface*. Untuk membuat identifier perlu memperhatikan hal – hal berikut :

- Membedakan huruf besar dan huruf kecil (*case sensitive*)
- Nama *identifier* bisa terdiri dari satu atau beberapa karakter yang terdiri dari angka (0 – 9), huruf (A – Z, a – z), dollar (\$), garis bawah/*underscore* (\_)
- Penamaan *identifier* harus dimulai dengan huruf, *underscore* (\_), atau karakter Dollar (\$). *Identifier* tidak boleh dimulai dengan angka
- Simbol – simbol operator tidak boleh digunakan, seperti ( \* ), ( + ), ( / ), ( - )
- Dalam penamaan *identifier* tidak boleh menggunakan spasi
- Tidak boleh menggunakan kata – kata *keyword* dalam Java, misalnya : `public`, `class`, `static`, dll)
- Panjang nama *identifier* tidak dibatasi

## REFERENSI

- [1] Hariyanto, Bambang. 2017. *Esensi-Esensi Bahasa Pemrograman Java Revisi Kelima*. Bandung: Bandung Informatika.
- [2] docs.oracle.com. "Java Documentation", <https://www.oracle.com/java/technologies/>, diakses pada 17 Juli 2020.
- [3] Kelasterbuka. 2018. "Belajar Java [Dasar] - 01 - Apa itu Java?", Diakses dari <https://www.youtube.com/watch?v=uHyfQV0kbgo&list=PLZS-MHyElRo51w0Hmqi0C8h2KWNzDfo6F> pada 17 Juli 2020.
- [4] Kelasterbuka. 2018. "Belajar Java [Dasar] - 02 - Cara Kerja Java", Diakses dari <https://www.youtube.com/watch?v=OrgFwUl2tzQ&list=PLZS-MHyElRo51w0Hmqi0C8h2KWNzDfo6F&index=2> pada 17 Juli 2020.
- [5] oracle.com. "Java SE Downloads", <https://www.oracle.com/java/technologies/javase-downloads.html/>, diakses pada 17 Juli 2020.