

# PENGENALAN TEKNOLOGI JAVA

Bambang Sumarno HM  
Jurdik Matematika FMIPA UNY  
email: bambang@uny.ac.id

## A. Kilas Balik Sejarah Java

Berbicara mengenai Java dapat dikenakan pada ada dua bagian, yaitu: (a) Java sebagai bahasa pemrograman, dan (b) Java sebagai sebuah platform. Java sebagai bahasa pemrograman dimulai ketika James Gosling dkk (tahun 1990), pemrogram yang bekerja pada Sun Microsystems mendapat tugas menciptakan program untuk mengendalikan perangkat elektronika rumah tangga, yang dikenal dengan nama proyek 'Green'.

Gosling (dibantu oleh rekan-rekannya Patrick Naughton, Chris Warth, Ed Frank, dan Mike Sheridan) mulai mendesain program tersebut dengan bahasa pemrograman C++, karena adanya sifat berorientasi obyek. Saat menggunakan C++, Gosling mulai merasakan bahwa bahasa tersebut tidak cocok untuk tugas yang dibebankan kepada timnya. Beberapa permasalahan yang muncul, seperti: pewarisan majemuk (*multiple inheritance*) dan bug program berupa kebocoran memori (*memory leaks*). Gosling segera memutuskan untuk merancang sendiri bahasa pemrograman komputer yang sederhana dan mampu menghindarkannya dari masalah-masalah tersebut dengan tetap memanfaatkan sintaks dasar dan sifat berorientasi objek dari C++.

Akhirnya lahir Java (awalnya diberi nama Oak) sebagai sebuah bahasa pemrograman berbasis *object-oriented* baru yang portable dan dapat diimplementasikan pada beragam peralatan dengan kapasitas memori terbatas. Oleh karenanya, Java dibekali dengan kemampuan untuk dapat dipergunakan pada beragam platform (portable). Hal ini menjadikan aplikasi yang ditulis dengan Java dapat langsung dijalankan pada platform yang mendukung Java tanpa perlu mengkompilasi ulang kode sumber Java tersebut. Itulah mengapa dalam pemrograman Java dikenal istilah "*write once, run everywhere*", sekali menuliskan kode program, dapat dijalankan di bawah platform manapun.

Java menurut definisi dari Sun Microsystem adalah sekumpulan teknologi untuk membuat perangkat lunak dan menjalankannya pada komputer standalone ataupun pada lingkungan jaringan. Java sebagai sebuah platform, terdiri atas dua bagian utama, yaitu: (a) **Java Virtual Machine** (JVM), dan **Java Application Programming Interface** (Java API). Pada dasarnya, terdapat berbagai jenis platform (sistem operasi) tempat aplikasi/perangkat lunak berjalan, seperti: Macintosh, Microsoft Windows, Unix, Netware, Linux, dan OS/2. Seringnya, aplikasi/perang lunak yang berjalan pada suatu platform (misalnya Windows) tidak dapat dijalankan di platform yang lain (misalnya Linux) tanpa harus kompilasi ulang atau pengubahan kode program. Java platform merupakan perangkat lunak yang menjadi mesin virtual bagi aplikasi Java untuk dieksekusi. Java berdiri di atas sebuah mesin interpreter yang diberi nama JVM. JVM inilah yang akan membaca file bytecode (ekstensi **.class**) dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Jadi aplikasi Java tidak perlu dikompilasi ulang jika telah dikompilasi di suatu platform dan akan dijalankan di platform berbeda dengan saat dikompilasi, karena aplikasi Java dijalankan di atas JVM.

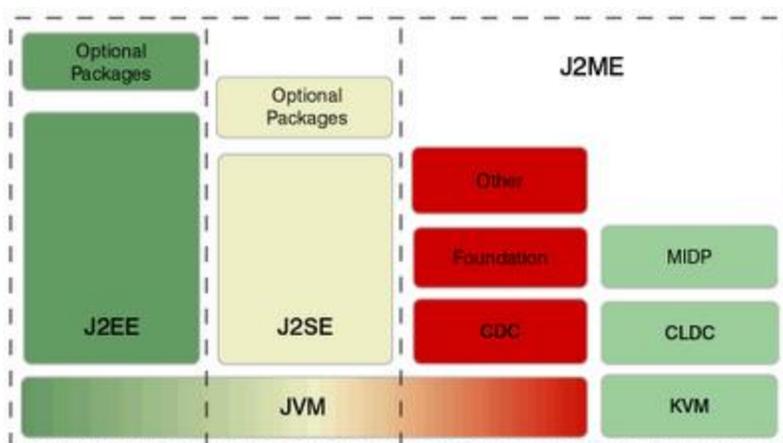
Platform Java terdiri dari sekumpulan library, compiler, debugger dan alat lain yang dipaket dalam *Java Development Kit* (JDK). Java 2 adalah generasi yang sekarang sedang berkembang dari platform Java. Agar sebuah program Java dapat dijalankan, maka file sumber (ekstensi **.java**) harus dikompilasi menjadi file bytecode. Untuk menjalankan file byte-code tersebut dibutuhkan *Java Runtime Environment* (JRE) yang memungkinkan pemakai untuk menjalankan program Java, hanya menjalankan tidak membuat kode baru lagi. JRE terdiri dari JVM dan library Java yang digunakan.

Pada saat ini, Java Platform telah ada secara default pada berbagai sistem operasi, antara lain:

- Windows 9.X/NT/2000
- Sun Solaris
- MacOS
- Novell Netware 4.0
- HPUx
- Hitachi OS
- AIX
- Irix
- Unixware (SCO)

Java dan class pendukungnya didistribusikan dalam tiga bagian yang masing-masing memiliki konsentrasi tersendiri, yaitu:

<b>Java 2 Enterprise Edition (J2EE)</b>	untuk aplikasi enterprise dengan focus pada pengembangan sisi webserver, termasuk seriet, JSP, EJB, XML
<b>Java 2 Standard Edition (J2SE)</b>	untuk aplikasi desktop
<b>Java 2 Micro Edition (J2ME)</b>	untuk piranti dengan sumberdaya terbatas (perangkat mobile)



Gambar 1. Konstruksi edisi Java (J2EE, J2SE, dan J2ME)

- **J2EE (Java 2 Enterprise Edition)**

J2EE adalah kelompok dari beberapa API (*Application Programming Interface*) dari Java dan teknologi selain Java. J2EE dirancang untuk membuat aplikasi yang rumit, dan sering dipandang sebagai *middle-ware* atau teknologi yang berjalan di server. Pada kenyataannya J2EE tidak hanya terbatas untuk itu, bahkan juga mencakup teknologi yang dapat digunakan di semua lapisan dari sebuah sistem informasi. Implementasi J2EE menyediakan kelas dasar dan API dari Java yang mendukung pengembangan dan rutin standard untuk aplikasi client maupun server, termasuk aplikasi yang berjalan di web browser.

- **J2SE (Java 2 Second Edition)**

J2SE adalah inti/ dasar dari bahasa pemrograman Java. JDK, tool open source dari Sun, adalah salah satu tool dari J2SE untuk mengkompilasi dan menjalankan program Java. Tool J2SE yang salah satunya adalah JDK 1.7 dapat diunduh pada <http://java.sun.com/>.

- **J2ME (Java 2 Micro Edition)**

J2ME adalah lingkungan pengembangan yang dirancang untuk meletakkan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya dengan sumberdaya terbatas. Teknologi J2ME memiliki beberapa keterbatasan. Jika diaplikasikan pada ponsel, J2ME sangat tergantung pada perangkat (device) yang digunakan, dapat dari segi merk ponsel atau dukungan terhadap teknologi J2ME. Jadi sangat mungkin terjadi dapat berjalan dengan baik pada suatu perangkat, tetapi tidak dapat berfungsi baik pada perangkat lainnya. Keterbatasan lainnya adalah pada ukuran aplikasi, karena memori pada ponsel sangat terbatas. Bahkan sebagian ponsel tidak mengizinkan aplikasi J2ME menulis pada file, karena alasan keamanan.

J2ME membawa Java ke dunia informasi, komunikasi, dan perangkat komputasi (selain perangkat komputer desktop) yang biasanya dengan sumberdaya lebih kecil. J2ME biasa digunakan pada telepon selular, personal digital assistants (PDA), tablet, dan sejenisnya.

J2ME adalah bagian dari J2SE, sehingga tidak semua library yang ada pada J2SE dapat digunakan pada J2ME; Tetapi J2ME mempunyai beberapa library khusus yang tidak dimiliki J2SE.

## **B. Teknologi Java**

Java API menyediakan beberapa fitur yang menarik untuk pengembangan aplikasi, antara lain:

- **Applet**

Java Applet merupakan program Java yang berjalan di atas browser. Penggunaan Java Applet menjadikan halaman HTML akan lebih dinamis dan menarik. Fitur ini sangat cocok untuk pengembangan aplikasi-aplikasi berbasis web.

- **Java Networking**

Java Networking merupakan sekumpulan API yang menjadikan fungsi-fungsi untuk aplikasi-aplikasi jaringan. Java Networking menyediakan akses untuk TCP, UDP, IP Address dan URL, tetapi tidak untuk ICMP dikarenakan alasan keamanan.

- **Java Database Connectivity (JDBC)**

JDBC menyediakan sekumpulan API yang dapat digunakan untuk mengakses basisdata, seperti: Oracle, MySQL, PostgreSQL, atau Microsoft SQL Server.

- **Java Security**  
Java Security menyediakan sekumpulan API yang dapat digunakan untuk mengatur keamanan dari aplikasi Java baik secara *high level* maupun *low level*, seperti: *public/private key management and certificates*.
- **Java Swing**  
Java Swing menyediakan sekumpulan API yang dapat digunakan untuk membangun aplikasi GUI (*Graphical User Interface*) dengan model yang lebih beragam.
- **Java RMI**  
Java RMI menyediakan sekumpulan API yang dapat digunakan untuk membangun aplikasi Java yang mirip dengan model RPC (*Remote Procedure Call*), sehingga obyek Java dapat dipanggil (call) secara remote pada jaringan komputer.
- **Java 2D/ 3D**  
Java 2D/ 3D menyediakan sekumpulan API yang dapat digunakan untuk membangun grafik 2D/ 3D yang menarik dan juga akses ke printer.
- **Java Server Pages (JSP)**  
Berkembang dari Java Servlet yang digunakan untuk menggantikan aplikasi-aplikasi CGI. JSP yang mirip ASP dan PHP merupakan alternatif terbaik untuk solusi aplikasi internet.
- **Java Native Interface (JNI)**  
JNI menyediakan sekumpulan API yang dapat digunakan untuk mengakses fungsi-fungsi pada library (\*.dll atau \*.so) yang dibuat dengan bahasa pemrograman yang lain, seperti: C++.
- **Java Sound**  
Java Sound menyediakan sekumpulan API yang dapat digunakan untuk memanipulasi sound.
- **Java (Interface Definition Language) IDL + (Common Object Request Broker) COBRA**  
Java IDL menyediakan dukungan Java untuk implementasi COBRA yang merupakan model objek terdistribusi untuk solusi aplikasi besar di dunia jaringan.
- **Java Card**  
Java Card utamanya digunakan untuk aplikasi-aplikasi pada smart card, yang sederhana wujudnya seperti SIM card pada handphone.
- **Java Telephony Application Programming Interface (JTAPI)**  
JTAPI menyediakan sekumpulan API yang dapat digunakan untuk memanfaatkan alat-alat telepon, sehingga akan cocok untuk aplikasi-aplikasi CTI (*Computer Telephony Integration*) yang dibutuhkan seperti ACD (*Automatic Call Distribution*), PC-PBX dan lainnya.

### C. Keunggulan Java

Java merupakan teknologi yang unik dan revolusioner dengan semboyan "*write once run anywhere*" sehingga program Java dapat dijalankan di berbagai platform sistem operasi seperti Linux, Windows maupun Unix. Berikut ini adalah beberapa kelebihan pemrograman Java dibandingkan dengan bahasa pemrograman lainnya.

- **Bersifat portable dan platform independent**  
Program Java dapat dijalankan di berbagai platform asalkan JVM untuk platform tersebut tersedia.

- **Pembuang Sampah (Garbage Collection)**

Program Java membuang sendiri “sampah-sampah” yang tak berguna, artinya program tidak perlu menghapus obyek yang dialokasikannya di memori karena dealokasi memori dilakukan secara otomatis. Ini menjadikan program Java bebas dari permasalahan pengelolaan memori.

- **Bebas Arsitektur (Architecture Neutral)**

Pada dasarnya, program Java dirancang untuk bekerja pada berbagai arsitektur prosesor dan berbagai sistem operasi. Oleh karena itu, program java bekerja dengan interpreter Java untuk mengubah program Java menjadi byte-code sebelum dijalankan oleh suatu mesin tertentu.

- **Tangguh (Robust)**

Berangkat dari hasil interpreter Java memeriksa seluruh akses sistem yang dilakukan program, maka program Java tidak akan membuat sistem menjadi crash. Bila terjadi masalah serius, program Java membuat pengecualian (*exception*) yang dapat ditangani dan dikelola oleh program tanpa beresiko memacetkan sistem.

- **Aman (Secure)**

Sistem Java sangat handal dalam mengelola memori, tidak hanya memverifikasi seluruh akses ke memori, tetapi juga menjamin tidak ada virus yang “membonceng” pada program yang sedang berjalan. Hal ini juga dikarenakan Java tidak mendukung pointer, sehingga program tidak mendapat akses ke area sistem dimana ia tidak mendapat otorisasi.

- **Dapat Diperluas**

Program Java mendukung metode *native code*, yaitu memungkinkan pemrogram menulis fungsi dalam bahasa lain, biasanya C++ yang dapat dieksekusi secara lebih cepat karena langsung berjalan di atas perangkat keras yang bersangkutan, dari pada yang ditulis dalam Java yang berjalan di atas JVM. Metode native code dikaitkan secara dinamis (*dynamically linked*) ke program Java, yakni dikaitkan dengan program saat runtime.

## D. Karakteristik Java

Java dikenal sebagai bahasa pemrograman tingkat tinggi dengan karakteristik/sifat-sifat berikut:

- **Sederhana (Simple)**

Java dirancang untuk mudah dipelajari, khususnya bagi pemrogram C++ akan mudah sekali untuk berpindah ke Java. Selain itu pengembang Java banyak membuang fitur-fitur yang tidak diperlukan seperti pointer matematis, casting tipe implicit, struktur atau union, operator overloading, template, header filer, maupun pewarisan majemuk (multiple inheritance).

- **Berorientasi Objek (Object Oriented)**

Java menggunakan kelas-kelas untuk mengelola kode ke dalam modul-modul logika. Pada saat runtime, program menciptakan objek dari kelas-kelas yang ada. Kelas-kelas Java dapat merupakan warisan dari kelas lain, tetapi pewarisan majemuk tidak diperkenankan.

- **Bertipe Statik**

Semua objek yang digunakan dalam program harus dideklarasikan sebelum digunakan. Cara ini memungkinkan kompilator Java melaporkan dan melokalisir terjadinya konflik-konflik tipe.

- **Terkompilasi**

Sebelum program Java dapat berjalan, maka harus dikompilasi oleh kompilator Java (*javac*). Hasil kompilasinya berupa file kode byte (bytecode), mirip dengan kode mesin, yang dapat dieksekusi di bawah sistem operasi apapun asal memiliki interpreter Java (JVM). Jadi dapat dikatakan program Java merupakan bahasa yang dikompilasi sekaligus diinterpretasi.

- **Multi-threaded**

Program Java dapat berisi eksekusi *multiple threads*, yang memungkinkan program mengulas citra di layar pada satu thread, sementara tetap dapat menerima masukan dari keyboard pada thread utamanya. Semua aplikasi memiliki paling tidak satu thread yang menggambarkan alur eksekusi utama program.

## E. Perbandingan Java dengan C++

Kedekatan Java dengan C++ menjadikan perlunya mengetahui perbandingan yang ada di antara mereka. Bahasa pemrograman C++ sangat tangguh tetapi rumit, sedangkan bahasa pemrograman Java dirancang lebih sederhana (sehingga lebih mudah dipelajari). Perbandingan Java dengan C++ dapat dilihat dari berbagai aspek sebagai berikut.

### Perbedaan Secara Umum

- Pada Java, manajemen memori dinamis dan pointer dilakukan secara otomatis. Otomatisasi ini menghilangkan kemungkinan bagi penggunaan pointer yang dapat merusak kestabilan program dan sistem operasi. Pada C/ C++ pemrogram melakukan sendiri alokasi dan dealokasi memori.
- Java tidak mengenal multiple inheritance seperti pada C/ C++. Multiple inheritance pada C/ C++ membingungkan dan berakibat pada sulitnya pembuatan aplikasi dan compiler. Sebagai ganti yang lebih baik, Java menggunakan interface.
- Java menggunakan bahasa OOP murni sedangkan C++ adalah bahasa OOP hybrid (dapat menggunakan model OOP maupun procedural).
- Java menggunakan model Exception dalam menangani kesalahan program.
- Java tidak menggunakan *compiler-directive* pada kode sumber (*source code*).
- Java memiliki fasilitas *garbage collection* yang bekerja secara otomatis untuk menjamin ketersediaan memori dengan membebaskan memori yang tidak terpakai lagi.
- Java sangat portable, baik file sumbernya maupun hasil kompilasinya. Hal ini berbeda dengan C/ C++ yang hanya portable pada level kode sumbernya saja dengan melakukan kodifikasi ketika dipindahkan ke platform yang berbeda.
- Spesifikasi bahasa dan implementasi Java sudah ditetapkan dan tidak mengikuti suatu mesin atau platform tertentu. Hal ini berbeda dengan C/ C++ yang sangat tergantung pada mesin dan platform yang akan digunakan.

### Berdasarkan Tujuan Desain

C++ didesain terutama untuk sistem pemrograman, yaitu pengembangan bahasa pemrograman C. Untuk itu prosedur bahasa pemrograman didesain untuk eksekusi program yang lebih efisien. pada C++ telah ditambahkan penunjang untuk program berorientasi objek dengan jenis statis, kecuali dalam hal pengendaliannya, mencakup pengaturan sumber daya dan pemrograman generik, khususnya juga ditambahkan '*standard library*' termasuk algoritma dan isi program generik.

Java pada awalnya dirancang untuk mendukung jaringan komputer. Java dipercaya sebagai mesin virtual yang aman dan sangat mudah. Java dipaket (*bundle*) dengan desain pustaka (libraries) yang luas untuk menyediakan abstraksi yang lengkap dari suatu platform. Java adalah bahasa pemrograman berorientasi objek berjenis statis yang penggunaan syntaxnya mirip dengan bahasa pemrograman C, tapi tidak kompatibel dengannya. Java didesain dari awal, dengan tujuan agar mudah digunakan dan dapat diakses oleh khalayak yang lebih luas.

### Berdasarkan Tujuan Pengembangan

Berdasarkan tujuan pengembangannya, Java dengan C++ memunculkan perbedaan prinsip dan perbedaan desain di masing-masing pihak sebagai berikut.

C++	Java
Kurang lebih kompatibel dengan bahasa sumber ( <i>source code</i> ) bahasa C.	Didesain dari awal tidak kompatibel dengan bahasa pemrograman manapun.
Memungkinkan panggilan langsung ke sistem perpustakaan asli ( <i>native system libraries</i> ).	Panggilan melalui Java Native Interface.
Dijalankan dengan fasilitas sistem keamanan yang rendah.	Dijalankan dengan mesin virtual yang aman (terlindungi).
Mendukung aritmatik yang tak terdaftar.	Tidak mendukung aritmatik yang tak terdaftar.
Tidak ada batasan standard atau ukuran untuk semua jenis tipe data numerik. Hanya diberikan ukuran relatif.	Pembatasan standard dan ukuran untuk semua jenis data.
Parameter yang digunakan dapat dengan value, pointer, atau reference.	Parameter yang digunakan selalu dengan value (objek yang diakses melalui referensi diubah ke value terlebih dulu, tidak langsung dari objek tersebut).
Pengaturan memory melalui pihak ketiga yang menyediakan pengumpulan sampah ( <i>garbage collection</i> ).	Pengumpulan sampah secara otomatis, walaupun demikian secara manual dapat diatur oleh pemrogram.
Standard Library C++ mempunyai lebih banyak pengaturan dan fungsi.	Standard Library Java berkembang sesuai masing-masing release/versi.
Pewarisan majemuk.	Pewarisan tunggal, bila dibutuhkan pewarisan majemuk hanya dapat melalui interface.

Dikompilasi dari berbagai sumber di internet.