

# BAB 1

## PENGENALAN TEKNOLOGI JAVA

---

### 1.1. SEJARAH SINGKAT JAVA

---

Berbicara mengenai Java ada dua bagian yang dibicarakan, yakni :

- Java sebagai bahasa pemrograman.
- Java sebagai sebuah platform.

**Java sebagai bahasa pemrograman** dimulai ketika pada tahun 1990, **James Gosling**, pemrogram yang bekerja pada **Sun Microsystems** mendapat tugas menciptakan program untuk mengendalikan perangkat elektronika rumah tangga.

*Gosling* dan kawan-kawan mulai mendesain program tersebut dengan bahasa pemrograman C++, yaitu bahasa pemrograman kelas wahid yang menjadi unggulan para pemrogram profesional saat itu, karena sifatnya yang berorientasi objek itu. Saat mulai menggunakan C++, *Gosling* mulai merasakan bahwa bahasa tersebut tidak cocok untuk tugas yang dibebankan kepada timnya. Mereka mulai mendapat masalah karena berbagai kerumitan yang dimiliki oleh C++, misalnya masalah pewarisan majemuk (**multiple inheritance**) dan masalah **bug** program, misalnya kebocoran memori (**memory leaks**). *Gosling* segera memutuskan untuk merancang sendiri bahasa pemrograman komputer yang sederhana dan mampu menghindarkannya dari masalah-masalah yang ia temui ketika menggunakan bahasa pemrograman C++.

Meskipun *Gosling* tidak menyukai kerumitan yang dimiliki oleh C++, tetapi ia menyukai sintaks dasar dan sifat berorientasi objek yang dimiliki oleh C++. Oleh karena itu, ketika mulai merancang bahasa pemrogramannya sendiri, ia menggunakan C++ sebagai modelnya dengan membuang fitur-fitur yang membuatnya susah. Rancangan bahasa pemrograman yang telah selesai ia buat, diberi nama **Oak** (konon diilhami oleh pohon *Oak* yang tumbuh di depan jendela kantornya).

Bahasa pemrograman **Oak** ini digunakan pertama kali dalam proyek yang disebut **proyek Green**, yaitu sistem pengendali alat-alat elektronik rumah tangga. Dengan sistem pengendali ini memungkinkan pengguna memanipulasi berbagai peralatan elektronik rumah tangga, termasuk TV, VCR, lampu dan telepon, dengan menggunakan telepon genggam yang disebut **Star7**. *Star7* ini menggunakan layar sentuh sebagai sarana antarmuka (interface) dengan pemakai. Layar sentuh *Star7* menampilkan sejumlah citra teranimasi, salah satu diantaranya adalah **Duke**, yang menjadi mascot java (seperti Penguin yang menjadi mascot Linux).

Berikutnya **Oak** dipakai dalam **proyek video-on-demand (VOD)**, yakni sebagai perangkat lunak yang mengendalikan sistem televisi interaktif. Meskipun proyek *Star7* dan *VOD* tidak pernah menjadi produk yang dipasarkan, *Sun Microsystems* memberikan kesempatan agar bahasa pemrograman **Oak** berkembang dan matang. Pada saat itulah *Sun* mengganti nama bahasa pemrograman **Oak** yang dibuat timnya menjadi bahasa pemrograman **Java** (konon diilhami saat minum kopi salah seorang timnya mengatakan Java, maksudnya asal bijih kopi yang sedang diminumnya) pada tanggal 23 Mei 1995.

**Java sebagai sebuah platform**, terdiri atas dua bagian utama, yakni :

- Java Virtual Machine (JVM)
- Java Application Programming Interface (Java API)

Pada dasarnya, ada berbagai macam platform (sistem operasi) tempat aplikasi-aplikasi perangkat lunak (software) dieksekusi seperti Microsoft Windows, Unix, Linux, Netware, Macintosh, dan OS/2. Namun, aplikasi-aplikasi yang berjalan pada suatu platform (misalnya Windows) tidak akan bisa dijalankan di platform yang lain (misalnya Linux) tanpa usaha kompilasi ulang, bahkan pengubahan kode program. Java platform merupakan perangkat lunak yang menjadi mesin virtual bagi aplikasi Java untuk dieksekusi. 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 Java Virtual Machine (JVM).

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)

---

## 1.2. TEKNOLOGI JAVA

---

Java Application Programming Interface (Java API) menyediakan fitur-fitur yang menarik untuk mengembangkan aplikasi, antara lain :

- **Applet**  
Java Applet merupakan program Java yang berjalan di atas browser. Dengan menggunakan Java Applet, maka halaman HTML akan lebih dinamis dan menarik, sangat cocok untuk pengembangan aplikasi-aplikasi berbasis web.

- **Java Networking**  
Java Networking merupakan sekumpulan Application Programming Interface (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 sekuriti.
- **Java Database Connectivity (JDBC)**  
JDBC menyediakan sekumpulan Application Programming Interface (API) yang dapat digunakan untuk mengakses database seperti Oracle, MySQL, PostgreSQL, Microsoft SQL Server.
- **Java Security**  
Java Security menyediakan sekumpulan Application Programming Interface (API) yang dapat digunakan untuk mengatur *security* dari aplikasi Java baik secara *high level* maupun *low level*, seperti *public/private key management* dan *certificates*.
- **Java Swing**  
Java Swing menyediakan sekumpulan Application Programming Interface (API) yang dapat digunakan untuk membangun aplikasi-aplikasi GUI (Graphical User Interface) dan model GUI yang diinginkan bisa bermacam-macam.
- **Java RMI**  
Java RMI menyediakan sekumpulan Application Programming Interface (API) yang dapat digunakan untuk membangun aplikasi-aplikasi Java yang mirip dengan model RPC (Remote Procedure Call), jadi objek-objek Java bisa di-*call* secara *remote* pada jaringan komputer.
- **Java 2D/ 3D**  
Java 2D/ 3D menyediakan sekumpulan Application Programming Interface (API) yang dapat digunakan untuk membangun grafik-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 (Java Server Page) yang mirip ASP dan PHP merupakan alternatif terbaik untuk solusi aplikasi internet
- **Java Native Interface (JNI)**  
JNI menyediakan sekumpulan Application Programming Interface (API) yang dapat digunakan untuk mengakses fungsi-fungsi pada library (\*.dll atau \*.so) yang dibuat dengan bahasa pemrograman yang lain seperti C++, dan Basic.
- **Java Sound**  
Java Sound menyediakan sekumpulan Application Programming Interface (API) yang dapat digunakan untuk memanipulasi *sound*.

- **Java IDL + COBRA**  
Java IDL (*Interface Definition Language*) menyediakan dukungan Java untuk implementasi COBRA (*Common Object Request Broker*) 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 Application Programming Interface (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.

Sebenarnya ada banyak sekali API-API yang disediakan oleh Java yang bisa dimanfaatkan sesuai kebutuhan, dapat dilihat di situs utama Java, yakni <http://java.sun.com>

---

### 1.3. KELEBIHAN JAVA

---

Mengapa Java sering dibicarakan dan memiliki daya tarik bagi para programmer? Karena Java merupakan teknologi yang unik dan revolusioner dan merupakan teknologi yang memiliki 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 sembarang platform asalkan Java Virtual Machine (JVM) untuk platform tersebut tersedia.
- **Pembuang Sampah (Garbage Collection)**  
Program Java membuang sendiri “sampah-sampah” yang tak berguna, artinya program tidak perlu menghapus objek-objek yang dialokasikannya di memori dan dealokasi memori secara otomatis. Ini menjadikan program Java bebas dari masalah-masalah pengelolaan memori.
- **Bebas Arsitektur (Architecture Neutral)**  
Pada dasarnya, program Java tidak dirancang untuk prosesor ataupun sistem operasi tertentu. Ia dirancang untuk bekerja pada berbagai arsitektur prosesor dan berbagai sistem operasi. Oleh karena itu, ia bekerja dengan interpreter Java untuk mengubah

program Java menjadi *byte-code* sebelum dijalankan oleh suatu mesin tertentu.

- **Tangguh (Robust)**

Karena 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*). *Exception* ini 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. Karena Java tidak mendukung pointer, 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 bisa dieksekusi secara lebih cepat karena langsung berjalan di atas perangkat keras yang bersangkutan, dari pada yang ditulis dalam Java yang berjalan di atas *Java Virtual Machine (JVM)*. *Metode native code* dikaitkan secara dinamis (*dynamically linked*) ke program Java, yakni dikaitkan dengan program saat runtime. Pada perkembangannya, bila Java berkembang lebih jauh dan mampu mengeksekusi fungsi-fungsi lebih cepat dari bahasa lain, *metode native* mungkin tidak diperlukan lagi.

---

### 1.3. KARAKTERISTIK JAVA

---

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

- **Sederhana (Simple)**

Java dirancang untuk mudah dipelajari, terutama bagi pemrogram yang telah mengenal C++ akan mudah sekali untuk berpindah ke Java. Selain itu pengembang Java banyak membuang fitur-fitur yang tidak diperlukan seperti bahasa pemrograman tingkat tinggi, seperti *pointer* matematis, *casting* tipe *implicit*, struktur atau *union*, *operator overloading*, *template*, *header filer*, maupun pewarisan majemuk (*multiple inheritance*). Pemrograman Java bisa dikatakan semudah C dan seampuh C++.

- **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-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. Hasil kompilasinya berupa file kode byte (*byte-code*), yang mirip dengan kode mesin, yang dapat dieksekusi di bawah sistem operasi apapun asal memiliki interpreter Java. 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.

---

## 1.4. PERBEDAAN JAVA DENGAN C++

---

Dibawah ini akan dijelaskan perbandingan bahasa pemrograman Java dengan bahasa pemrograman C++ dari berbagai aspek, sebagai berikut;

### PERBEDAAN TUJUAN PENDESAINANNYA

Perbedaan antara bahasa pemrograman Java dengan bahasa pemrograman C++ dapat ditelusuri dari sistem pewarisannya, karena mereka memiliki desain yang berbeda tujuannya :

- C++ didisain terutama untuk sistem pemrograman, mengembangkan bahasa pemrograman C. Untuk itu prosedur bahasa pemrograman didisain 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 dibendel dengan desain library 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 didisain dari awal, dengan tujuan agar mudah digunakan dan dapat diakses oleh khalayak yang lebih luas.

## PERBEDAAN TUJUAN PENGEMBANGANNYA

Perbedaan tujuan pengembangan bahasa pemrograman Java dengan bahasa pemrograman C++, mengakibatkan perbedaan prinsip dan perbedaan desain antara masing-masing bahasa pemrograman tersebut.

<b>C++</b>	<b>Java</b>
Kurang lebih kompatibel dengan bahasa sumber (source code) bahasa C.	Didisain dari awal tidak kompatibel dengan bahasa pemrograman manapun.
Memungkinkan panggilan langsung ke sistem perpustakaan asli (native system libraries).	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 <i>value</i> , <i>pointer</i> atau <i>reference</i> .	Parameter yang digunakan selalu dengan <i>value</i> , walaupun objek diakses melalui references akan diubah ke <i>value</i> terlebih dulu, tidak langsung dari objek tersebut.
Pengaturan memory melalui pihak ketiga yang menyediakan pengumpulan sampah (garbage collection).	Pengumpulan sampah secara otomatis, walaupun demikian secara manual dapat diatur oleh programmer.
Standard Library C++ mempunyai lebih banyak pengaturan dan fungsi dari standard library pada Java.	Standard Library Java berkembang sesuai masing-masing release/ versi.
Pewarisan majemuk.	Pewarisan tunggal, bila dibutuhkan pewarisan majemuk hanya dapat melalui interface.

Bahasa pemrograman C++ sangat tangguh tetapi rumit, sedangkan bahasa pemrograman Java dirancang lebih sederhana (dan oleh karenanya lebih mudah dipelajari).

## PERBEDAAN UMUM

- Pada Java, manajemen memory 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 memory.
- 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 source code.
- Java memiliki fasilitas garbage collection yang bekerja secara otomatis untuk menjamin ketersediaan memory dengan membebaskan memory 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, itupun 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.