

Java (platform perangkat lunak)

Java adalah satu set [perangkat lunak komputer](#) dan spesifikasi yang dikembangkan oleh [Sun Microsystems](#), yang kemudian diakuisisi oleh [Oracle Corporation](#), yang menyediakan sistem untuk mengembangkan [aplikasi perangkat lunak](#) dan menerapkan hal itu dalam sebuah [cross-platform](#) lingkungan komputasi. digunakan di berbagai macam [platform komputasi](#) dari [perangkat embedded](#) dan [ponsel](#) ke [server perusahaan](#) dan [superkomputer](#). Sementara mereka yang kurang umum dari standalone aplikasi Java, [Java applet](#) yang berjalan di aman, lingkungan [tersandbox](#) untuk memberikan banyak fitur dari aplikasi asli dan dapat tertanam ke dalam [HTML](#) halaman.

Penulisan dalam [bahasa pemrograman Java](#) adalah cara utama untuk menghasilkan kode yang akan digunakan sebagai [kode byte](#) pada sebuah [Mesin Virtual Java](#) (JVM); bytecode [compiler](#) juga tersedia untuk bahasa-bahasa lain, termasuk [Ada](#), [JavaScript](#), [Python](#), dan [Ruby](#). Selain itu, beberapa bahasa telah dirancang untuk berjalan secara native pada JVM, termasuk Scala, Clojure, dan Apache Groovy. sintaks meminjam banyak dari [C](#) dan [C++](#), tetapi berorientasi objek memiliki dimodelkan setelah [Smalltalk](#) dan Objective-C.^[1] eschews tertentu tingkat rendah konstruksi seperti pointer dan memiliki memori yang sangat sederhana model di mana setiap objek yang dialokasikan di heap dan semua variabel dari jenis objek referensi. Manajemen memori ditangani melalui otomatis terintegrasi [pengumpulan sampah](#) yang dilakukan oleh JVM.

Pada November 13, 2006, Sun Microsystems membuat sebagian besar implementasi dari tersedia di bawah [GNU General Public License](#) (GPL)

Terbaru versi Java 8, hanya didukung (misalnya dengan security update) versi 2016. Oracle (dan lain-lain) telah mengumumkan bahwa menggunakan versi yang lebih tua (dari 8) mereka JVM pelaksanaan menghadirkan risiko serius akibat belum terselesaikan masalah keamanan.

Platform

Platform Java adalah seperangkat program yang memfasilitasi mengembangkan dan menjalankan program yang ditulis dalam [Java](#) bahasa pemrograman. Java platform akan mencakup eksekusi mesin (disebut [mesin virtual](#)), compiler dan satu set [perpustakaan](#); mungkin juga ada tambahan [server](#) dan alternatif perpustakaan yang tergantung pada persyaratan. Java adalah tidak spesifik untuk setiap prosesor atau [sistem operasi](#) seperti Java platform telah diimplementasikan untuk berbagai macam perangkat keras dan sistem operasi dengan maksud untuk memungkinkan program Java untuk menjalankan identik pada semua dari mereka. Berbeda platform target kelas yang berbeda dari perangkat dan aplikasi domain:

- Java Card: Sebuah teknologi yang memungkinkan kecil aplikasi berbasis Java (applet) yang akan berjalan dengan aman pada [smart card](#) dan kecil yang sama-memori perangkat.
- Java ME (Micro Edition): Menentukan beberapa set yang berbeda dari perpustakaan (dikenal sebagai profil) untuk perangkat dengan penyimpanan terbatas, layar, dan kapasitas daya. Hal ini sering digunakan untuk mengembangkan aplikasi untuk perangkat mobile, Pda, TV [set-top box](#), dan printer.
- Java SE (Standard Edition): Untuk keperluan umum digunakan pada Pc desktop, server, dan perangkat sejenis.
- Java EE (Enterprise Edition): SE ditambah berbagai Api yang berguna untuk multi-tier [client-server](#) aplikasi enterprise.

Platform Java terdiri dari beberapa program, masing-masing menyediakan sebagian dari kemampuan secara keseluruhan. Misalnya, Java compiler, yang mengubah kode sumber Java menjadi bytecode Java (bahasa menengah untuk JVM), diberikan sebagai bagian dari Java Development Kit (JDK). The [Java Runtime Environment](#) (JRE), melengkapi JVM dengan [just-in-time \(JIT\) compiler](#), yang mengubah menengah bytecode menjadi asli kode mesin on the fly. Platform Java juga mencakup serangkaian luas dari perpustakaan.

Komponen penting dalam platform Java compiler bahasa, perpustakaan, dan lingkungan runtime yang di intermediate mengeksekusi bytecode menurut aturan yang tercantum dalam virtual machine spesifikasi. taek.

Jantung platform Java adalah konsep "virtual machine" yang mengeksekusi program [bytecode Java](#). Bytecode ini tetap bisa berjalan, tidak peduli terhadap jenis perangkat keras atau sistem operasinya. Ada JIT (Just In Time) compiler dalam *Java Virtual Machine* atau JVM. JIT compiler menerjemahkan bytecode Java ke native prosesor instruksi pada saat run-time dan cache kode asli dalam memori selama eksekusi.

Penggunaan bytecode sebagai perantara bahasa yang memungkinkan program Java untuk berjalan pada platform apapun yang memiliki mesin virtual yang tersedia. Penggunaan JIT compiler berarti bahwa aplikasi Java, setelah penundaan singkat selama pemuatan dan setelah mereka memiliki "pemanasan" dengan menjadi semua atau sebagian besar JIT-dikompilasi, cenderung berjalan secepat asli program. Sejak JRE versi 1.2, Matahari JVM pelaksanaan sudah termasuk [just-in-time compiler](#) bukan seorang penerjemah.

Meskipun program-program Java [cross-platform](#) atau platform independen, kode Mesin Virtual Java (JVM) yang mengeksekusi program-program ini tidak. Setiap operasi yang didukung platform memiliki JVM.

Bahasa Java telah mengalami beberapa perubahan sejak rilis JDK (Java Development Kit) 1.0 pada tanggal 23 Januari 1996, serta berbagai penambahan dari kelas-kelas dan paket-paket standar [perpustakaan](#). Sejak J2SE 1.4 Java Community Process (JCP) telah diatur evolusi dari Bahasa Java. JCP menggunakan *Java Spesifikasi Permintaan* (JSRs) untuk mengusulkan dan menentukan penambahan dan perubahan untuk platform Java. Dalam *Bahasa Keterangan* (JLS) menentukan bahasa; perubahan JLS dikelola di bawah JSR 901

Matahari dirilis *JDK 1.1* pada tanggal 19 Februari 1997. Utama termasuk penambahan luas retooling dari AWT acara model, kelas batin ditambahkan untuk bahasa, JavaBeans dan JDBC.

J2SE 1.2 (8 Desember 1998) – Codename *Bermain anak*. Ini dan rilis berikutnya melalui J2SE 5.0 yang namanya *Java 2* dan nama versi "J2SE" (Java 2 Platform, Standard Edition) diganti JDK untuk membedakan basis platform J2EE (Java 2 Platform, Enterprise Edition), J2ME (Java 2 Platform, Micro Edition). Tambahan utama termasuk refleksi, koleksi framework, Java IDL (interface bahasa deskripsi implementasi CORBA interoperabilitas), dan integrasi Ayunan grafis API ke inti kelas. Sebuah [Java Plug-in](#) dirilis, dan Matahari JVM dilengkapi dengan [JIT compiler](#) untuk pertama kalinya.

J2SE 1.3 (8 Mei 2000) – Codename *Kestrel*. Perubahan penting termasuk bundling HotSpot JVM (JVM HotSpot pertama kali dirilis pada bulan April 1999 untuk J2SE 1.2 JVM), JavaSound, Penamaan dan Direktori Interface (JNDI) dan Platform Java Debugger Arsitektur (JPDA).

J2SE 1.4 (6 Februari 2002) – Codename *Merlin*. Ini menjadi rilis pertama dari platform Java yang dikembangkan dalam Masyarakat sebagai Proses JSR 59.^[17] perubahan Besar termasuk ekspresi reguler dimodelkan setelah [Perl](#), terkecuali chaining, terpadu [XML](#) parser dan [XSLT](#) processor (JAXP), dan Java Web Start.

J2SE 5.0 (Agustus 30, 2004) – Codename *Tiger*. Itu awalnya berjumlah 1.5, yang masih digunakan sebagai internal nomor versi. yang Dikembangkan di bawah JSR 176, Harimau menambahkan beberapa signifikan fitur bahasa baru termasuk untuk masing-masing loop, obat generik, autoboxing dan var-args.

Java SE 6 (11 Desember 2006) – Codename *Mustang*. Itu dibundel dengan manajer database dan memfasilitasi penggunaan bahasa scripting dengan JVM (seperti [JavaScript](#) menggunakan [Mozilla's](#) Badak engine). Pada versi ini, Matahari diganti nama "J2SE" dengan *Java SE* dan menjatuhkan ".0" dari nomor versi. perubahan besar Lainnya termasuk dukungan untuk pluggable penjelasan (JSR 269), banyak [GUI](#) perbaikan, termasuk UI asli perangkat tambahan untuk mendukung tampilan dan nuansa dari [Windows Vista](#), dan perbaikan untuk Platform Java Debugger Arsitektur (JPDA) & JVM Alat Antarmuka untuk pemantauan yang lebih baik dan pemecahan masalah.

Java SE 7 (28 Juli 2011) - dengan kode nama *Dolphin*. Versi ini dikembangkan di bawah JSR 336. Ini menambahkan sejumlah kecil perubahan bahasa termasuk string di switch, mencoba-dengan-sumber-sumber dan jenis inferensi untuk generic contoh penciptaan. JVM diperpanjang dengan dukungan untuk bahasa dinamis, sedangkan kelas perpustakaan diperpanjang antara lain dengan bergabung/garpu kerangka,¹ peningkatan new file I/O library dan dukungan untuk protokol jaringan seperti SCTP. Java 7 Update 76 dirilis pada Januari 2015, dengan berakhirnya tanggal 14 April 2015

Pada bulan Juni 2016, setelah publik terakhir update Java 7, "[jarak jauh dieksplotasi](#)" bug keamanan di Java 6, 7 dan 8 di mana diumumkan. Yang current version, *Java SE 8* (18 Maret 2014). Perubahan penting termasuk

bahasa tingkat dukungan untuk ekspresi lambda (penutupan) dan metode default, Proyek Nashorn JavaScript runtime, Tanggal dan Waktu yang baru API terinspirasi oleh Joda Waktu, dan penghapusan PermGen. Versi ini tidak secara resmi didukung pada platform Windows XP. Namun, karena akhir dari 7 siklus hidup ini adalah versi direkomendasikan untuk pengguna XP. Sebelumnya, hanya tidak resmi instalasi manual metode yang telah dijelaskan untuk Windows XP SP3. Hal ini mengacu pada JDK8, mengembangkan platform untuk Java yang juga mencakup sepenuhnya berfungsi [Java Runtime Environment](#). 8 didukung pada Windows Server 2008 R2 SP1, Windows Vista SP2 dan Windows 7 SP1, Ubuntu 12.04 LTS dan lebih tinggi (dan beberapa Os lain).

JDK 9 Akses Awal membutuhkan Windows 7 SP1, Windows Server 2012, Ubuntu 16.04 LTS atau lebih tinggi (atau OS lain yang didukung, seperti OS X atau Solaris).

Selain bahasa perubahan, perubahan yang signifikan telah dibuat untuk Java class library selama bertahun-tahun, yang telah berkembang dari beberapa ratus kelas dalam JDK 1.0 untuk lebih dari tiga ribu di J2SE 5.0. Seluruh Api yang baru, seperti Ayunan dan Java 2D, telah berkembang, dan banyak dari asli JDK 1.0 kelas dan metode yang telah usang.

Java SE 10 (JDK 10) untuk umum di bulan Maret 2018 ini maka Oracle telah melakukan rilis cepat tahun ini. Karena kurang lebih 6 bulan sebelumnya mereka baru saja merilis Java 9. Dan ini sesuai dengan pernyataan mereka pada tahun lalu mengenai perilisan cepat. Saat ini Oracle dengan perubahan yang disebut dengan “siklus inovasi” akan melakukan panggilan untuk perilisan fitur dalam jangka waktu setiap 6 bulan, pembaruan rilis dalam setiap kuartal, dan juga dukungan jangka panjang (LTS) akan rilis setiap 3 tahun. Rencana berikutnya adalah Java 11, yang perilisannya direncanakan pada bulan September. Kemudian versi LTS berikutnya setelah itu adalah Java 17, yang dijadwalkan akan dirilis pada bulan September, 2021. Sementara itu wakil presiden Pengembangan Perangkat Lunak di Java Platform Group Oracle, Georges Saab, menyatakan bahwa Oracle telah berkomitmen untuk melakukan evolusi dengan cepat dengan memberikan inovasi baru pada platform Java. Java 10 menjadi yang pertama dalam siklus rilis baru dari model lisensi Oracle. Ia juga menyatakan bahwa dengan kemudahan dan kesederhanaan perilisan ini, mereka sangat bangga dengan memperkenalkan fitur-fitur baru yang berguna, menghilangkan elemen yang tidak perlu dan lebih mudah digunakan bagi para developer.

Fitur dan Peningkatan Baru Pada Java 10

Java 10 hadir dengan 12 peningkatan tambahan baru yang didefinisikan melalui proses Proposal Peningkatan JDK (JEP), yang merupakan versi Oracle dari proses Permintaan Spesifikasi Java (JSR) dari JCP. Peningkatan tersebut adalah:

1. *Inferensi Tipe Variabel Lokal*: Meningkatkan bahasa Java untuk memperluas tipe inferensi ke deklarasi variabel lokal dengan menggunakan *initializers*. Oleh karena itu, Java memperkenalkan *var*, yang merupakan sesuatu yang umum dan biasa digunakan dalam bahasa pemrograman lainnya.
2. *Menggabungkan berbagai repositori JDK forest* ke dalam satu repositori (tunggal) untuk menyederhanakan dan merampingkan development.
3. *Garbage Collector Interface*: Meningkatkan isolasi *source code* dari *garbage collector* berbeda dengan memperkenalkan *interface garbage collector (GC)* yang bersih.
4. *Full GC paralel untuk G1*: Memperbaiki kasus latensi G1 terburuk dengan membuat *full GC* paralel.
5. *Application Data-Class Sharing*: Untuk meningkatkan startup dan footprint, JEP ini memperluas fitur Class-Data Sharing (“CDS”) yang ada untuk memungkinkan kelas aplikasi ditempatkan dalam arsip bersama.
6. *Thread-Local Handshakes*: Memperkenalkan cara melakukan callback pada thread tanpa harus melakukan safe point VM secara global. Membuatnya menjadi mungkin dan murah untuk menghentikan thread tunggal dan tidak hanya semua thread atau tidak satu pun.

7. Menghapus tool Native-Header Generator, dengan menghapus javah tool dari JDK karena telah digantikan oleh keunggulan fungsionalitas di javac.
8. Ekstensi tambahan pada *Unicode Language-Tag*: Peningkatkan pada java.util.Locale dan API terkait untuk mengimplementasikan ekstensi Unicode tambahan dari tag bahasa BCP 47.
9. Alokasi Heap pada Perangkat Memori Alternatif: Ini memungkinkan HotSpot VM untuk mengalokasikan objek heap Java pada perangkat memori alternatif, seperti NV-DIMM, yang dapat ditentukan oleh pengguna.
10. Mengaktifkan compiler JIT berbasis Java, Graal, untuk digunakan sebagai compiler JIT eksperimental pada platform Linux / x64.
11. Root Certificate: Menyediakan secara default kumpulan dari *root Certification Authority (CA)* di JDK.
12. Versi Rilis Berbasis Waktu: Merevisi skema versi-string dari Java SE Platform dan JDK, dan informasi versi terkait, untuk model rilis berbasis waktu saat ini dan masa yang akan datang.

Kemudian Java mengeluarkan versi terbaru yaitu Java 15 melanjutkan siklus rilis enam bulannya dan merupakan rilis berukuran sedang yang berguna bagi sebagian besar pengembang. Pembaruan Java sangat cepat. Sepertinya saya baru saja selesai menginstal JDK15 di komputer saya, kemudian 16 sudah tersedia. Meskipun telah dikeluhkan dalam banyak aspek, Java tetap menjadi bahasa pemrograman yang paling banyak digunakan di seluruh dunia sepanjang tahun. Dan itu selalu menjaga vitalitasnya selama bertahun-tahun untuk pembaruan, pengoptimalan, dan inovasi yang berkelanjutan.

Java masih menjadi bahasa pengkodean utama saya meskipun saya baru-baru ini menulis semakin banyak kode Go, terkadang Python dan Scala. Layanan terpenting perusahaan masih perlu menggunakan Java. Salah satu alasannya cukup untuk menjelaskan, "**Semua orang tahu Java**" membuatnya lebih stabil dan lebih mudah dipelihara. Ketika JDK17 dirilis pada September 2021, kami pasti akan bermigrasi ke sana pada titik tertentu. Burung awal□ selalu yang tercepat beradaptasi di masa depan.

Pembaruan Java sangat cepat. Sepertinya saya baru saja selesai menginstal JDK15 di komputer saya, kemudian 16 sudah tersedia. Meskipun telah dikeluhkan dalam banyak aspek, Java tetap menjadi bahasa pemrograman yang paling banyak digunakan di seluruh dunia sepanjang tahun. Dan itu selalu menjaga vitalitasnya selama bertahun-tahun untuk pembaruan, pengoptimalan, dan inovasi yang berkelanjutan.

Java masih menjadi bahasa pengkodean utama saya meskipun saya baru-baru ini menulis semakin banyak kode Go, terkadang Python dan Scala. Layanan terpenting perusahaan masih perlu menggunakan Java. Salah satu alasannya cukup untuk menjelaskan, "**Semua orang tahu Java**" membuatnya lebih stabil dan lebih mudah dipelihara. Ketika JDK17 dirilis pada September 2021, kami pasti akan bermigrasi ke sana pada titik tertentu. Burung awal□ selalu yang tercepat beradaptasi di masa depan.