

JAVA

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 dapattertanam ke dalam [HTML](#) halaman.

PLATFORM

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 bytcode menjadi asli kode mesin on the fly. Platform Java juga mencakup serangkaian luas dari perpustakaan

FITUR-FITUR

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.