

Sejarah perkembangan java

[Bahasa pemrograman](#) Java pertama lahir dari The Green Project, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan [Oak](#). Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, [James Gosling](#) dan Bill Joy, beserta sembilan pemrogram lainnya dari [Sun Microsystems](#). Salah satu hasil proyek ini adalah maskot Duke yang dibuat oleh [Joe Palrang](#).

Pertemuan proyek berlangsung di sebuah gedung perkantoran Sand Hill Road di Menlo Park. Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program Java Oak pertama, yang ditujukan sebagai pengendali sebuah peralatan dengan teknologi layar sentuh (**touch screen**), seperti pada PDA sekarang ini. Teknologi baru ini dinamai “*7” (Star Seven).

Setelah era Star Seven selesai, sebuah anak perusahaan TV kabel tertarik ditambah beberapa orang dari proyek The Green Project. Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 Hamilton Avenue, Palo Alto.

Perusahaan baru ini bertambah maju: jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian Internet sebagai medium yang menjembatani kerja dan ide di antara mereka. Pada awal tahun 1990-an, Internet masih merupakan rintisan, yang dipakai hanya di kalangan akademisi dan militer.

Mereka menjadikan perambah (**browser**) Mosaic sebagai landasan awal untuk membuat perambah Java pertama yang dinamai Web Runner, terinspirasi dari film 1980-an, Blade Runner. Pada perkembangan rilis pertama, Web Runner berganti nama menjadi Hot Java.

Pada sekitar bulan Maret 1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Kesuksesan mereka diikuti dengan untuk pemberitaan pertama kali pada surat kabar **San Jose Mercury News** pada tanggal [23 Mei 1995](#).

Sayang terjadi perpecahan di antara mereka suatu hari pada pukul 04.00 di sebuah ruangan hotel Sheraton Palace. Tiga dari pimpinan utama proyek, Eric Schmidt dan George Paolini dari Sun Microsystems bersama Marc Andreessen, membentuk Netscape.

Nama Oak, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja “bapak java”, James Gosling. Nama Oak ini tidak dipakai untuk versi release Java karena sebuah perangkat lunak sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi “Java”. Nama ini diambil dari kopi murni yang digiling langsung dari biji (kopi tubruk) kesukaan Gosling.

Compiler

Compiler (kompilator) adalah sebuah program komputer yang berguna untuk menerjemahkan program komputer yang ditulis dalam bahasa pemrograman tertentu menjadi program yang ditulis dalam bahasa pemrograman lain.

Terlepas dari pengertiannya yang luas, istilah **compiler** biasa digunakan untuk program komputer yang menerjemahkan program yang ditulis dalam bahasa pemrograman tingkat tinggi (semacam bahasa Pascal, C++, BASIC, FORTRAN, Visual Basic, Visual C#, Java, xBase, atau COBOL) menjadi bahasa mesin, biasanya dengan bahasa Assembly sebagai perantara.

Arsitektur **compiler** modern biasanya bukan lagi merupakan program tunggal namun merupakan rangkaian komunikasi antar program dengan tugas spesifik masing-masing. Program-program tersebut beserta tugasnya secara umum terdiri dari:

- **Compiler** itu sendiri, yang menerima kode sumber dan menghasilkan bahasa tingkat rendah (**assembly**)
- **Assembler**, yang menerima keluaran **compiler** dan menghasilkan berkas objek dalam bahasa mesin
- **Linker**, yang menerima berkas objek keluaran **assembler** untuk kemudian digabungkan dengan pustaka-pustaka yang diperlukan dan menghasilkan program yang dapat dieksekusi (**executable**)

Compiler yang menggunakan arsitektur ini misalnya GCC, Clang dan FreeBASIC.

Beberapa **compiler** tidak menggunakan arsitektur di atas secara gamblang, dikarenakan komunikasi antar program jauh lebih lambat dibandingkan jika komunikasi dilakukan secara internal di dalam satu program. Sehingga compiler-compiler tersebut mengintegrasikan assembler dan linker di dalam compiler. Namun, biasanya arsitektur yang digunakan pun tidak kaku dan mengizinkan penggunaan assembler maupun linker eksternal (berguna jika assembler dan linker internal bermasalah). Compiler yang menggunakan arsitektur ini salah satunya adalah Free Pascal.

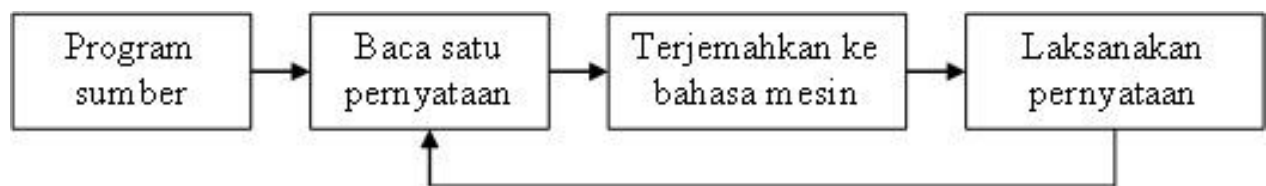
Interpreter

Dalam ilmu komputer, **penerjemah** atau lebih dikenal dengan **interpreter** merupakan perangkat lunak yang berfungsi melakukan eksekusi sejumlah instruksi yang ditulis dalam suatu bahasa pemrograman tanpa terlebih dahulu menyusunnya menjadi program bahasa mesin. Interpreter umumnya menggunakan salah satu strategi berikut untuk menjalankan program:

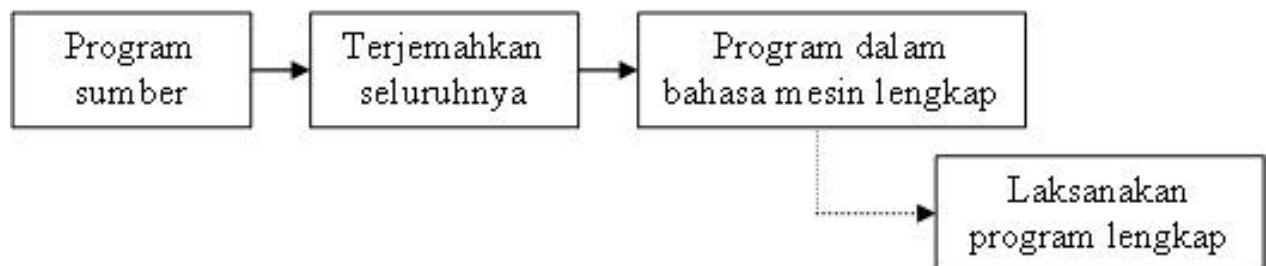
1. Mengeksekusi kode sumber secara langsung, atau
2. Menerjemahkannya ke dalam serangkaian p-code kemudian mengeksekusinya, atau
3. Mengeksekusi kode yang telah dikompilasi sebelumnya oleh compiler yang merupakan bagian dari sistem penerjemahan.

Proses ini sangat berbeda dengan compiler, dimana pada compiler, hasilnya sudah langsung berupa satu kesatuan perintah dalam bentuk bahasa mesin, dimana proses penterjemahan dilaksanakan sebelum program tersebut dieksekusi.

Perl, Python, Ruby, dan MATLAB adalah beberapa contoh perangkat lunak penerjemah bertipe 2, sementara Java termasuk dalam kategori tipe 3, namun dalam beberapa kasus Java dapat digolongkan pula ke dalam kategori tipe 2.



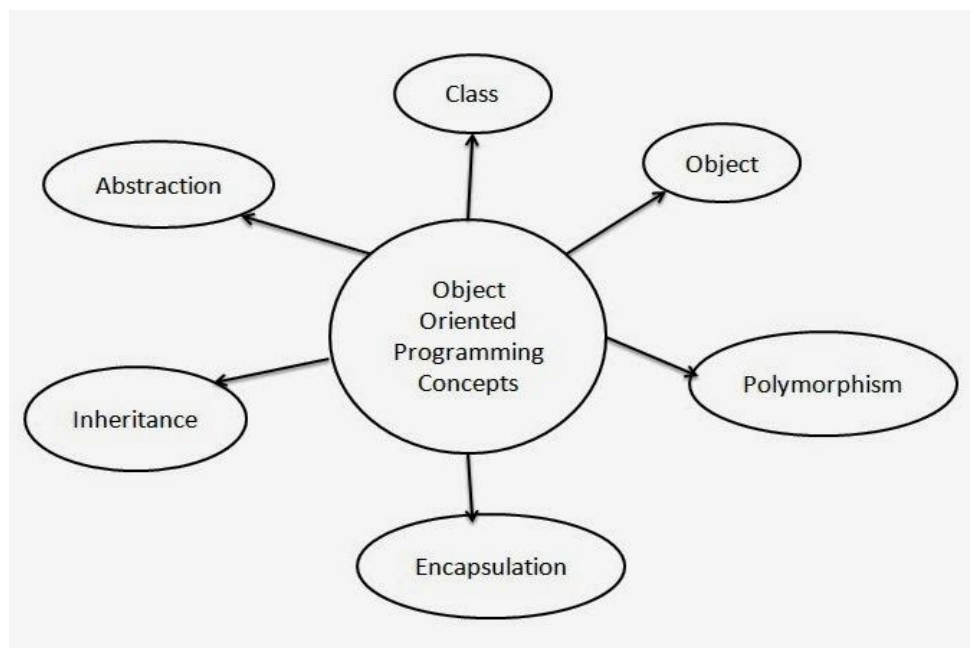
a) Dengan interpreter



b) Dengan compiler

Pemrograman berorientasi objek (Inggris: *object-oriented programming* disingkat **OOP**) merupakan **paradigma pemrograman** berdasarkan konsep "objek", yang dapat berisi **data**, dalam bentuk *field* atau dikenal juga sebagai atribut; serta kode, dalam bentuk fungsi/prosedur atau dikenal juga sebagai *method*. Semua data dan fungsi di dalam paradigma ini dibungkus dalam *kelas-kelas* atau *objek-objek*. Bandingkan dengan logika **pemrograman terstruktur**. Setiap objek dapat menerima **pesan**, memproses data, dan mengirim pesan ke objek lainnya.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam **teknik peranti lunak** skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

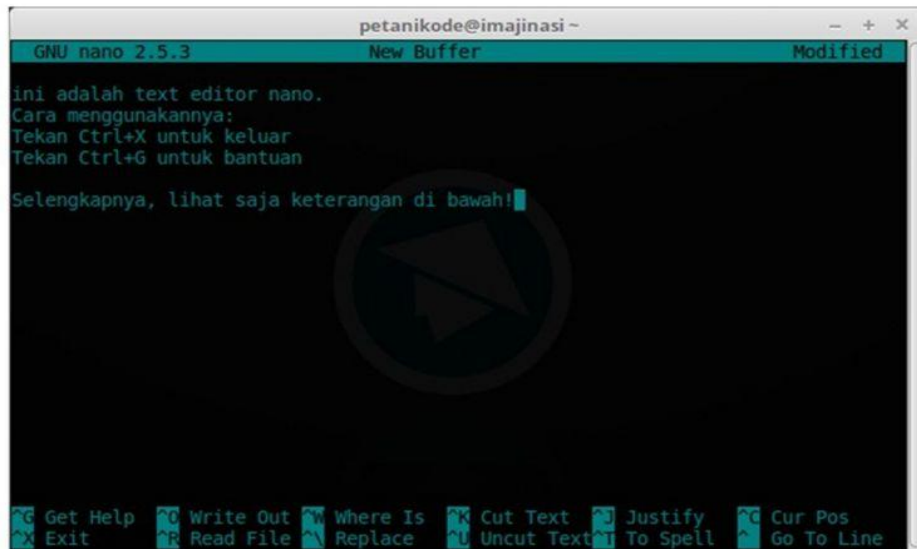


(https://id.m.wikipedia.org/wiki/Pemrograman_berorientasi_objek)

Persiapan awal pemrograman Java pertama adalah mempersiapkan alat-alatnya. Beberapa alat yang harus dipersiapkan diantaranya adalah:

- Teks Editor

Teks editor dipakai untuk menuliskan kode dan anda juga bisa memakai teks editor favorit anda selain contoh yang kami berikan dibawah ini.



1. JDK [Java Development it]

JDK atau Java Development Kit ini dipakai untuk melakukan kompilasi, debugging atau memeriksa error dan juga untuk menjalankan program Java. Pada JDK ini, ada dua hal penting yang harus anda ketahui, yakni:

Kompilator: Untuk menerjemahkan kode Java menjadi bytecode.

Debugger: Berguna untuk memeriksa erros pada kode.

Pada sistem operasi Windows, anda bisa memakai JDK dari Oracle, sementara pada Linux anda bisa memakai openJDK. Untuk nama paket openJDK pada Linux adalah openjdk. Pakailah fasilitas *apt-cache search* untuk mencari tahu tentang openjdk yang disediakan pada server repositori.

Disediakan tiga openJDK dimana versi terbarunya adalah openjdk 8. Jika pada Linux yang anda pakai belum tersedia versi 8, maka direkomendasikan untuk memakai versi 7 saja. Untuk install openjdk 8, silahkan anda ketik perintah dibawah ini:

sudo apt-get install openjdk-8-jdk

Jika sudah diinstall, silahkan periksa versi berapa yang sudah terpasang dengan memakai perintah: **java -version**

- JRE [Java Runtime Environment]

Jika JRE atau Java Runtime Environment dipakai untuk menjalankan program Java, maka JRE bisa dipasang jika anda ingin menjalankan program atau aplikasi Java. JRE pada Linux bisa diinstall lewat terminal dan biasanya sesudah install openjdk, maka JRE juga akan ikut terinstall. Namun jika belum, maka bisa diinstall sendiri. Berikut adalah perintah untuk install JRE versi 8:

sudo apt-get install openjdk-8-jre

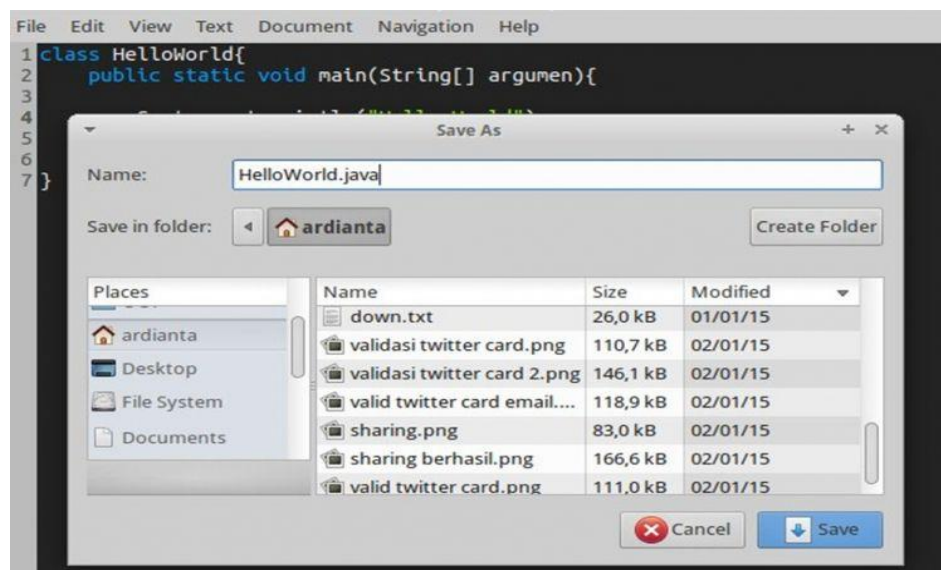
Sesudah semua siap, maka waktunya anda mencoba membuat sebuah program. Program yang akan kita buat kali ini adalah program **Hello World** yang sudah sangat dikenal sedunia. Bagi anda yang seorang calon programmer, maka harus bisa membuat program ini.

- Tulis Program

Silahkan anda buka teks editor kemudian tulis program berikut ini:

```
class HelloWorld{  
    public static void main(String[] argumen){  
        System.out.println("Hello World!");  
    }  
}
```

Kemudian, simpan dengan nama **HelloWorld.java** pada direktori home supaya lebih mudah diakses.

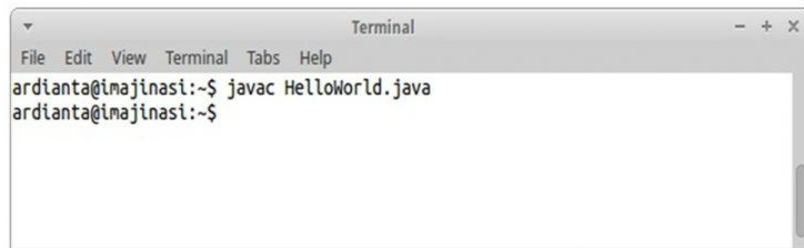


- Kompilasi Kode

Sesudah itu, silahkan anda buka terminal kemudian ketik perintah dibawah ini untuk mengompilasi:

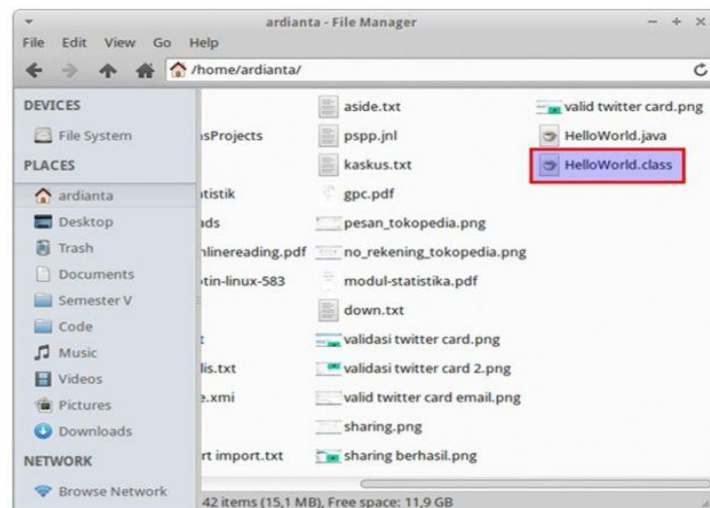
```
javac HelloWorld.java
```

Karena disimpan di direktori home, maka anda hanya perlu langsung mengkompilasinya saja.



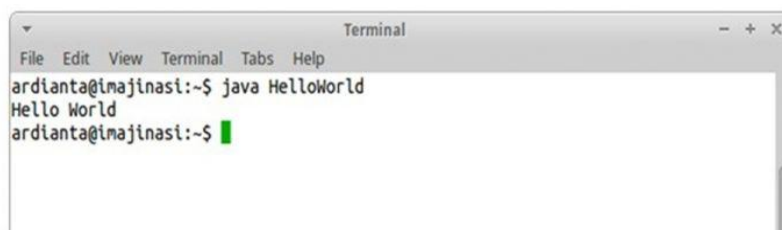
```
Terminal
File Edit View Terminal Tabs Help
ardianta@imajinasi:~$ javac HelloWorld.java
ardianta@imajinasi:~$
```

Jika tidak ada error, berarti program sudah sukses dikompilasi. Sesudah itu, maka akan dihasilkan sebuah berkas baru bernama HelloWorld.class dan berkas ini berisi bytecode yang akan dijalankan JRE.



- Jalankan Program

Untuk menjalankan program, pakailah perintah **java NamaKelas**.



```
Terminal
File Edit View Terminal Tabs Help
ardianta@imajinasi:~$ java HelloWorld
Hello World
ardianta@imajinasi:~$
```

