



POLITEKNIK MASAMY INTERNASIONAL

SK Menristekdikti RI Nomor: 731/KPT/I/2018

Jalan Ikan Paus No.10-15 Kertosari Banyuwangi - 68411
Telp (0333) 3384593 – <http://polmain.info>

Form:
B.Ak/eva/04/20

PROGRAM STUDI D3 TEKNIK KOMPUTER

Nama Dosen : *Arif Fahmi, S.T.,M.T.*

Mata Kuliah : Pemrograman Berorientasi Object

Semester : 4 (Empat)

Kode Mata Kuliah : TKV4044

Th. Akdm : 2019/2020.

BAB

Pengenalan Bahasa Pemrograman Java

SUB BAB :

1. Apa itu Teknologi Java
2. Cara kerja Pemrograman Java
3. Perbedaan Pemrograman Prosedural dengan Pemrograman Berorientasi Object
4. Java IDE (Integrated Development Environment)
5. Konsep Pemrograman Berorientasi Object

TUJUAN MATERI

1. Mahasiswa memahami perbedaan pemrograman prosedural dengan pemrograman berorientasi object (*)
2. Mahasiswa memahami konsep objek oriented dan aplikasinya (*)
3. Mahasiswa memahami konsep Java IDE (Integrated Development Environment)

Catatan :

* = Berbasis pemrograman Java

REFERENSI

1. An Introduction to Object Oriented Programming with Java, C. Thomas Wu, McGraw-Hill, New York, 2010.
2. J.Eck, D. (2006). *Introduction to Programming Using Java*. Geneva.
3. <https://medium.com/@aliirawan/konsep-pemrograman-berorientasi-objek-di-java-2cf6bca41f74>

Pengenalan Bahasa Pemrograman JAVA

1. Apa Itu Teknologi Java ?

1.1 Sebuah Bahasa Pemrograman

Sebagai sebuah bahasa pemrograman, Java dapat membuat seluruh bentuk aplikasi, *desktop*, *web* dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain.

Java adalah bahasa pemrograman yang **berorientasi objek (OOP)** dan dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*.

1.2 Sebuah Development Environment

Sebagai sebuah peralatan pembangun, teknologi Java menyediakan banyak *tools* : *compiler*, *interpreter*, penyusun dokumentasi, paket kelas dan sebagainya.

A. Compiler

Compiler adalah suatu program yang menerjemahkan bahasa program (*source code*) kedalam bahasa objek (*obyek code*). Compiler menggabungkan keseluruhan bahasa program, mengumpulkannya dan kemudian menyusunnya kembali.

B. Interpreter

Interpreter adalah Perangkat lunak yang mampu mengeksekusi code program (yang ditulis oleh programmer) lalu menterjemahkannya ke dalam bahasa mesin, sehingga mesin melakukan instruksi yang diminta oleh programmer tersebut. Perintah-perintah yang dibuat oleh programmer tersebut dieksekusi baris demi baris, sambil mengikuti logika yang terdapat di dalam kode tersebut.

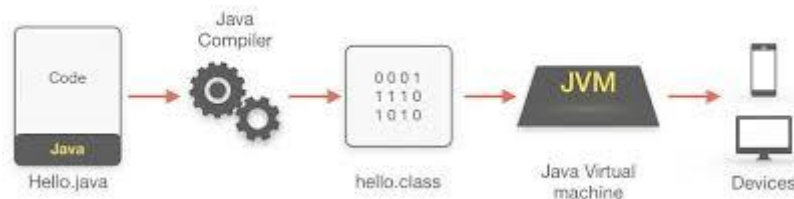
1.3 Sebuah Aplikasi

Aplikasi dengan teknologi Java secara umum adalah aplikasi serba guna yang dapat dijalankan pada seluruh mesin yang memiliki *Java Runtime Environment* (JRE).

Java Runtime Environment (JRE) adalah satu set alat perangkat lunak untuk pengembangan aplikasi Java. Ini menggabungkan Java Virtual Machine (JVM), kelas platform inti dan mendukung library. JRE merupakan bagian dari Java Development Kit (JDK), tapi dapat didownload secara terpisah. JRE pada awalnya dikembangkan oleh Sun Microsystems Inc., anak perusahaan yang sepenuhnya dimiliki oleh Oracle Corporation. Juga dikenal sebagai runtime Java.

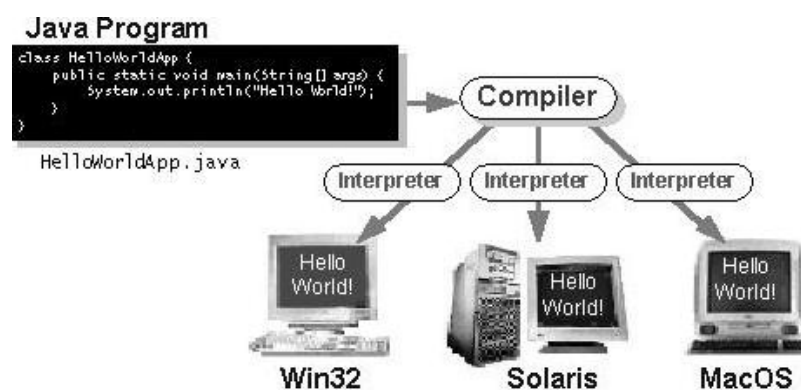
2. Cara Kerja Pemrograman JAVA

Kode pada bahasa pemrograman Java dituliskan dalam file teks dan disimpan dengan nama file yang sama persis dengan nama classnya serta diberikan ekstensi java (*.java). File tersebut kemudian di kompilasi dengan compiler java (javac) sehingga akan menghasilkan 1 file dengan ekstensi class (*.class). File berekstensi class (*.class) yang akan dijalankan diatas Java Virtual Machine (JVM) dengan menggunakan java launcher tool (java). Jika digambarkan akan seperti gambar di bawah ini :



Gambar1 : cara kerja pemrograman java

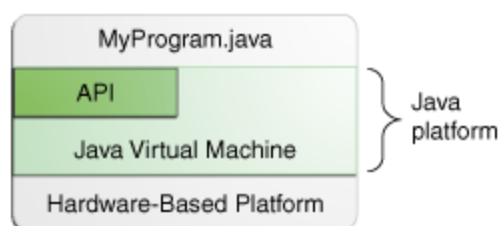
Adanya Java Virtual Machine (JVM), program java yang telah dibuat dapat dijalankan pada berbagai komputer dengan arsitektur dan Sistem Operasi yang berbeda.



Gambar2 : Multiplatform (dapat dijalankan di semua sistem operasi)

Platform java memiliki dua komponen, yaitu :

1. Java Virtual Machine (JVM) merupakan dasar bagi platform Java yang berjalan di atas sistem operasi.
2. Java Application Programming Interface (API) merupakan kumpulan kode java siap pakai untuk berbagai keperluan seperti untuk User Interface, networking dan lain sebagainya.



Gambar3 : Platform Java

3. *Perbedaan Pemrograman Prosedural dan Pemrograman Berorientasi Object*

3.1 *Pemrograman Prosedural (Terstruktur)*

Pemrograman Prosedural merupakan suatu proses untuk mengimplementasikan urutan langkah penyelesaian suatu masalah dalam bentuk program dan merupakan suatu aktifitas pemrograman yang dilakukan dengan memperhatikan setiap urutan dari setiap langkah perintah yang dikerjakan secara sistematis, logis, dan tersusun berdasarkan algoritma yang sederhana dan dapat dengan mudah dipahami. *Prinsip dari pemrograman prosedural* adalah Jika suatu proses telah sampai pada suatu titik ataupun langkah tertentu, maka proses selanjutnya tidak boleh mengeksekusi langkah sebelumnya ataupun kembali ke baris sebelumnya, kecuali pada langkah - langkah untuk proses pengulangan atau berulang (Loop). Bahasa pemrograman yang mendukung pemrograman prosedural, antara lain : Cobol, Basic, Pascal, Fortran, dan C.

3.2 *Pemrograman Berorientasi Object*

Pemrograman Berorientasi Objek (PBO) atau dalam bahasa inggris disebut *Object Oriented Programming (OOP)* merupakan sebuah paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalamnya dibungkus dalam suatu kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

3.3 *Perbedaan mendasar*

Perbedaan mendasar antara Pemrograman Berorientasi Objek dan Pemrograman Prosedural adalah dengan menggunakan Pemrograman Berorientasi Objek maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sedangkan untuk Pemrograman Terstruktur, menggunakan prosedur/tata cara yang teratur untuk mengoperasikan data struktur.

Kalau di prosedural ada "*function*", di OOP ada "*method*". Kalau di prosedural ada "*modules*", di OOP ada "*objects*". Kalau di prosedural ada "*argument*", di OOP ada "*message*". Begitu juga dengan "*variabel*" yang ada di prosedural, di OOP lebih dikenal dengan nama "*atribut*".

Tabel 1. Perbandingan Pemrograman Prosedural dengan Berorientasi Object

Pemrograman Prosedural	Pemrograman Berorientasi Obejct
Memecah program dalam fungsi dan data	Menggabungkan fungsi dan data dalam kelas - kelas atau objek - objek
Memiliki ciri Sequence (berurutan), Selection (pemilihan) dan Repetition (perulangan)	Memiliki ciri Encapsulation (pengemasan), Inheritance (penurunan sifat) dan

	Polymorphism (perbedaan bentuk dan perilaku)
Struktur program rumit karena berupa urutan proses dan fungsi-fungsi	Struktur program ringkas, cukup dengan membuat Objek dan class lalu bekerja berdasarkan object dan class tersebut.
Re-use kode program kurang	Kode program sangat re-usable. object dan class dapat digunakan berkali-kali, sehingga dapat menghemat space memori.
Efektif digunakan untuk menyelesaikan masalah kecil dan tidak cocok untuk menyelesaikan masalah yang rumit, karena nantinya akan kesulitan menemukan solusi permasalahan ketika terjadi error	Efektif digunakan untuk menyelesaikan masalah besar, karena OOP terdiri dari class-class yang memisahkan setiap kode program menjadi kelompok - kelompok kecil, sesuai dengan fungsinya
Mudah diawal, namun kompleks diproses selanjutnya	Sulit diawal (karena harus membuat class) namun selanjutnya akan terasa mudah dan cepat
Eksekusi lebih lambat karena setiap perintah dikerjakan berurutan	Eksekusi lebih cepat karena dieksekusi bersamaan, program hanya mengatur Objek, properties dan method-nya saja
Efektif digunakan untuk menyelesaikan masalah kecil dan tidak cocok untuk menyelesaikan masalah yang rumit, karena nantinya akan kesulitan menemukan solusi permasalahan ketika terjadi error	Efektif digunakan untuk menyelesaikan masalah besar, karena OOP terdiri dari class-class yang memisahkan setiap kode program menjadi kelompok - kelompok kecil, sesuai dengan fungsinya

Pada dasarnya Pemrograman Berorientasi Objek maupun Prosedural tidak ada yang dapat dikatakan lebih baik karena keduanya memiliki spesifikasi tersendiri dalam pemrogramannya. Pemrograman prosedural fokus pada bagaimana cara komputer menangani masalah, sedangkan Pemrograman Berorientasi Objek fokus pada masalah yang ditangani dengan menggunakan komputer. Pemrograman prosedural biasanya digunakan untuk membuat program yang simpel/ sederhana karena masih mudah dan tidak banyak dilakukan perubahan yang berarti, sedangkan Pemrograman Berorientasi Objek cocok digunakan untuk pembuatan software yang rumit dan kompleks karena memberikan berbagai kemudahan kepada programmer seperti yang telah disebutkan diatas. Pemrograman Berorientasi Objek lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan Pemrograman Berorientasi Objek lebih mudah dikembangkan dan dirawat.

4. JAVA IDE (Integrated Development Environment) : Jcreator, Eclipse, Netbeans

Untuk membuat sebuah program Java, minimal kita membutuhkan dua buah *software*, yaitu sebagai berikut :

A. Java 2 SDK, Standard Edition (J2SE)

J2SE atau *Java 2 Standard Edition* merupakan bahasa pemrograman Java untuk aplikasi desktop yang merupakan *object-oriented programming*. Pada J2SE, terdiri dari dua buah produk yang dikeluarkan untuk membantu dalam membuat aplikasi tanpa tergantung dari *platform* yang digunakan, yaitu :

- **Java SE Runtime Environment (JRE)**

Java Runtime Environment (JRE) menyediakan perpustakaan, *Java Virtual Machine* (JVM), dan komponen lain untuk menjalankan *applet* dan aplikasi yang ditulis dengan bahasa pemrograman Java. Selain itu, terdapat dua buah kunci teknologi yang merupakan bagian JRE, yaitu : Java Plug-in, yang memungkinkan menjalankan *applet* di browser populer dan Java Web Start, yang menyebarkan aplikasi mandiri melalui jaringan. JRE tidak mengandung utilitas seperti *compiler* atau *debugger* untuk mengembangkan *applet* dan aplikasi.

<https://www.oracle.com/java/technologies/javase-downloads.html>

- **Java Development Kit (JDK)**

Java Development Kit (JDK) merupakan perangkat lunak yang digunakan untuk manajemen dan membangun berbagai aplikasi Java. JDK merupakan superset dari JRE, berisikan segala sesuatu yang ada di JRE ditambahkan *compiler* dan *debugger* yang diperlukan untuk membangun aplikasi.

<https://www.oracle.com/java/technologies/javase-downloads.html>

B. IDE (Integrated Development Environment) JAVA

IDE atau *Integrated Development Environment* adalah suatu software aplikasi yang menyediakan fasilitas komprehensif bagi para programmer komputer termasuk Java untuk proses pengembangan software. Secara standar IDE ini memiliki :

- Editor untuk source code
- Debugger
- Build automation

IDE dapat memberikan kemudahan dan kecepatan bagi programmer. Terdapat banyak IDE dan terkadang memilih mana yang akan digunakan cukup menyita waktu. Adapun beberapa IDE JAVA yang populer digunakan dalam pemrograman JAVA.

a. Jcreator

JCreator yaitu program Java yang berdesign interface dimana kita bisa membuat aplikasi, project serta program Java yang kita inginkan. JCreator juga dapat digunakan untuk menjalankan aplikasi Java dengan menu lebih banyak dan mudah dalam penggunaannya, seperti menu compile, run dan lain-lain. Pada program ini dapat digunakan untuk membuat project template.

JCreator terdapat dua versi yaitu *JCreator Pro* dan *JCreator LE*. Perbedaannya terletak pada lisensi yaitu *JCreator Pro* harus membayar dalam batas waktu dua bulan, sedangkan *JCreator LE* merupakan *compiler* gratis, yang berbasis *open source*.

<http://www.jcreator.org/download.htm>

b. Eclipse

Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform independent). <https://www.eclipse.org/>

Berikut ini adalah sifat dari Eclipse:

1. *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. *Mult-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi. Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak seperti dokumentasi, pengujian perangkat lunak, pengembangan web, dan lain sebagainya.

c. NetBeans

Netbeans adalah sebuah aplikasi IDE (Integrated Development Environment) buatan SUN Microsystems. aplikasi ini berbasiskan Java dan berjalan di atas swing. Swing sendiri adalah sebuah teknologi Java untuk mengembangkan sebuah aplikasi Desktop yang dapat dijalankan di berbagai macam OS. yaitu Windows, Linux, MAC OS, Solaris dan OS lain yang mendukung suatu JVM yang sepadan.

Pada awalnya, aplikasi ini diperuntukkan dalam suatu pengembangan pemrograman Java. Namun, aplikasi ini juga mendukung bahasa pemrograman lainnya, secara khusus seperti PHP, C/C++ dan HTML5.

<https://netbeans.org/downloads/6.1/index.html>

d. BlueJ

BlueJ adalah IDE khusus yang dibuat sebagai media pengajaran bahasa pemrograman Java. IDE ini dikembangkan oleh Michael Kölling pada tahun 1999 dan terus dikembangkan sampai sekarang berkat dukungan dari Sun Microsystem (sekarang Oracle) dan University of Kent. BlueJ juga merupakan aplikasi *open source* dibawah lisensi GNU GPL versi 2. BlueJ dikembangkan untuk mendukung pembelajaran dan pengajaran pemrograman berorientasi objek , dan desainnya berbeda dari lingkungan pengembangan lainnya.

<https://www.bluej.org/>

e. Borland Jbuilder

JBuilder® adalah salah satu Java IDE paling lengkap dan power full yang tersedia dengan dukungan untuk server aplikasi Java EE 5 komersial dan open source terkemuka. JBuilder memprofilkan kode dan alat tuning kinerja, yang mencakup kemampuan untuk profil aplikasi Web, memungkinkan pengembang Java untuk memberikan aplikasi yang berkinerja tinggi dan dapat diskalakan.

<https://borland-jbuilder.software.informer.com/10.0/>

f. IntelliJ IDEA

IntelliJ IDE merupakan sebuah IDE yang dikembangkan oleh JetBrains. IntelliJ IDEA tersedia dalam 2 (dua) versi, yaitu versi Community Edition dengan lisensi *Apache 2 Licensed* yang bisa kita gunakan secara gratis dan versi Ultimate Edition untuk penggunaan komersial.

IDE yang ditulis dengan Java dan Kotlin ini dirilis versi pertamanya pada januari 2011, dan langsung menjadi IDE pertama untuk pengembangan aplikasi berbasis Java. Ia mendukung navigasi kode tingkat lanjut dan kemampuan refactoring yang saling terintegrasi.

<https://www.jetbrains.com/idea/>

g. Dr. Java

DrJava adalah IDE Java ringan yang dirancang terutama untuk siswa dan pemula di Jawa yang secara aktif dikembangkan dan dikelola oleh grup JavaPLT di Rice University . Antarmukanya telah dikembangkan menggunakan alat bantu Sun Microsystems dan oleh karena itu ia memiliki penampilan yang konsisten pada platform yang berbeda. DrJava memiliki kemampuan untuk secara interaktif mengevaluasi kode Java dari konsol dan untuk menyajikan output juga ke konsol yang sama. Ini memiliki banyak fitur lain yang telah dirancang untuk pengguna tingkat lanjut juga. DrJava menawarkan fasilitas uji JUnit.

<http://www.drjava.org/download.shtml>

h. jGRASP

jGRASP adalah IDE yang ringan. Terdapat fitur otomatis dalam memvisualisasikan coding. IDE ini dikembangkan dalam Java. Kompatibel untuk berbagai platform. Visualisasi lain adalah control structure diagram (CSD), profil graph, UML dan lain-lain.

<https://www.jgrasp.org/>

i. Oracle Jdeveloper

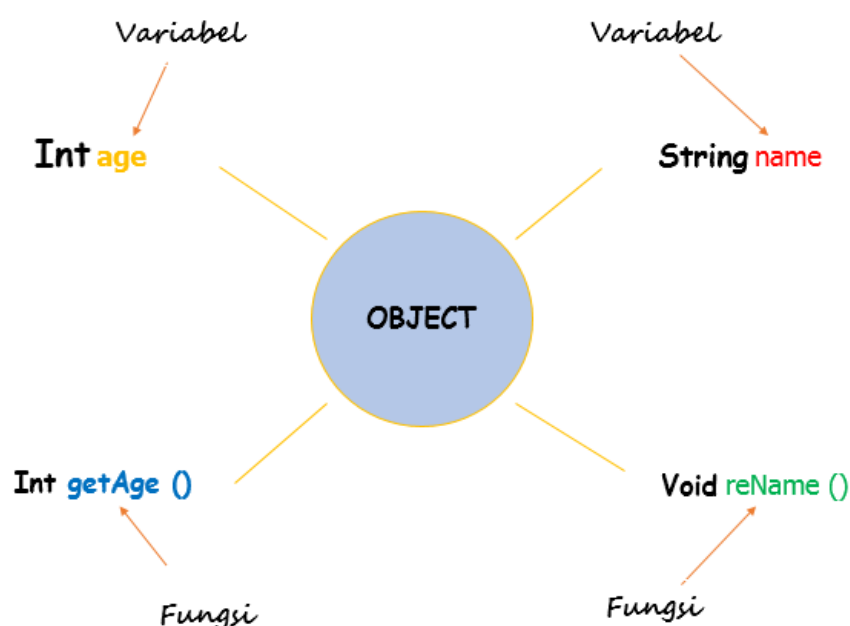
Oracle JDeveloper adalah IDE yang gratis. Cocok untuk pengembangan aplikasi berbasis SOA dan Java EE. IDE ini juga menawarkan pengembangan end-to-end lengkap untuk Oracle Fusion Middleware dan Oracle Fusion Apps dengan dukungan pengembangan lengkap dan siklus hidup aplikasi.

Analisis : Semua program IDE untuk Java sama-sama memiliki keunggulan khusus tersendiri, namun bukan itu yang terpenting. Yang terpenting adalah The Brainware, seseorang yang mampu dan memiliki keahlian untuk menjalankan program-program tersebut. Dialah yang dapat menggunakan program-program tersebut sesuai dengan kebutuhannya dan arahnya.

<https://www.oracle.com/tools/downloads/Jdeveloper-12c-downloads.html>

5. Konsep Pemrograman Berorientasi Object

Pemrograman berorientasi object atau yang lazim dikenal dengan sebutan asing -*Object Oriented Programming (OOP)* merupakan sebuah paradigma atau teknik pemrograman yang berorientasikan Objek. Pada OOP, Fungsi dan variabel dibungkus dalam sebuah objek atau *class* yang dapat saling berinteraksi, sehingga membentuk sebuah program.



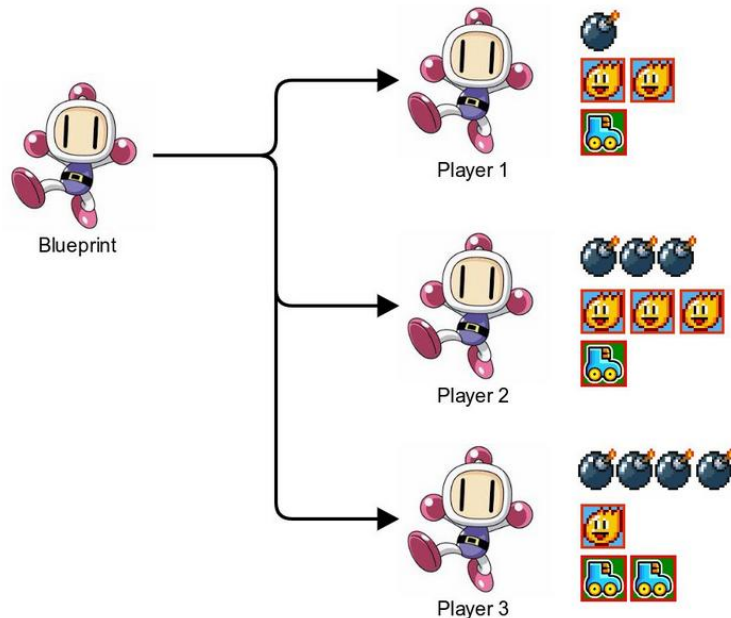
Gambar 4. Gambaran pemrograman berorientasi object

Dengan demikian, tidak akan ada lagi kode yang “berantakan.” Semua akan kita bungkus dalam objek. Tapi sebelum membuat objek, kita harus membuat class.

A. Class

Class adalah sebuah *prototype* atau *blueprint* (rancangan) dari object. Sebuah class mendefinisikan *state* dan *behavior* yang dirancang. Kemudian dari class tersebut dapat dibuat *instance* atau *object* nya.

Lihat ilustrasi berikut ini:



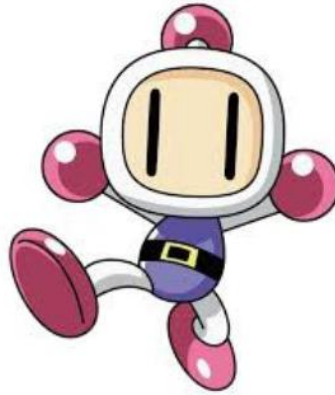
Gambar 5 ilustrasi tentang class

Dari sebuah class merupakan Blueprint, kita dapat membuat beberapa objek. Objek akan memiliki *state* dan *behavior* seperti yang ditentukan oleh blueprintnya. Namun setiap object dapat memiliki state yang berbeda-beda, misalkan jumlah bom yang berbeda, jumlah fire yang berbeda, dan speed sepatu yang berbeda-beda.

B. Object

Object adalah suatu bagian dari software yang mencatat *state* dan *behavior*. Yang dimaksud dengan *state* adalah data atau di Java sering disebut sebagai variabel, yang menyimpan suatu nilai. Sedangkan *behavior* adalah fungsi apa yang bisa dilakukan oleh *Object* tersebut.

Sebagai contoh dalam game Bomberman, apa yang menjadi *state*, dan apa yang menjadi *behavior*.



Object example

Gambar 6 ilustrasi object

State:

Apa yang dimiliki oleh character bomberman ?

- Jumlah nyawa (integer)
- Jumlah bom yang bisa dikeluarkan maksimal (integer)
- Jumlah bom yang sedang dikeluarkan (integer)
- Apakah bisa menendang bom ? (boolean)
- Apakah bisa melempar bom ? (boolean)
- Apakah keracunan ? (boolean)
- Berapa jumlah panjang ledakan ? (integer)
- Apakah mendapatkan bom tipe maksimal ledakan ? (boolean)
- Berapa speed/kecepatan gerak ? (integer)

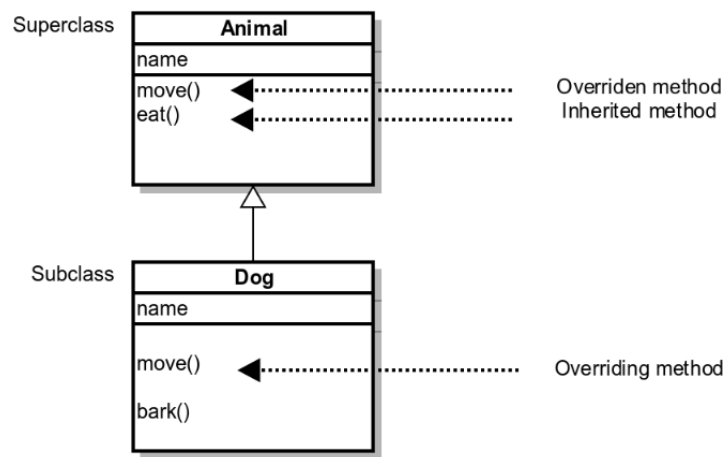
Behavior:

Apa yang bisa dilakukan oleh character bomberman ?

- *Gerak ke atas*
- *Gerak ke kiri*
- *Gerak ke kanan*
- *Gerak ke bawah*
- *Menaruh bom*
- *Menendang*
- *Melempar*

C. Inheritance

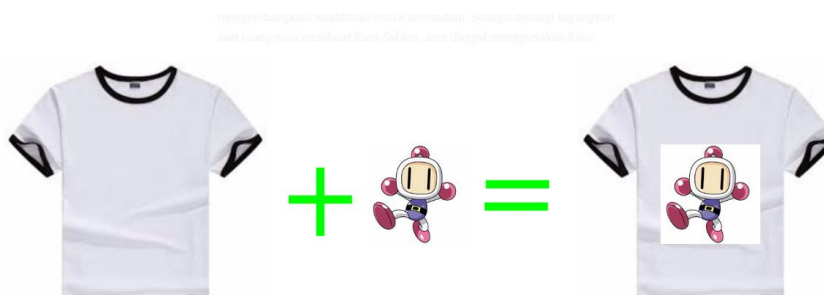
Inheritance adalah konsep di mana kita dapat membuat class baru sebagai turunan dari class yang sudah ada, kemudian kita dapat menambahkan fitur atau kemampuan lain (*state* atau *behavior*) di class turunannya.



Gambar 7 skematik konsep inheritance

Pada gambar di atas, Animal memiliki 2 method (behavior), `move()` dan `eat()`. Dog adalah class turunan dari Animal, di mana method `eat()` diturunkan ke class Dog, sehingga class Dog otomatis mempunyai method `eat()` juga tanpa harus kita coding lagi. Method `move()` dilakukan perubahan dengan membuat method `move()` yang baru untuk menyesuaikan. Sedangkan method `bark()` adalah method tambahan yang tidak dipunyai oleh Animal.

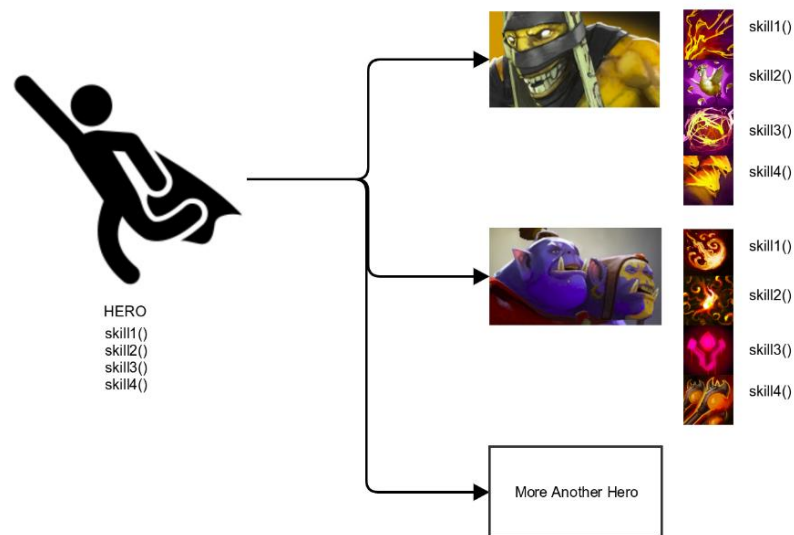
Dengan konsep di atas, kita dapat reuse code yang ada di Animal, lalu dapat mengembangkan/modifikasi sesuai kebutuhan. Sebagai analogi bayangkan saat orang mau membuat Kaos Sablon, user tinggal menggunakan Kaos polos yang sudah ada, kemudian ditambahkan Sablon, maka akan menjadi kaos yang lain.



Gambar 8 konsep inheritance

D. Interface

Interface adalah sarana standarisasi antara class dengan dunia luar dari class. Dalam interface akan memiliki method kosong tanpa implementasi, namun membuat suatu standard, saat class melakukan implementasi terhadap interface tersebut maka class akan harus memenuhi method-method yang sudah secara standard ditentukan oleh interface tersebut. Sebagai contoh misalkan kita memiliki sebuah class bernama Hero, di mana memiliki method, `jurus1`, `jurus2`, `jurus3`, dan `jurus4`, namun method-method ini merupakan method kosong yang belum ada implementasi. Lihat ilustrasi berikut ini:



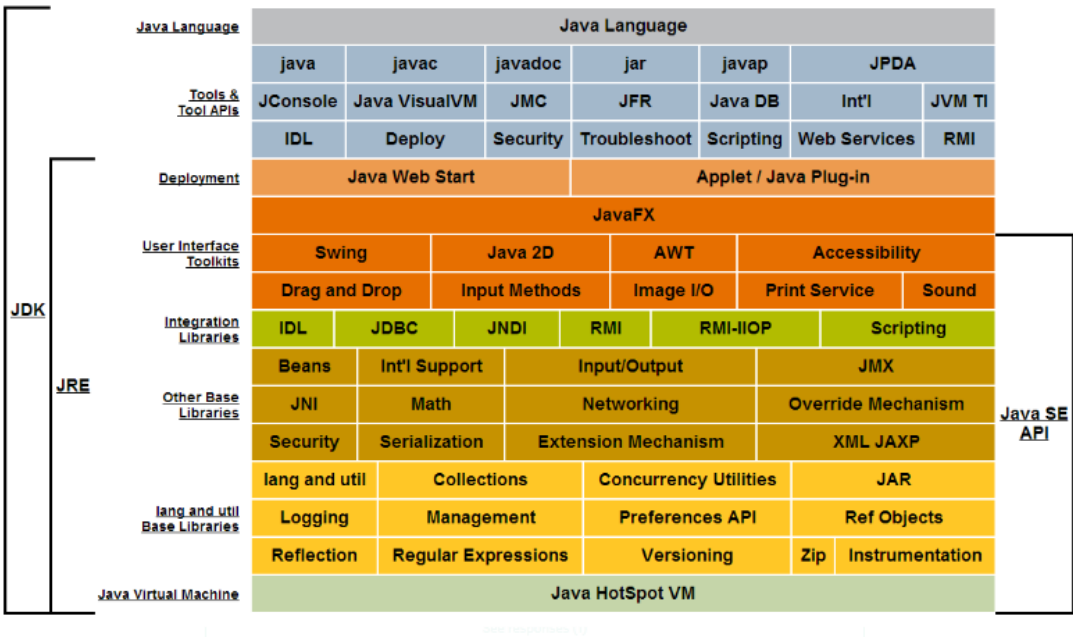
Gambar 9 Konsep interface

E. Package

Package adalah sebuah *namespace* di mana digunakan untuk mengelompokkan class dan interface. Dengan mengelompokkan class/interface ke dalam package-package sangat berguna untuk mengatur lebih mudah.

Beberapa contoh:

- Swing – adalah package untuk membuat program dengan Graphical User Interface
- Image I/O – digunakan manipulasi image, resize, transform, dan sebagainya
- Print Service – package untuk kebutuhan printing
- JDBC – package untuk konektivitas ke database
- XML JAXP – package untuk XML processing



Gambar 10 package java



Tugas

Catatan.

- a. Tugas dikumpulkan maks pada hari Jum'at tgl 03-04-2020 pukul 16.00
- b. Tugas boleh dikumpulkan dalam bentuk (PDF,PPT,WORD)
- c. File dikirim ke email : fahmi03031995@gmail.com
- d. Format file dengan tata cara: NamaMhs_NIM_PBO

Soal

1. Buatlah resume tentang sejarah singkat lahirnya pemrograman JAVA
2. Apa yang dimaksud dengan dengan compiler dan interpreter jelaskan konsep kerjanya serta alur proses kerjanya
3. Jelaskan apa yang dimaksud dengan pemrograman berorientasi object serta konsep kerjanya (* penjelesan gunakan bahasa sendiri apabila mengutip referensi cantumkan kutipanya)
4. Sebutkan dan Jelaskan apa saja yang diperlukan dalam proses pembuatan pemrograman JAVA