

JavaApplet

Sumber : Buku Dasar Pemrograman Java™ 2
karangan Abdul Kadir &
Dasar-dasar Pemrograman Java di
<http://www.ilmukomputer.com>

Pengenalan Java

Java dikembangkan oleh Sun Microsystems pada Agustus 1991, dengan nama semula Oak. Pada Januari 1995, diganti menjadi Java.

Java dikenal sebagai bahasa yang menarik karena mendukung sumber daya Internet yang saat ini populer, yaitu World Wide Web. Java juga mendukung aplikasi client/server, baik dalam jaringan lokal (LAN) maupun jaringan berskala luas (WAN).

Java merupakan hasil perpaduan sifat dari sejumlah bahasa pemrograman yaitu C, C++, Object-C, SmallTalk dan Common LISP.

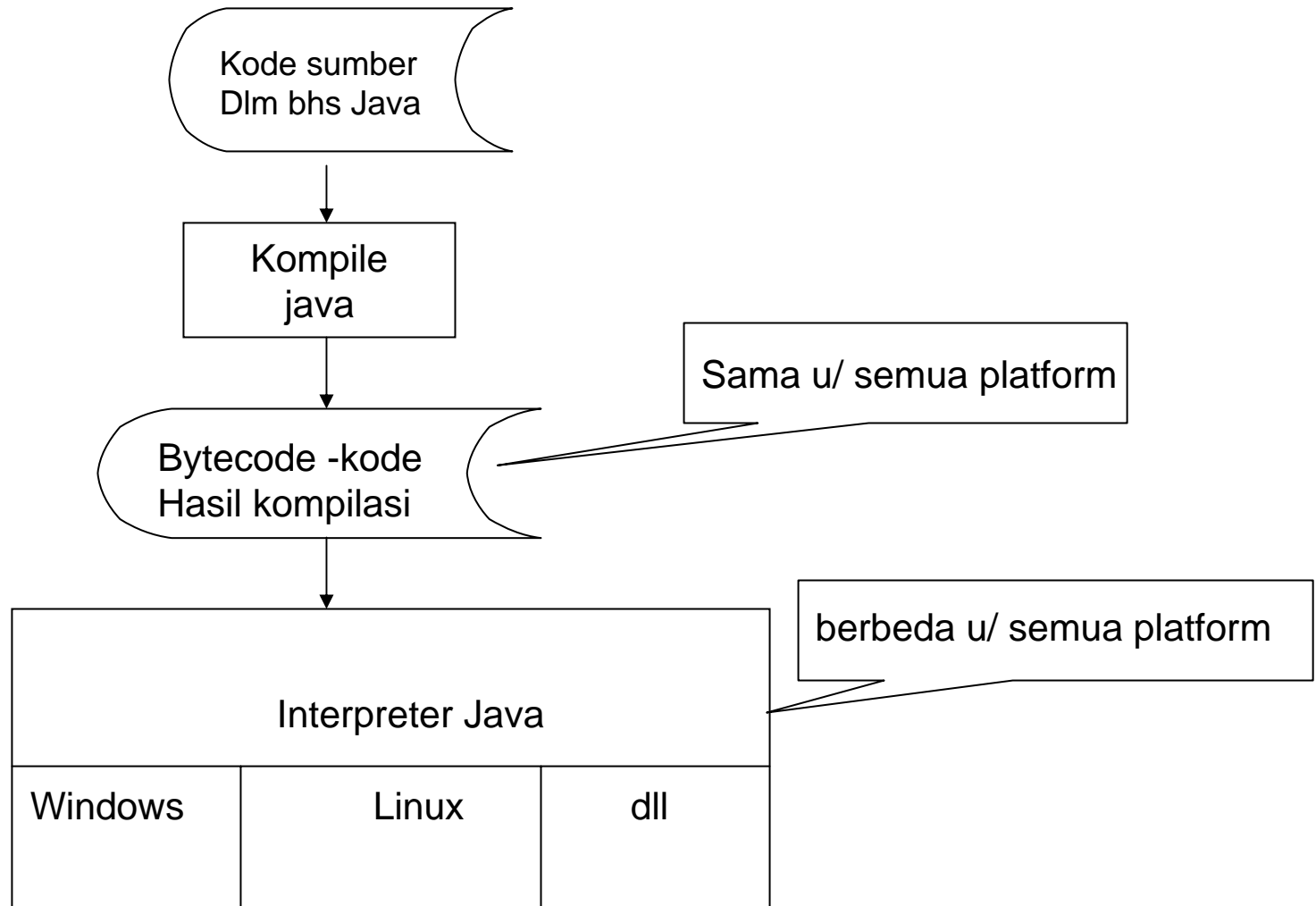
Program Java bersifat tidak bergantung pada platform.

Java merupakan bahasa pemrograman berorientasi objek, maka Java menggunakan class untuk membentuk suatu objek.

Program Java dapat dibedakan menjadi 2 jenis, yaitu:

- Applet, program dibuat dengan Java, dapat diletakkan pada Web server dan diakses melalui Web browser.
- Aplikasi, program yang dibuat dengan Java yang bersifat umum. Aplikasi dapat dijalankan secara langsung, tidak perlu browser.

Skema kompilasi - eksekusi



Susunan Kode Sumber .java

Dalam mengembangkan program Java, Anda menulis class dan class. Class ini ditulis dalam kode sumber yang disimpan sebagai file teks biasa berekstension .java.

Dalam file .java, dapat dideklarasikan :

- 1). package
- 2). import
- 3). satu atau lebih class

Deklarasi Package

Deklarasi package digunakan untuk mengelompokkan class-class. Sebuah package dapat mempunyai satu atau lebih sub-package ini sehingga dapat menyusun sebuah hirarki.

Tata cara penulisan :

```
package <namapackage>.<namasubpackage>.<namasubsubpackage>;
```

Susunan Kode Sumber .java

Contoh deklarasi package adalah :

```
package java.awt; atau package org.apache.tomcat.core;
```

Deklarasi package bersifat opsional, tidak harus ada. Tanpa deklarasi package maka sebuah class dikelompokkan ke dalam default package.

Deklarasi Import

Deklarasi import, bersifat opsional, digunakan untuk menunjukkan package atau class yg digunakan dalam sebuah program Java.

Contohnya adalah :

```
import java.io.RandomAccessFile; import java.net.*;
```

Deklarasi import tidak akan menjadikan class atau package yang Anda cantumkan digabungkan dengan program Anda saat kompilasi atau saat diluncurkan.

Susunan Kode Sumber .java

Deklarasi import digunakan untuk menunjukkan java atau javac dalam menemukan class yang Anda gunakan.

Deklarasi Class

Deklarasi class merupakan kandungan utama sebuah file berekstension .java.

Deklarasi class terutama memuat :

1. nama class, bisa dilengkapi dengan kendali akses, deklarasi extends maupun deklarasi implements. Pola yang lumrah adalah :
2. public class ClassName
3. extends SuperClassName
4. implements Interface1Name, Interface2Name, Interface3Name
5. constructor, yang dipanggil pada saat dibuat instans dari class.
6. deklarasi variabel-variabel
7. deklarasi prosedur-prosedur, yang di dalam Java disebut method.

Menulis, Menampilkan, & Menjalankan Aplikasi

- Tulis program dengan menggunakan editor teks yang anda sukai, simpan dengan *namafile.java*, misal : SelamatBelajar.java

// Berkas: SelamatBelajar.java

```
public class SelamatBelajar {  
    public static void main(String[ ] args) {  
        System.out.println("Selamat Belajar Java");  
    }  
}
```

- Kompilasi program tsb dengan cara :

```
javac SelamatBelajar.java
```

Maka akan terbentuk class → SelamatBelajar.class

- Jalankan program dengan mengetik :

```
java SelamatBelajar
```

maka akan muncul di layar : Selamat Belajar Java

Menulis, Menampilkan, & Menjalankan Applet

Langkah 1 : Tulislah kode sumber

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda.

```
import java.applet.*;
import java.awt.*;
public class SalamKeadilanApplet extends Applet
{
    public void paint(Graphics g)
    {
        Font font = new Font("Courier", Font.BOLD, 20);
        g.setFont(font);
        g.drawString("Salam keadilan !", 20, 30);
    }
}
```


Menulis, Menampilkan, & Menjalankan Applet

Langkah 2 : Lakukan compilation dengan javac

```
javac SalamKeadilanApplet.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file SalamKeadilanApplet.class di dalam directory yang sama.

Langkah 3 : Luncurkan text editor lainnya untuk menulis file HTML

Untuk memuat SalamKeadilanApplet, digunakan tag `<applet>` `</applet>`

```
<html>
```

```
<body>
```

```
  <APPLET code="SalamKeadilanApplet"
```

```
    width="400"
```

```
    height="100">
```

```
  </APPLET>
```

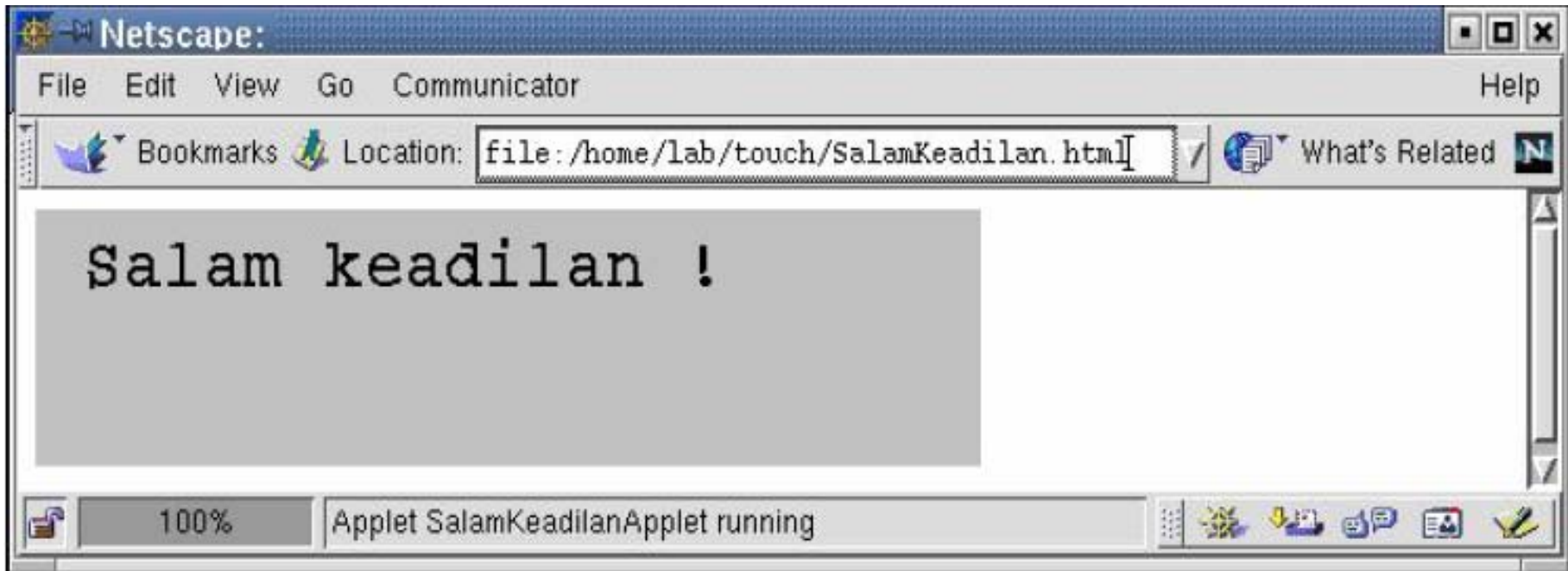
```
</body>
```

```
</html>
```

Menulis, Menampilkan, & Menjalankan Applet

Langkah 4 : Jalankan di atas browser

Untuk menjalankannya, bukalah file SalamKeadilan.html dari Web browser.



Variable

Di dalam bahasa pemrograman Java, sebuah variable mempunyai nama, tipe dan harga.

Dalam deklarasi variable setidaknya dibutuhkan nama dan tipe.

Tata cara penulisan pertama adalah : type name;

Dengan deklarasi ini, variable belum mempunyai harga.

Contoh : `int var1; var1 = 3; String var5 = "Purwokerto";`

```
public class LatDeklarasiVariable
{ public static void main(String[] args) {
int var1; var1 = 3; int var2 = 5;
double var3 = 1.99; boolean var4 = false;
String var5= "Purwokerto";
System.out.println(var1);
System.out.println(var2);
System.out.println(var3);
System.out.println(var4);
System.out.println(var5); } }
```

```
C:\latjava> java LatDeklarasiVariable
3
5
1.99
false
Purwokerto
```

Variable

Literal string dapat pula mengandung karakter seperti newline, tab, dsb. Bahkan bisa pula mengandung tanda petik ganda.

Literal String	Keterangan
"abc\ndef"	Antara abc dan def terdapat newline
"abc\tdef"	Antara abc dan def terdapat tab
"String\"Halo\""	Menyatakan string: "Halo"

```
// Berkas: LiteralString.java
```

```
public class LiteralString {  
    public static void main(String[] args) {  
        System.out.println("abc\ndef");  
        System.out.println("abc\tdef");  
        System.out.println("\"Halo\"");  
    }  
}
```

```
C:\latjava> java LiteralString  
abc  
def  
abc      def  
"Halo"
```

Basic Mathematical Operators

* / % + - are the mathematical operators

* / % have a higher precedence than + or -

```
double myVal = a + b % d - c * d / b;
```

Is the same as:

```
double myVal = (a + (b % d)) - ((c * d) / b);
```

Statements & Blocks

Sebuah statement sederhana adalah pernyataan yang diakhiri titik koma :

```
name = "Fred";
```

Sebuah block adalah gabungan pernyataan yang ditutup dalam tanda kurung kurawal :

```
{  
    name1 = "Fred"; name2 = "Bill";  
}
```

Blocks dapat terdiri dari beberapa blocks

Flow of Control

- Java menjalankan satu statement berurutan sesuai urutan penulisan
- Beberapa statement Java adalah statement flow of control :
 - Alternation : if, if else, switch
 - Looping : for, while, do while
 - Escapes : break, continue, return

If – The Conditional Statement

- Statement if meng-evaluasi sebuah expression & jika evaluasi benar maka tindakan tertentu akan dijalankan. Misal jika nilai dari x lebih kecil dari 10, maka x sama dengan 10

```
if ( x < 10 ) x = 10;
```

- Hal ini dapat ditulis seperti :

```
if ( x < 10 )  
    x = 10;
```

- Atau, alternative lain :

```
if ( x < 10 ) { x = 10; }
```

Relational Operators

==	Equal (careful)	!=	Not equal
>=	Greater than or equal	<=	Less than or equal
>	Greater than	<	Less than

If... else

The if ... else statement evaluates an expression and performs one action if that evaluation is true or a different action if it is false.

```
if (x != oldx) {  
    System.out.print("x was changed");  
}  
else {  
    System.out.print("x is unchanged");  
}
```

Nested if ... else

```
if ( myVal > 100 ) {  
    if ( remainderOn == true) {  
        myVal = mVal % 100;  
    }  
    else {  
        myVal = myVal / 100.0;  
    }  
}  
else  
{  
    System.out.print("myVal is in range");  
}
```


Switch Statement

Tata cara penulisan switch-case adalah :

```
switch(variabel)
```

```
{
```

```
    case harga1 : pernyataan-1; break;
```

```
    case harga2 : pernyataan-2; break;
```

```
    case harga3 : pernyataan-3; break;
```

```
    default : pernyataan-default;
```

```
}
```

Contoh Switch

```
public class LatSwitchCase
{
public static void main(String[] args)
{
char inisial = 'M';
String nama = null;
switch(inisial)
{
case 'A' : nama = "Agung";break;
case 'E' : nama = "Eko";break;
case 'F' : nama = "Fikri";break;
case 'M' : nama = "Millati";break;
case 'S' : nama = "Syarif";break;
case 'T' : nama = "Toosa";break;
case 'Y' : nama = "Yudo";break;
default : nama = "Tak dikenal";
}
System.out.println("Nama Anda adalah " + nama);
}
}
```

```
$ java LatSwitchCase
Nama Anda adalah Millati
```

for loop

Loop n times

```
for ( i = 0; i < n; n++ )  
{  
    // this code body will execute n times  
    // i from 0 to n-1  
}
```

Nested for:

```
for ( j = 0; j < 10; j++ ) {  
    for ( i = 0; i < 20; i++ ){  
        // this code body will execute 200 times  
    }  
}
```

while loops

```
while(response == 1) {  
    System.out.print( "ID =" + userID[n]);  
    n++;  
    response = readInt( "Enter ");  
}
```

do {... } while loops

```
do {  
    System.out.print( "ID =" + userID[n] );  
    n++;  
    response = readInt( "Enter " );  
} while (response == 1);
```

Break

Break statement menyebabkan keluar dari innermost yg berisi while, do, for or switch statement.

```
for ( int i = 0; i < maxID, i++ ) {  
    if ( userID[i] == targetID ) {  
        index = i;  
        break;    }  
} // program jumps here after break
```

Continue

Hanya dapat digunakan dengan while, do atau for.

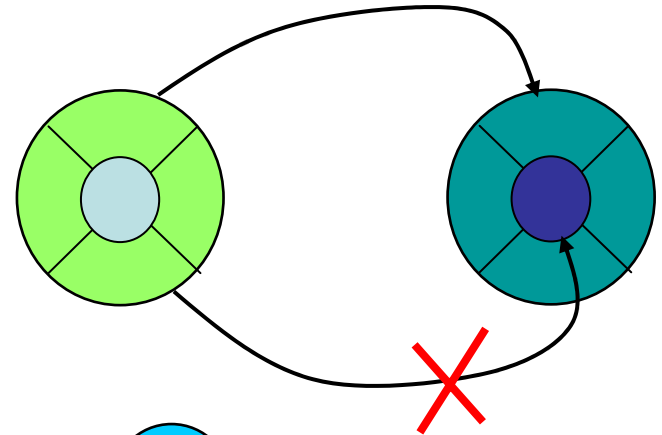
Continue statement menyebabkan innermost loop mulai perulangan berikutnya

```
for ( int i = 0; i < maxID; i++ ) {  
    if ( userID[i] != -1 ) continue;  
    System.out.print( "UserID " + i + " :" + userID);  
}
```

Tiga Prinsip OOP

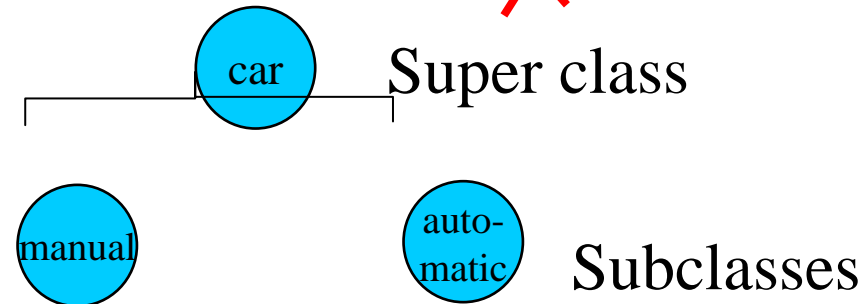
Encapsulation

- Object menyembunyikan fungsi mereka (**methods**) dan data (**instance variables**)



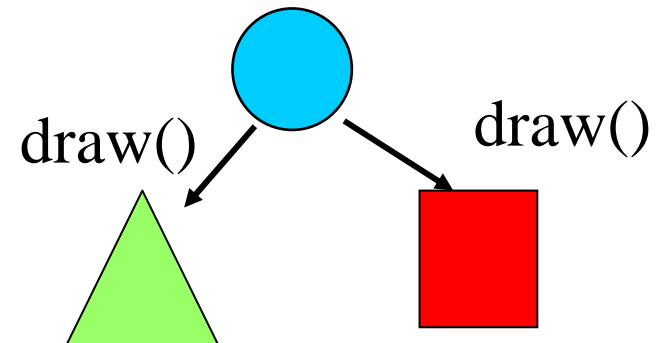
Inheritance

- Setiap **subclass** mewarisi seluruh variable dari **superclass-nya**



Polymorphism

- Interface sama menghubungkan tipe data berbeda



Methods

Suatu method adalah suatu urutan code yang diberi nama yang dapat dilibatkan oleh Java code lain

Suatu metoda mengambil beberapa parameter, melaksanakan beberapa perhitungan dan kemudian secara bebas mengembalikan nilai (atau obyek).

Methods dapat digunakan sebagai bagian dari statement expression.

```
public float convertCelsius(float tempC) {  
    return( ((tempC * 9.0f) / 5.0f) + 32.0 );  
}
```

Public/private

Methods/data dapat dideklasikan **public** atau **private** yang artinya method/data tersebut dapat atau tidak dapat diakses oleh code pada class lain ...

Good practice: - keep data private
- keep most methods private

Interface yang didefinisikan dengan baik antar class – menolong menghilangkan error

Using objects

Code pada sebuah class akan membuat sebuah instance dari class lain ...

```
Fruit plum=new Fruit();
```

```
int cal;
```

```
cal = plum.total_calories();
```

Dot operator membolehkan kita untuk mengakses (public) data/methods dalam Fruit class