

Java Fundamental

Saniati

saniati@teknokrat.ac.id

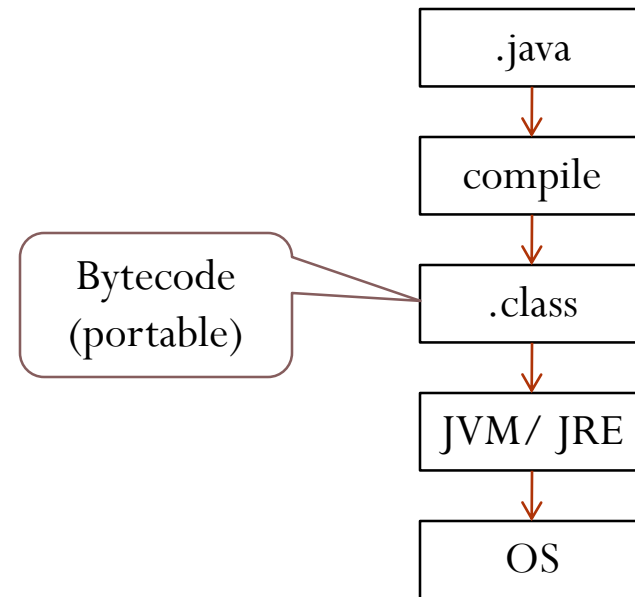
STMIK Teknokrat, Bandar Lampung

Java

- Turunan dari c/c++
- Punya moto WORA (*Write Once Run Anywhere*)
 - *Multiplatform*
 - development* : windows
 - execution* : linux/unix

Java Cycle Execution

- Kelebihan : *multiplatform*
- Kekurangan : lambat
- Keterangan:
 - JVM : Java Virtual Machine
 - JRE : Java Runtime Environment

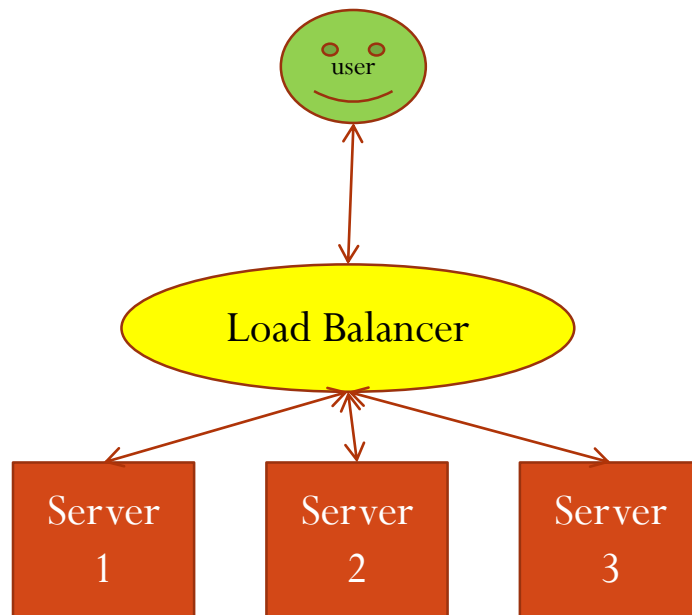


Java Edition

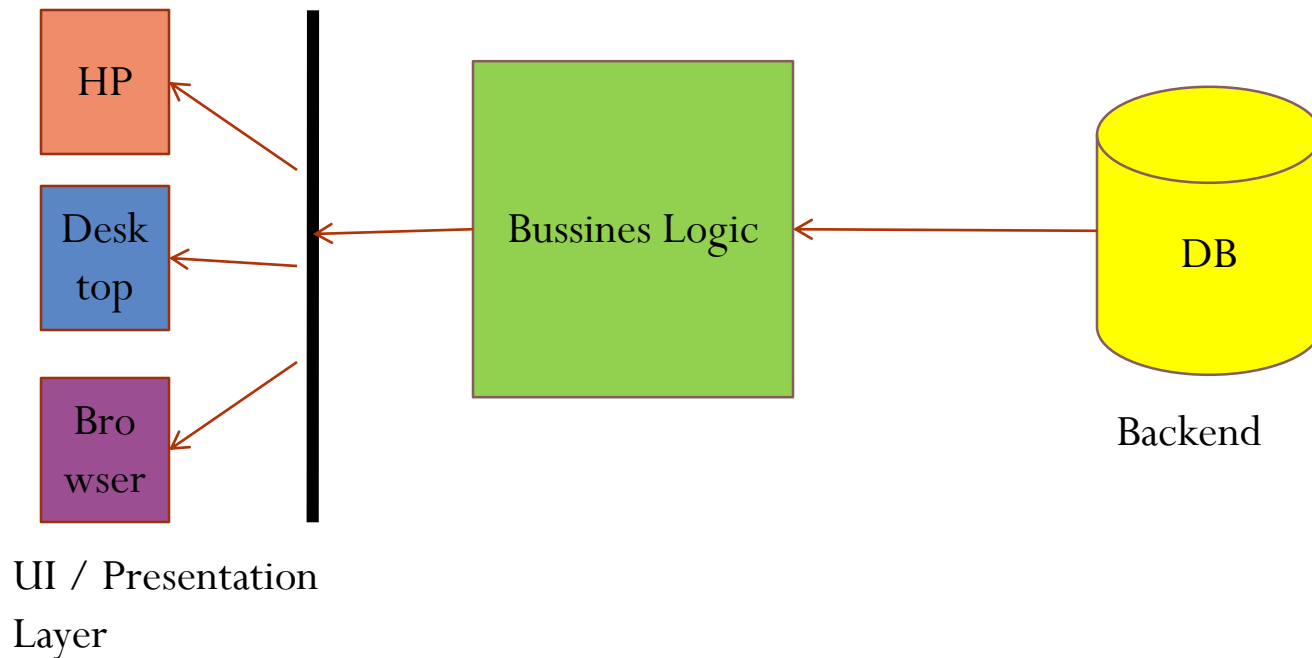
- JSE : Java Standart Edition
 - *Core technology / library*
 - Untuk membuat aplikasi *console / desktop*
- JME : Java Micro Edition
 - *Mobile apps.*
- JEE : Java Enterprise Edition
 - *Enterprise class app*
 - *Web app / web service / ejb*
 - Kelebihan, kemampuan melakukan *clustering* (High Availability Architecture/HAA) (*Distributed System*)

Clustering pada JEE

- Performance
 - Scalability (meningkatkan kapasitas processing)
 - Speed Up (mempercepat proses)
- Availability



Arsitektur Sistem Terdistribusi dari JEE



Java Development

- JDK / Java SDK (wajib)
 - JDK komersial : Oracle JRockit
- IDE (Integrated Development Environment) (tidak wajib)
 - Eclipse
 - Netbeans
 - JDeveloper (disarankan untuk JEE, *free* u/ komponen standar)
 - IntelliJ
 - Borland J Builder
 - Aspect J

Install dan Periksa JDK

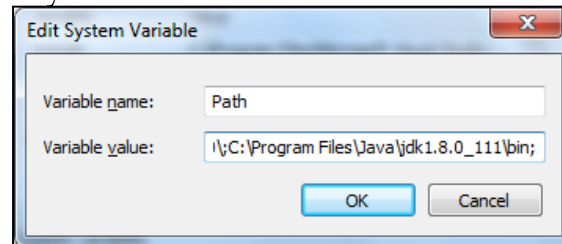
- Periksa JDK sudah terinstal atau belum dengan perintah “javac” pada *command line*.

```
C:\Users\lap1>javac
'javac' is not recognized as an internal or external command,
operable program or batch file.
```

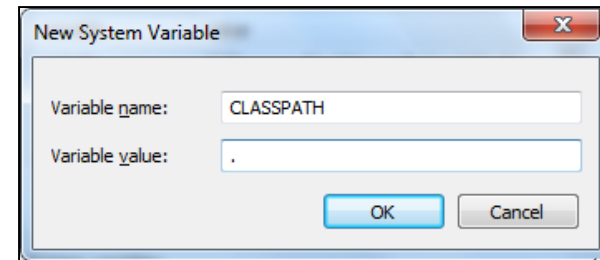
- Dua kemungkinan jika pesan tersebut tampil
 - JDK belum terinstal (solusi, *install* JDK)
 - JDK sudah terinstal, namun variable PATH belum di *setting*

Setting PATH

- Jika menggunakan JDK, harus ada 2 environment variable yang di set:
 1. PATH : diarahkan ke lokasi instalasi folder bin JDK. Tujuannya agar java compiler bisa dijalankan dari mana saja.
 - Cari folder bin dari jdk yg telah diinstall, copy address “C:\Program Files\Java\jdk1.8.0_111\bin”
 - Klik kanan My Computer, pilih Advance System Setting, pilih button Environment Variable, pada System Variable isi Variable Path dengan value address diatas.



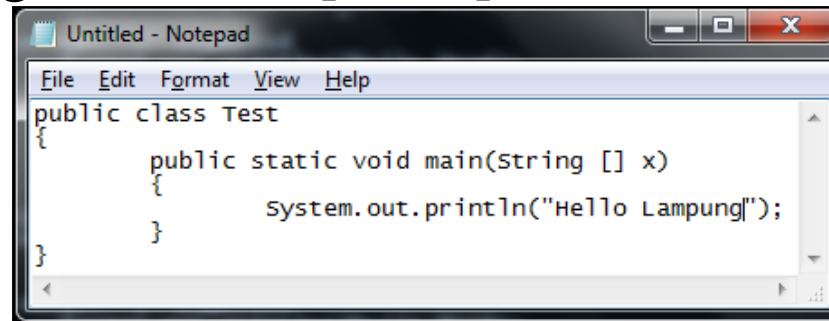
2. CLASSPATH : diarahkan ke *current* directori (.). Tujuannya agar bisa menjalankan hasil coding dari mana saja. Jika belum ada variabel path, buat dengan pilih New.



Program Pertama

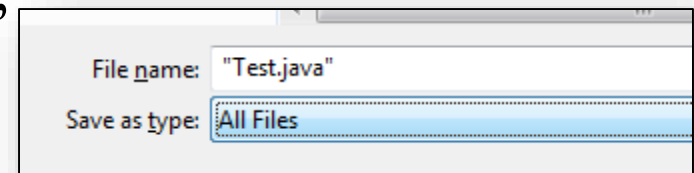
Java case sensitive

- Buat program di notepad seperti berikut:



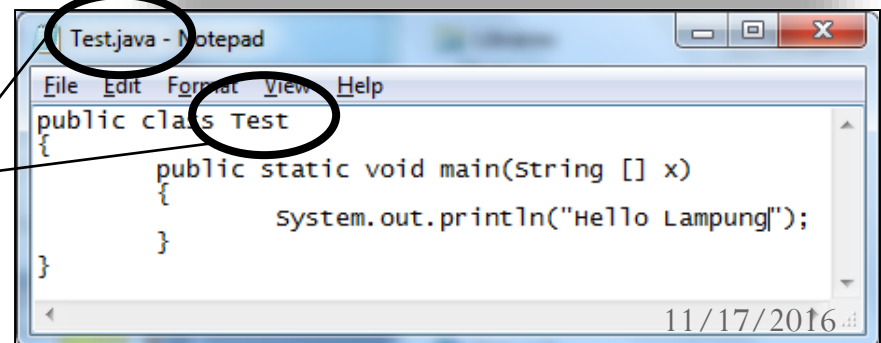
```
Untitled - Notepad
File Edit Format View Help
public class Test
{
    public static void main(String [] x)
    {
        System.out.println("Hello Lampung");
    }
}
```

- Lalu *save* dengan nama file yang sama dengan nama class, dengan ekstensi .java. “Test.java”



- Hasil seperti file berikut

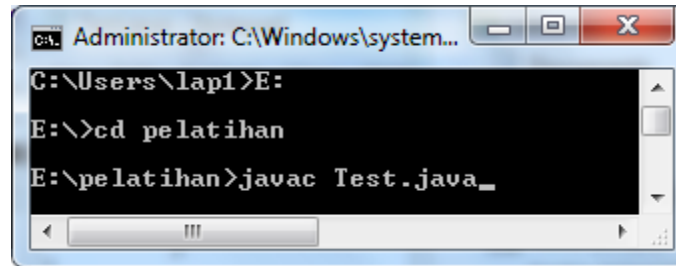
Nama file dan nama class sama



```
Test.java - Notepad
File Edit Format View Help
public class Test
{
    public static void main(String [] x)
    {
        System.out.println("Hello Lampung");
    }
}
```

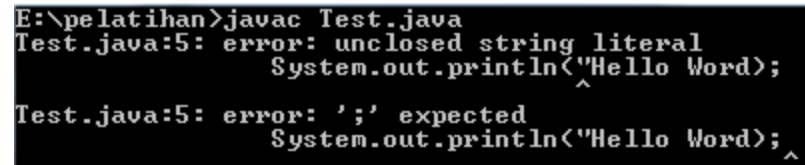
Compile

- Untuk *compile* melalui *command prompt* masuk ke direktori tempat penyimpanan file .java. Lalu jalankan perintah “javac NamaFile.java”



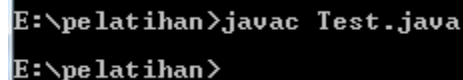
```
Administrator: C:\Windows\system...
C:\Users\lap1>E:
E:\>cd pelatihan
E:\pelatihan>javac Test.java
```

- Jika *error*, tampil pesan *error* seperti berikut:



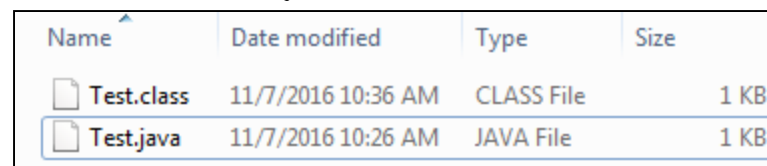
```
E:\pelatihan>javac Test.java
Test.java:5: error: unclosed string literal
    System.out.println("Hello Word);
                        ^
Test.java:5: error: ';' expected
    System.out.println("Hello Word);
                        ^
```

- Jika tidak ada *error*, maka akan tampil seperti berikut:



```
E:\pelatihan>javac Test.java
E:\pelatihan>
```

- Jika berhasil di direktori penyimpanan akan bertambah file .class



Name	Date modified	Type	Size
Test.class	11/7/2016 10:36 AM	CLASS File	1 KB
Test.java	11/7/2016 10:26 AM	JAVA File	1 KB

Run

- Untuk **run** melalui *command prompt* masuk ke direktori tempat penyimpanan file .class. Lalu jalankan perintah “java NamaFile”

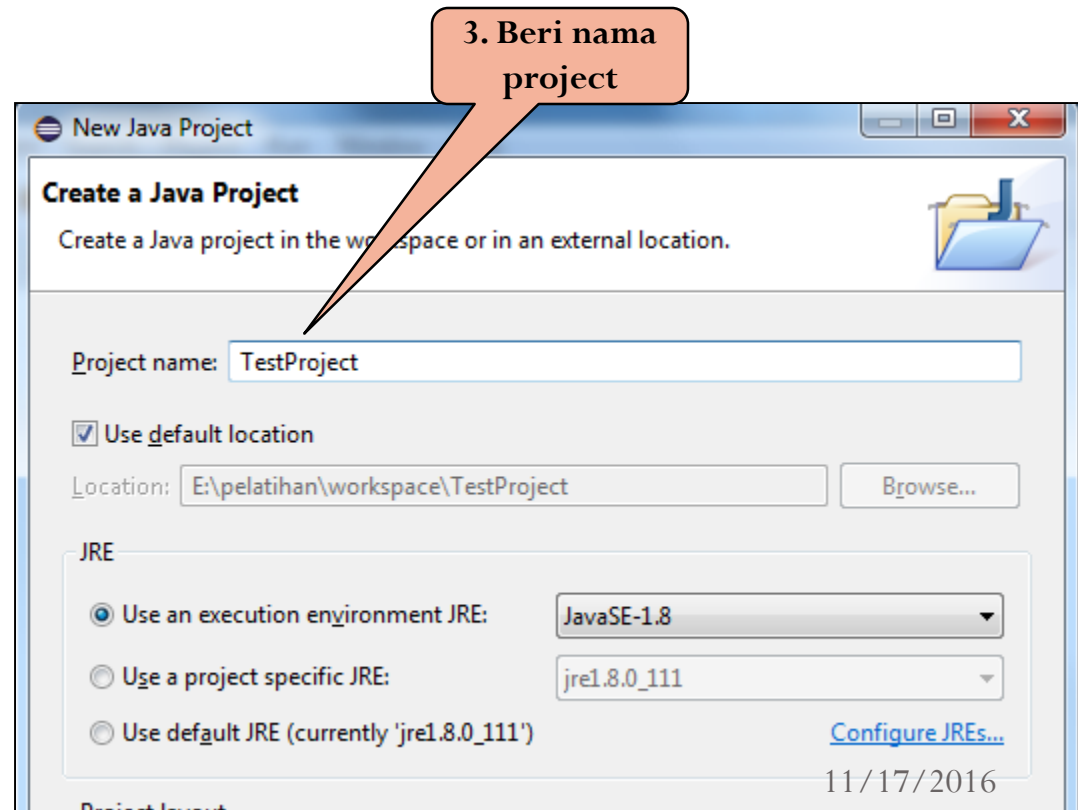
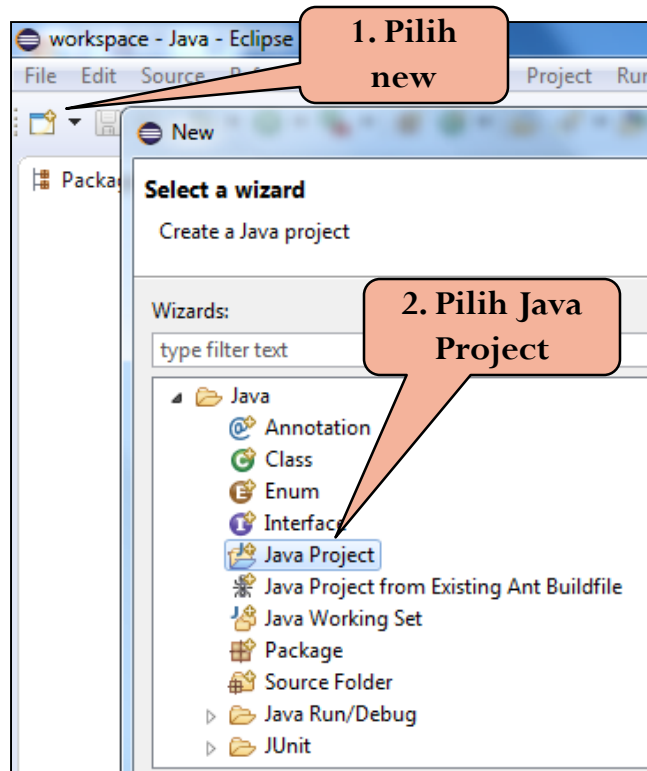
```
E:\pelatihan>java Test  
Hello Lampung
```

Install IDE (Eclipse)

- Buat folder pada direktori tertentu dengan nama “eclipse”
- Ekstrak isi master eclipse pada folder yang telah dibuat tadi
- Buat *shortcut* dari file app eclipse, lalu letakkan di tempat yang mudah dijangkau, seperti desktop.
- Buka IDE eclipse dengan klik *shortcut*, lalu atur letak *workspace*.

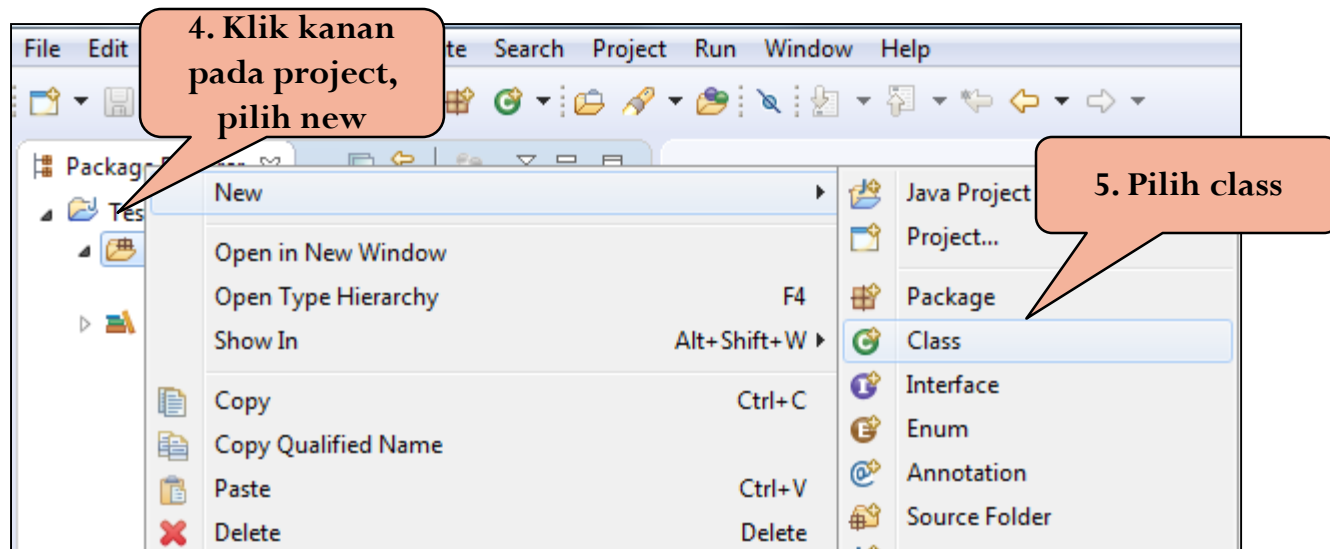
Program dengan IDE (Eclipse)

- Buat project baru



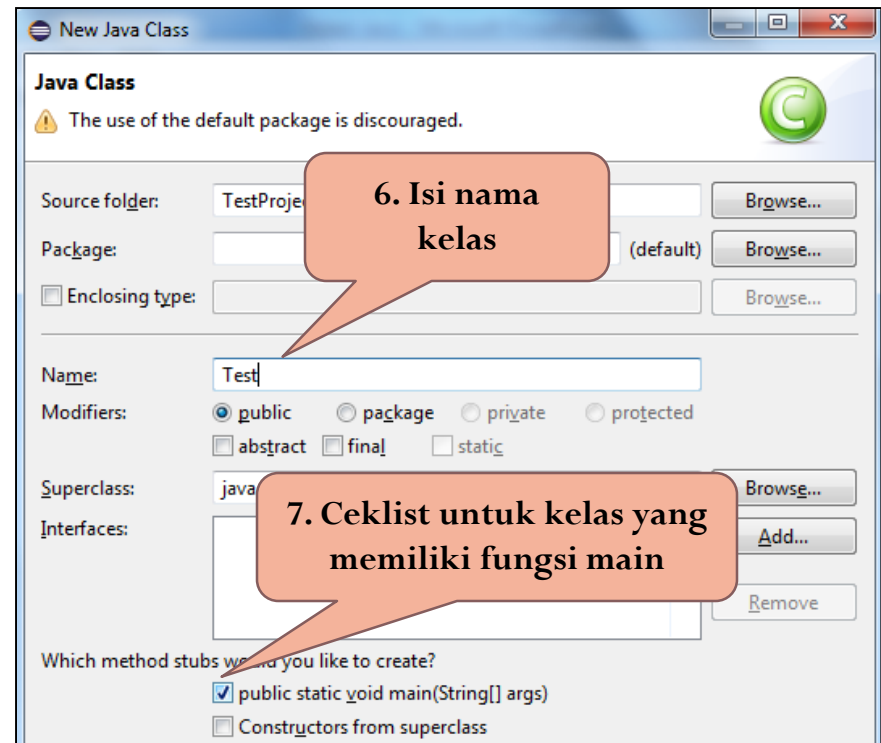
Buat Class

- Class merupakan *blue print* (cetak biru) dari object.



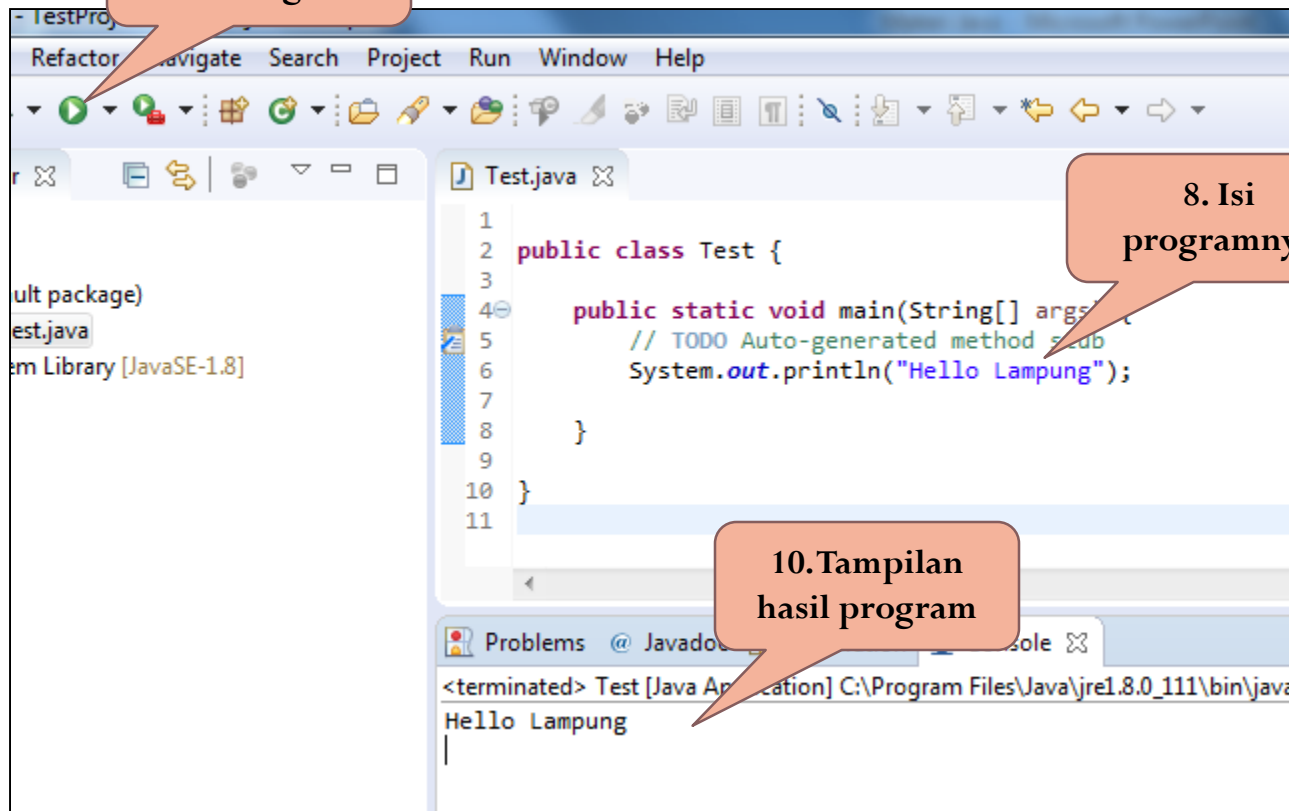
Class Main

- Class Main adalah “*Driver Class*”
- Sebuah class dalam java, hanya bisa dieksekusi dengan *command* “ java NamaClass” jika class tersebut mempunyai fungsi main



Program pertama dengan IDE

9. Compile dan Run Program



8. Isi programnya

10. Tampilan hasil program

Variabel pada Java

- Variabel adalah tempat menyimpan nilai sementara yang dapat diubah-ubah

- Bentuk umum deklarasi variabel :

```
datatype namavariabel;
```

- Contoh

```
int angka;  
string nama;
```

- Bentuk umum deklarasi + inisialisasi variabel :

```
datatype namavariabel = value;
```

- Contoh

```
int angka = 20;  
string nama = "saniati";
```

Jenis Tipe Data

- Primitive
 - a. Tipe data yang sudah *build in* dalam java.
 - b. Hanya menyimpan *value (scalar)*.
 - c. Nama tipe termasuk *reserved keyword* (tidak boleh digunakan sebagai identifier).
- Object (user defined)

Tipe data yang dibuat oleh user seperti class, interface, abstract class.

Tipe Data Primitiv

- Bilangan bulat (integer)
 - long (64 bit). Ex., long a = 123456782345L;
 - int (32 bit) Ex., int b = 345678;
 - short (16 bit) Ex., short c = 123;
 - byte (8 bit) Ex., byte d = 10;
 - Tipe *default* bilangan bulat di java yaitu int (32 bit).
- Single character
 - char (16 bit) , 16 bit agar support internationalization (unicode)
- Bilangan pecahan
 - float (32 bit) Ex., float e = 3.14f;
 - double (64bit) Ex., double f = 2.345678;
 - Tipe *default* bilangan pecahan di java yaitu double (64 bit).
- Boolean (hanya berisi value *true* & *false*)
 - boolean (platform dependen)

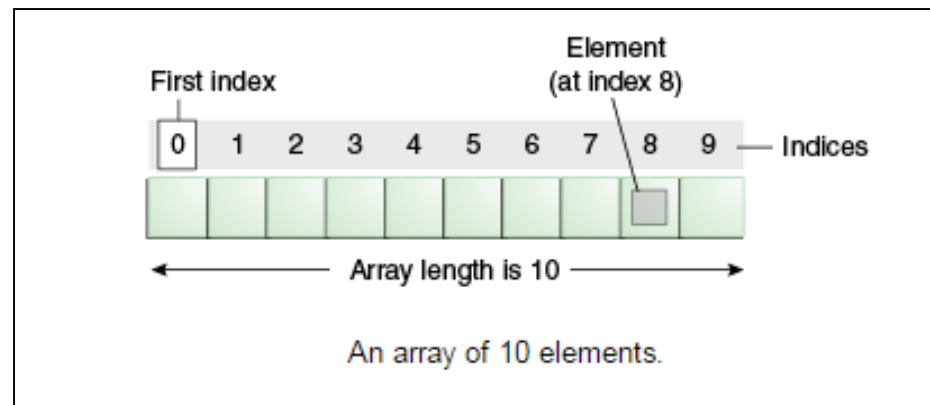
Default value

- Nilai default untuk tiap-tiap tipe data pada java

Data Type	Default Value (for fields)
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
String (or any object)	null
boolean	false

Array

- Array adalah (*collection*) kumpulan banyak data yang setipe.
- Jenis array:
 - Single dimension : dalam 1 elemen hanya ada 1 value.
 - Multi dimension (nested array) : dalam 1 elemen berisi array lagi
- Index array dimulai dari 0.



Deklarasi dan Inisialisasi Array (*single dim.*)

- Deklarasi array:

```
datatype [] namaarray;      atau   datatype namaarray[];  
int [] nilai;              atau   int nilai [];
```

- Deklarasi + inisialisasi:

- Cara 1

```
char [] b = {'a','b','c'};
```

- Cara 2, harus alokasi memory

```
char [] b = new char[3];
```

```
b[0] = 'a';
```

```
b[1] = 'b';
```

```
b[2] = 'c';
```

- Cara 3

```
char [] b ;
```

```
b = new char[]{'a','b','c'};
```

Deklarasi dan Inisialisasi Array (*multi dim.*)

- Deklarasi array:

```
datatype [][] namaarray;      atau      datatype [] namaarray[];  
int [][] nilai;                atau      int [] nilai [];
```

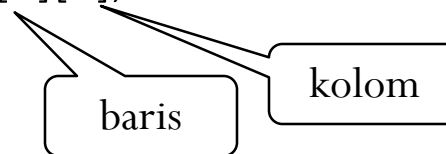
- Deklarasi + inisialisasi:

- Cara 1

```
char [][] b = {{'a','b'},{'d','e'},{'h','i'}};  
int [] a [] = {{1,2},{3,4},{5,6}}
```

- Cara 2, harus alokasi memory

```
char [][] b = new char[2][2];  
b[0][0] = 'a';  
b[0][1] = 'b';  
b[1][0] = 'c';  
b[1][1] = 'd';
```



- Cara 3

```
char [][] b ;  
b = new char[][] {{'a','b'},{'d','e'},{'h','i'}};
```

b.Length (baris)
b[0].length (kolom)

Java Control Structure

- Percabangan (*branching*)

- if
- if else
- if else if
- switch

Condition checking harus bertipe boolean

Switching condition harus bertipe integer/character, (jdk 7 bisa untuk string)

- Perulangan (*looping*)

- While
- For
- Do.. while

Condition checking harus bertipe boolean

If , if else, if else if

- Contoh if

```
int bil=10;
if(bil >= 0){
    System.out.print("Bilangan positif");
}
```

- Contoh if else

```
int bil=10;
if(bil >= 0){
    System.out.print("Bilangan positif");
}else{
    System.out.print("Bilangan negatif");
}
```

- Contoh if else if

```
int bil=10;
if(bil == 0){
    System.out.print("Bilangan netral");
}else if(bil>0){
    System.out.print("Bilangan positif");
}else{
    System.out.print("Bilangan negatif");
}
```

Switch Case

*Switching condition
integer*

```
int nilaimutu=5;
switch(nilaimutu)
{
    case 1:
        System.out.print("Excel");
        break;
    case 2:
        System.out.print("Good");
        break;
    case 3:
        System.out.print("Fair");
        break;
    case 4:
        System.out.print("Bad");
        break;
    case 5:
        System.out.print("Failed");
        break;
    default:
        System.out.print("Error");
}
```

*Switching condition
character*

```
char hurufmutu='a';
switch(hurufmutu)
{
    case 'a':
        System.out.print("Excel");
        break;
    case 'b':
        System.out.print("Good");
        break;
    case 'c':
        System.out.print("Fair");
        break;
    case 'd':
        System.out.print("Bad");
        break;
    case 'e':
        System.out.print("Failed");
        break;
    default:
        System.out.print("Error");
}
```

*Switching condition String,
untuk java versi terbaru*

```
String aksara="ga";
switch(aksara)
{
    case "ka":
        System.out.print("Excel");
        break;
    case "ga":
        System.out.print("Good");
        break;
    case "nga":
        System.out.print("Fair");
        break;
    case "pa":
        System.out.print("Bad");
        break;
    case "ba":
        System.out.print("Failed");
        break;
    default:
        System.out.print("Error");
}
```

For

- Perulangan increment

```
for(int i=0;i<10;i++){  
    System.out.println("Counter "+i);  
}
```

- Perulangan decrement

```
for(int i=10;i>=0;i--){  
    System.out.println("Counter "+i);  
}
```

While

- Perulangan increment

```
int j=0;
while(j<=10){
    System.out.println("Counter "+j);
    j++;
}
```

- Perulangan decrement

```
int j=10;
while(j>=0){
    System.out.println("Counter "+j);
    j--;
}
```

Do while

- Beda do while dengan while yaitu do while minimal dikerjakan satu kali walaupun kondisi tidak terpenuhi.

```
int j=-10;  
while(j>=0){  
    System.out.println("Counter "+j);  
    j--;  
}
```

Kondisi tidak pernah terpenuhi, isi while tidak pernah dikerjakan

```
int j=-10;  
do{  
    System.out.println("Counter "+j);  
    j--;  
}while(j>=0);
```

Kondisi tidak pernah terpenuhi, isi do while dikerjakan minimal sekali

Reference

- Pemaparan materi TOT Java Fundamental oleh bapak Tri Haryoko (7-11 Nopember 2016, Bandar Lampung)
- <https://docs.oracle.com/javase/tutorial/java>
- “Thinking in Java”, Bruce Eckel