

# **PRAKTIKUM 1**

---

## **PENGENALAN LINGKUNGAN JAVA**

---

### **A. TUJUAN PEMBELAJARAN**

1. Mengenal dan mempersiapkan lingkungan kerja Java
2. Mampu menerapkan langkah-langkah dalam menginstall Java Development Kit
3. Membuat program sederhana dengan Java
4. Mengkompilasi dan menjalankan program Java
5. Menganalisa beberapa problem yang terjadi saat pemrograman dan memberikan solusi

### **B. DASAR TEORI**

Untuk bisa bekerja dengan Java, maka kita perlu melakukan instalasi **Java Development Kit (JDK)** atau **Java 2 Software Development Kit (J2SDK)**. Setelah proses instalasi selesai, selanjutnya adalah melakukan penyetelan **PATH** dan **CLASSPATH** di dalam sistem. Penyetelan PATH sangat berguna untuk memberitahu sistem dimana kita meletakkan file-file utama Java (diantaranya java, javac, jdb, dan lain-lain). Sedangkan penyetelan CLASSPATH sangat berguna untuk memberitahu sistem dimana kita meletakkan file-file class yang akan kita libatkan dalam program kita. Penyetelan CLASSPATH ini biasa melibatkan dua item, yaitu tanda . (titik) yang menandakan direktori kerja dan tools.jar yang berisikan kumpulan file-file library standar yang disediakan oleh Java.

### **C. TUGAS PENDAHULUAN**

1. Buatlah makalah yang berisi tentang perkembangan teknologi Java dan uraikan berbagai macam teknologi Java serta aplikasinya saat ini.

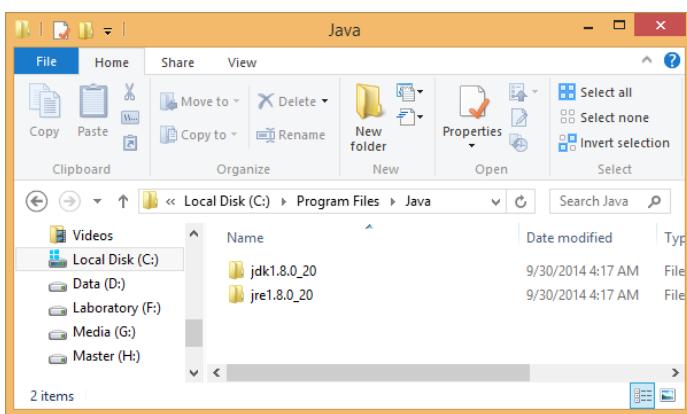
## D. PERCOBAAN

### Percobaan 1 : Instalasi Java

Langkah Percobaan:

#### 1. Instalasi JDK

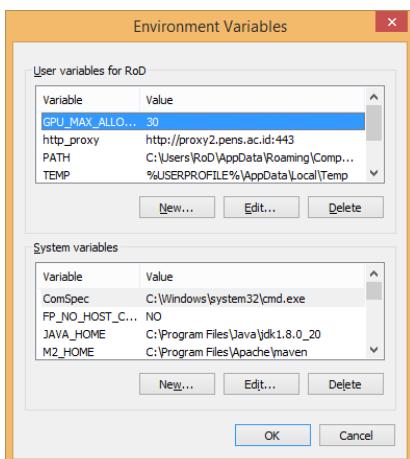
- Download file instalasi JDK sesuai dengan operating system yang anda pakai.  
<http://www.oracle.com/technetwork/java/javase/downloads/jdk9-downloads-3848520.html>
- Jalankan file instalasi JDK dan ikuti proses instalasi tahap demi tahap.
- Pilihlah direktori penginstallan secara default (biasanya ini akan membuat direktori baru atau c:\jdkxxx atau c:\j2sdkxxx dimana xxx adalah versi JDK yang di-install).
- Direktori hasil instalasi JDK



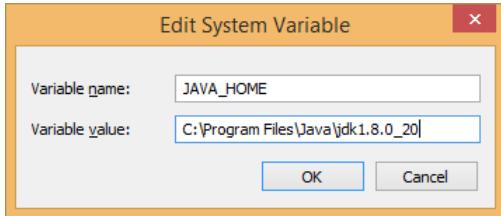
#### 2. Pengesetan PATH

- Masuk ke system setting OS anda melalui control panel

Control Panel → System and Security → System → Advance System Settings → Environment Variable



- Lakukan penambahan JAVA\_HOME pada System Variables dengan cara sebagai berikut:  
Variable value mengikuti direktori instalasi jdk dan versi jdk anda.



### 3. Pengesetan PATH

- Lakukan penambahan PATH pada System Variables dengan value sebagai berikut:  
**PATH = ...;%JAVA\_HOME%\bin**  
\*tambahkan value berwarna merah pada bagian paling akhir

### 4. Pengesetan CLASSPATH

Lakukan penambahan CLASSPATH pada System Variables dengan value sebagai berikut:

**CLASSPATH=...;%JAVA\_HOME%\lib\tools.jar**

### 5. Pengecekan hasil instalasi

- Buka command prompt
- Tulislah “javac” dan eksekusi

```
Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\RoD>javac
Usage: javac <options> <source files>
where possible options include:
  -g                         Generate all debugging info
  -g:none                     Generate no debugging info
  -g:lines,vars,source         Generate only some debugging info
  -nowarn                     Generate no warnings
  -verbose                    Output messages about what the compiler is doing
  -deprecation                Output source locations where deprecated APIs are u
sed
  -classpath <path>           Specify where to find user class files and annotati
on processors
  -cp <path>                  Specify where to find user class files and annotati
on processors
  -sourcepath <path>          Specify where to find input source files
  -bootclasspath <path>        Override location of bootstrap class files
  -extdirs <dirs>              Override location of installed extensions
  -endorseddirs <dirs>        Override location of endorsed standards path
  -proc<none,only>            Control whether annotation processing and/or compil
ation is done.
  -processor <class1>[,<class2>,<class3>]... Names of the annotation processors
to run; bypassed default discover process
  -processorpath <path>        Specify where to find annotation processors
  -parameters                 Generate metadata for reflection on method paramete
rs
  -d <directory>              Specify where to place generated class files
  -s <directory>              Specify where to place generated source files
  -h <directory>              Specify where to place generated native header file
  -s                         Specify whether or not to generate class files for
implicitly referenced files
  -encoding <encoding>         Specify character encoding used by source files
  -source <release>           Provide source compatibility with specified release
  -target <release>            Generate class files for specific VM version
  -profile <profile>           Check that API used is available in the specified p
rofile
  -version                   Version information
  -help                      Print a synopsis of standard options
  -Akey[=value]               Options to pass to annotation processors
  -X                         Print a synopsis of nonstandard options
  -J<flag>                  Pass <flag> directly to the runtime system
  -Werror                    Terminate compilation if warnings occur
  @<filename>                Read options and filenames from file

C:\Users\RoD>
```

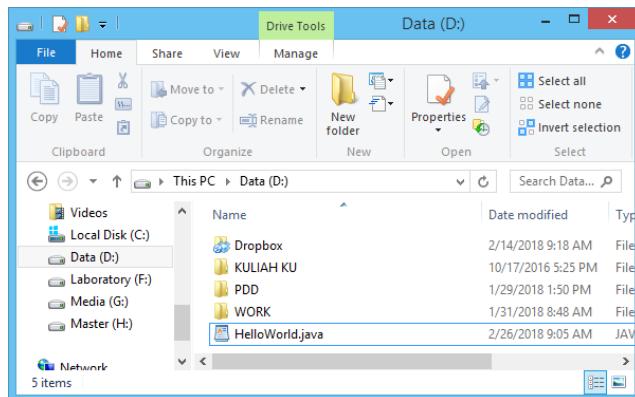
## Percobaan 2 : Membuat Aplikasi Hello world

Langkah Percobaan:

1. Tuliskan kode program berikut pada notepad

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

2. Simpanlah dengan nama “**HelloWorld.java**” pada direktori yang mudah dijangkau (misal C:  
atau D:)



3. Bukalah command prompt dan masuk pada direktori letak file HelloWorld.java anda
4. Eksekusi perintah berikut:

- javac HelloWorld.java
- java HelloWorld

```
D:\>javac HelloWorld.java  
D:\>java HelloWorld  
Hallo World...!  
D:\>
```

5. Maka akan muncul text **Hallo World...!** pada layar

### **Percobaan 3 : Menganalisa dan Memperbaiki kesalahan**

Langkah Percobaan:

1. Tulis kode program berikut

```
public class LuasKubus{  
    public static void main(String[] args){  
        int sisi = 5;  
        int luas = sisi*sisi*sisi;  
        System.out.println("Luas Kubus = " + luas);  
    }  
}
```

2. Simpan dengan nama LuasKubus.java

3. Compile (javac LuasKubus.java)
4. Kemudian Running (java LuasKubus)
5. Output Program

```
D:\>javac LuasKubus.java  
D:\>java LuasKubus  
Luas Kubus = 125  
D:\>
```

## **E. LATIHAN**

### **Latihan 1 : Menganalisa dan membenahi kesalahan program**

```
public class LuasKubus{  
    public void main(String[] args){  
        double r = 5;  
        double luas = Math.PI * r * r;  
        double keliling = 2 * Math.PI * r;  
        System.out.println("Luas = " + luas);  
        System.out.println("Keliling = " + keliling);  
    }  
}
```

- Compile dan jalankan Program diatas.
- Catatlah error yang terjadi.
- Analisa dan perbaiki error tersebut.

## F. TUGAS

1. Buatlah program untuk konversi suhu Celcius menjadi Fahrenheit!

## Lampiran

Cara memasukkan input melalui form JOptionPane!

```
import javax.swing.JOptionPane;  
  
public class InputPane {  
    public static void main(String args[]) {  
        int nilai;  
        String str = JOptionPane.showInputDialog("Masukkan nilai :");  
        nilai = Integer.parseInt(str);  
        System.out.println(nilai);  
    }  
}
```

## G. LAPORAN RESMI

Kumpulkan hasil Percobaan, Latihan dan Tugas di atas dan tambahkan analisa untuk tiap percobaan, latihan, dan tugas yang telah dibuat.