

---

Tugas 1

## 1. Instalasi Java

Jawaban:

```
C:\WINDOWS\system32\cmd. x + v

c:\>java.exe -version
java version "1.8.0_401"
Java(TM) SE Runtime Environment (build 1.8.0_401-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.401-b10, mixed mode)

C:\WINDOWS\system32\cmd. x + v

c:\>javac
Usage: javac <options> <source files>
where possible options include:
  @<filename>           Read options and filenames from file
  -Akey[=value]          Options to pass to annotation processors
  --add-modules <module>(<module>)*
                        Root modules to resolve in addition to the initial modules,
                        or all modules on the module path if <module> is ALL-MODULE-PATH.
  --boot-class-path <path>, -bootclasspath <path>
                        Override location of bootstrap class files
  --class-path <path>, -classpath <path>, -cp <path>
                        Specify where to find user class files and annotation processors
  -d <directory>         Specify where to place generated class files
  -deprecation           Output source locations where deprecated APIs are used
  --enable-preview       Enable preview language features.
                        To be used in conjunction with either -source or --release.
  -encoding <encoding>   Specify character encoding used by source files
  -endorseddirs <dirs>   Override location of endorsed standards path
  -extdirs <dirs>        Override location of installed extensions
  -g                    Generate all debugging info
  -g:{lines,vars,source} Generate only some debugging info
  -g:none               Generate no debugging info
  -h <directory>         Specify where to place generated native header files
  --help, -help, -?      Print this help message
  --help-extra, -X       Print help on extra options
  -implicit:{none,class} Specify whether to generate class files for implicitly referenced files
  -J<flag>               Pass <flag> directly to the runtime system
  --limit-modules <module>(<module>)*
                        Limit the universe of observable modules
  --module <module>(<module>)*, -m <module>(<module>)*
                        Compile only the specified module(s), check timestamps
  --module-path <path>, -p <path>
                        Specify where to find application modules
  --module-source-path <module-source-path>
                        Specify where to find input source files for multiple modules
  --module-version <version>
                        Specify version of modules that are being compiled
  -nowarn               Generate no warnings
```

## 2. Latihan 1 : Cara Kompilasi dan Menjalankan Program Java

Jawaban:

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

```
C:\WINDOWS\system32\cmd.  X + v

c:\>cd C:\Users\nadaf\Documents\PrakPBO

C:\Users\nadaf\Documents\PrakPBO>javac Hallo.java

C:\Users\nadaf\Documents\PrakPBO>java Hallo
Hallo...
```

3. Latihan 2 : Ketik program di bawah ini, bila terjadi kesalahan waktu compile dan run time, Betulkan !

**Jawaban:** Tidak ada kesalahan.

```
public class TestGreeting {
    public static void main (String[] args) {
        Greeting hello = new Greeting();
        hello.greet();
    }
}

class Greeting {
    public void greet() {
        System.out.println("hi");
    }
}
```

```
C:\WINDOWS\system32\cmd.  X + v

C:\Users\nadaf\Documents\PrakPBO>javac TestGreeting.java

C:\Users\nadaf\Documents\PrakPBO>java TestGreeting
hi

C:\Users\nadaf\Documents\PrakPBO>
```

4. Latihan 3 : Test1.java

**Jawaban:**

*The error:* karena memiliki 2 kelas

```
C:\Users\nadaf\Documents\PrakPBO>javac Test1.java
Test1.java:7: error: class TestAnother is public, should be declared in a file named TestAnother.java
public class TestAnother {
      ^
1 error
```

*The fixed:* menghilangkan **'public'**

```
public class Test1 {
    public static void main (String[] args) {
        System.out.println("What is wrong with this program?");
    }
}

class TestAnother {
    public void greet() {
        System.out.println("What is wrong with this program?");
    }
}
```

```
C:\Users\nadaf\Documents\PrakPBO>javac Test1.java
C:\Users\nadaf\Documents\PrakPBO>java Test1
What is wrong with this program?
C:\Users\nadaf\Documents\PrakPBO>
```

```
C:\Users\nadaf\Documents\PrakPB0>java Test1
What is wrong with this program?
```

C:\Users\nadaf\Documents\PrakPB0>

- ## 5. Latihan 4 : Test2.java

**Jawaban:**

```
public class Test2 {
    public static void main (String[] args) {
        System.out.println("What is wrong with this program?");
    }
}
```

```
C:\Users\nadaf\Documents\PrakPB0>javac Test2.java
C:\Users\nadaf\Documents\PrakPB0>java Test2
What is wrong with this program?
C:\Users\nadaf\Documents\PrakPB0>
```

```
C:\Users\nadaf\Documents\PrakPB0>java Test2
What is wrong with this program?
```

```
C:\Users\nadaf\Documents\PrakPB0>
```

- ## 6. Latihan 5 : Test3.java

**Jawaban:**

*The error:* kehilangan main methodnya '//'.  
 //

```
C:\Users\nadaf\Documents\PrakPBO>java Test3
Error: Main method not found in class Test3, please define the main method as:
    public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
```

*The fixed:*

```
public class Test3 {  
    public static void main (String[] args) {  
        System.out.println("What is wrong with this program?");  
    }  
}
```

```
C:\Users\nadaf\Documents\PrakPB0>javac Test3.java  
  
C:\Users\nadaf\Documents\PrakPB0>java Test3  
What is wrong with this program?  
  
C:\Users\nadaf\Documents\PrakPB0>
```