

Algoritma dan Pemrograman

Leon Andretti Abdillah

02

Basic Java with eclipse

Preparation

1. **Java Development Kit (JDK) versi 1.6 keatas.**

Java Development Kit merupakan perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode java menjadi bytecode yang dapat dimengerti dan dapat dijalankan oleh java runtime environment.

2. **Java Runtime Environment (JRE)**

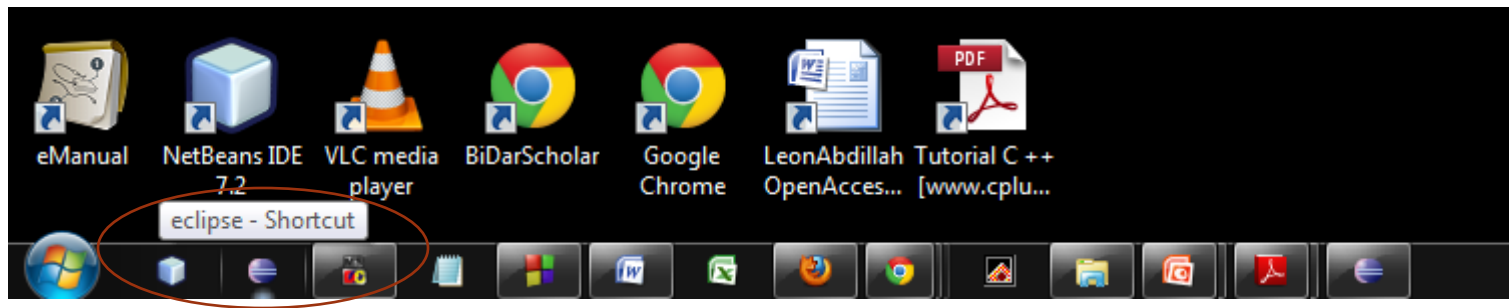
Java Runtime Environment merupakan perangkat lunak yang digunakan untuk menjalankan aplikasi yang dibangun menggunakan java. Versi JRE harus sama atau lebih tinggi dari JDK yang digunakan untuk membangun aplikasi agar aplikasi dapat berjalan sesuai dengan yang diharapkan.

3. **eclipse**

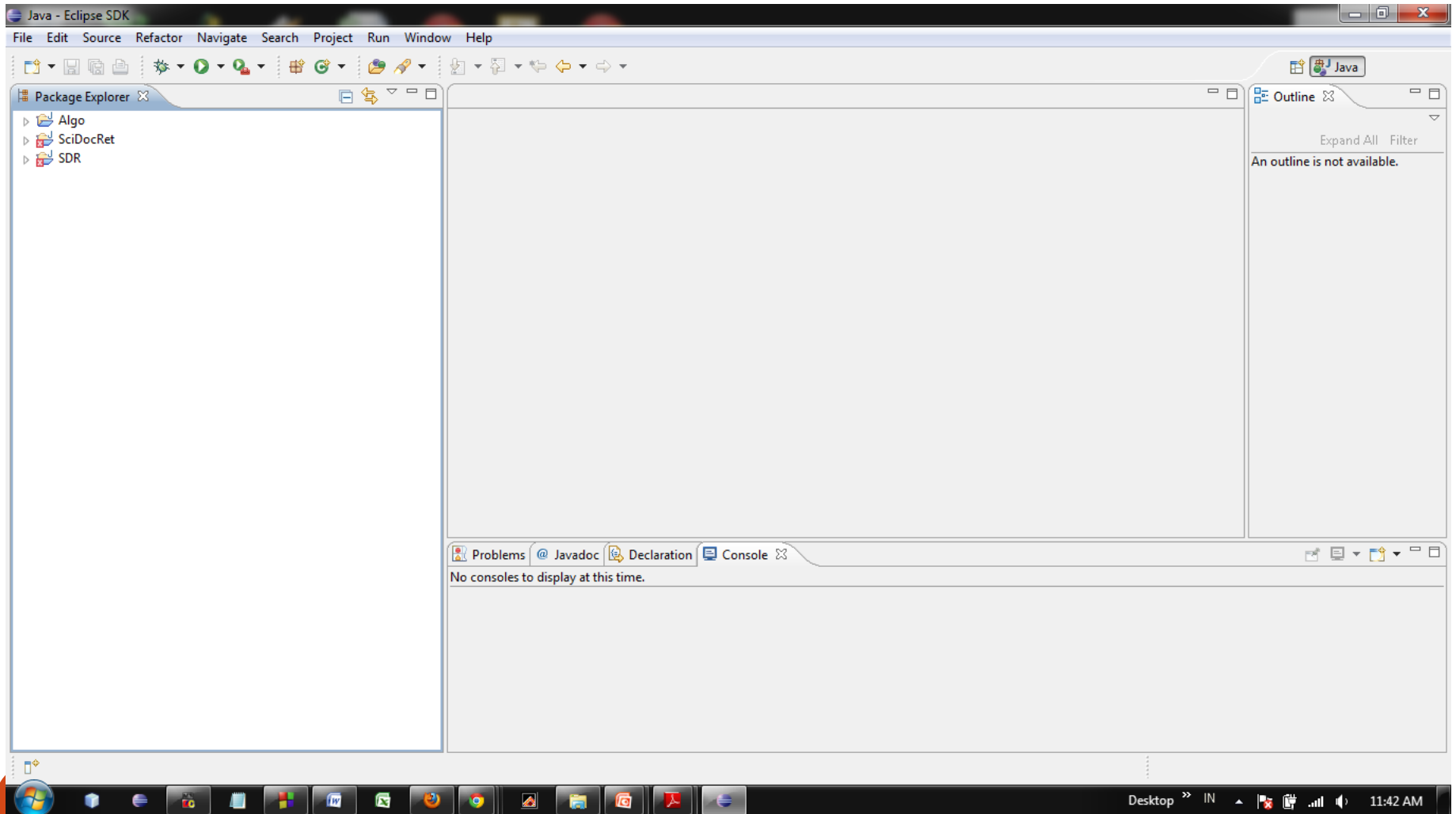
merupakan Java IDE yang dikembangkan oleh IBM dan pada 2001 menjadi nonprofit Eclipse Foundation (www.eclipse.org) untuk mengelola sebagai suatu open-source platform.

Run eclipse

- Click Icon eClipse

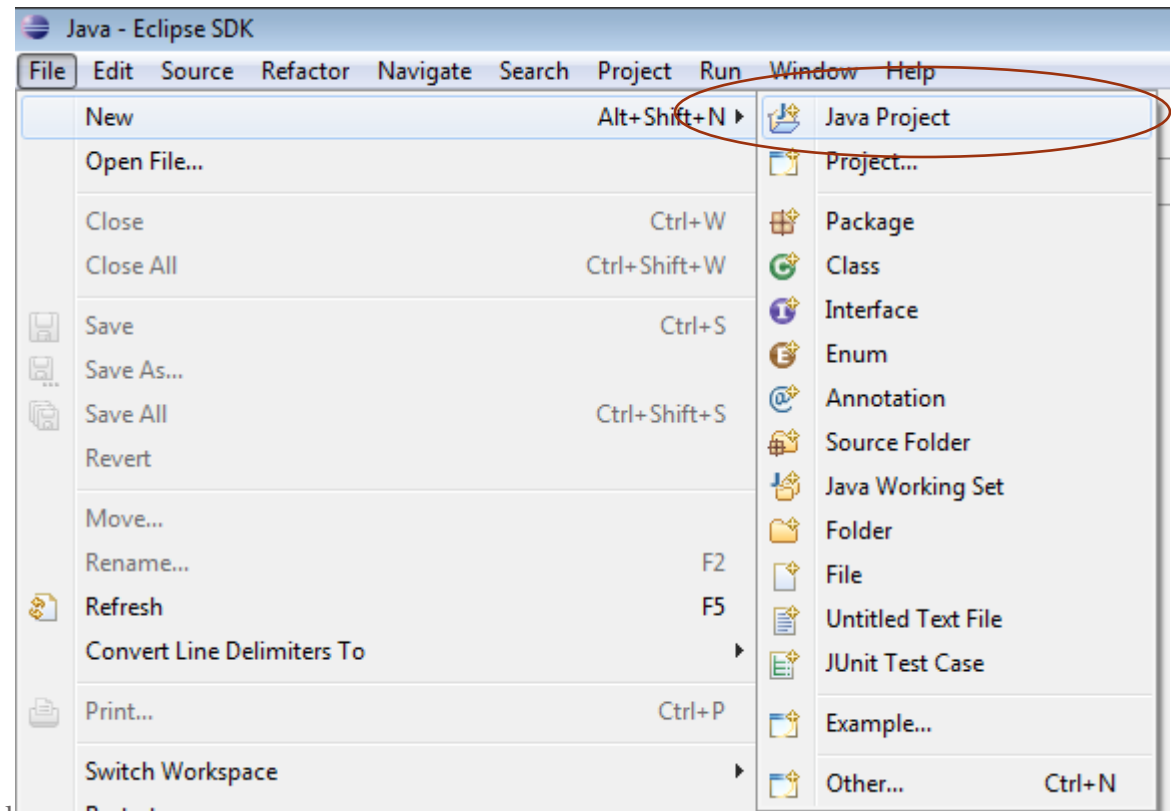


Eclipse ide (Integrated Development Environment)

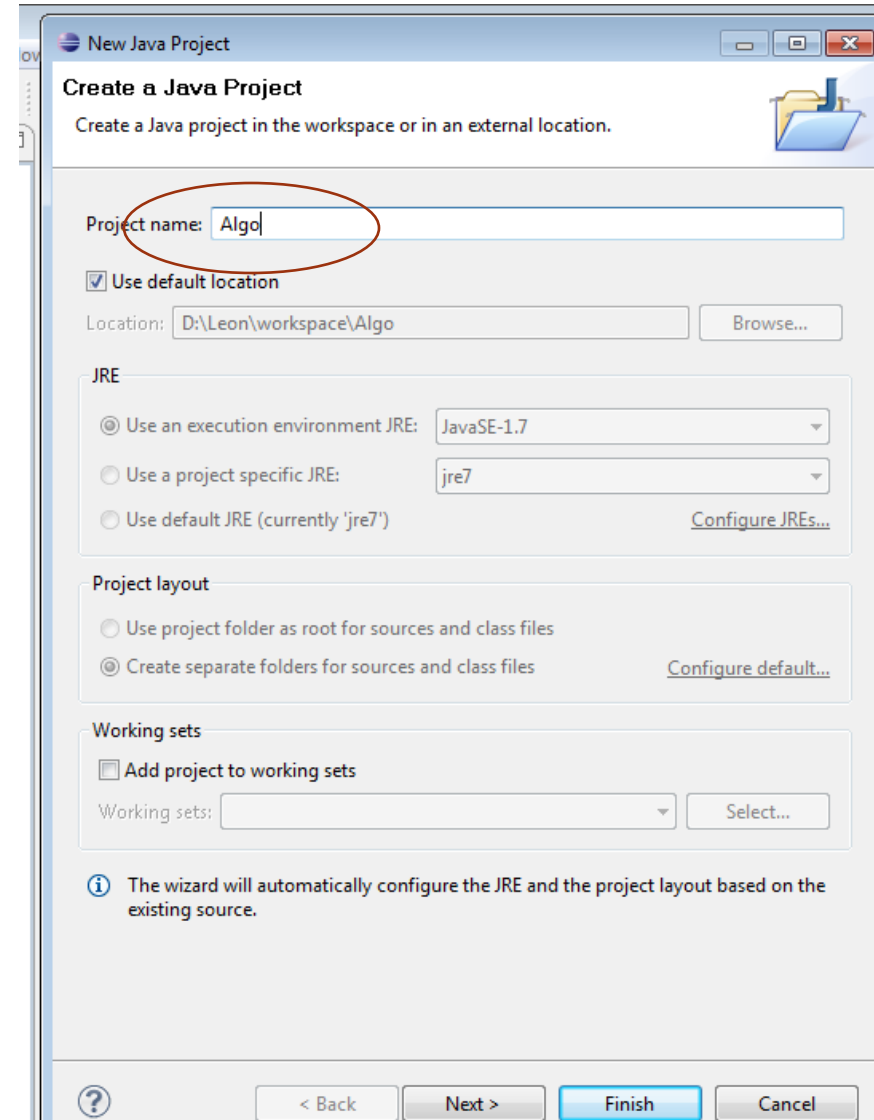


Create new java project

- File | New | Java Project

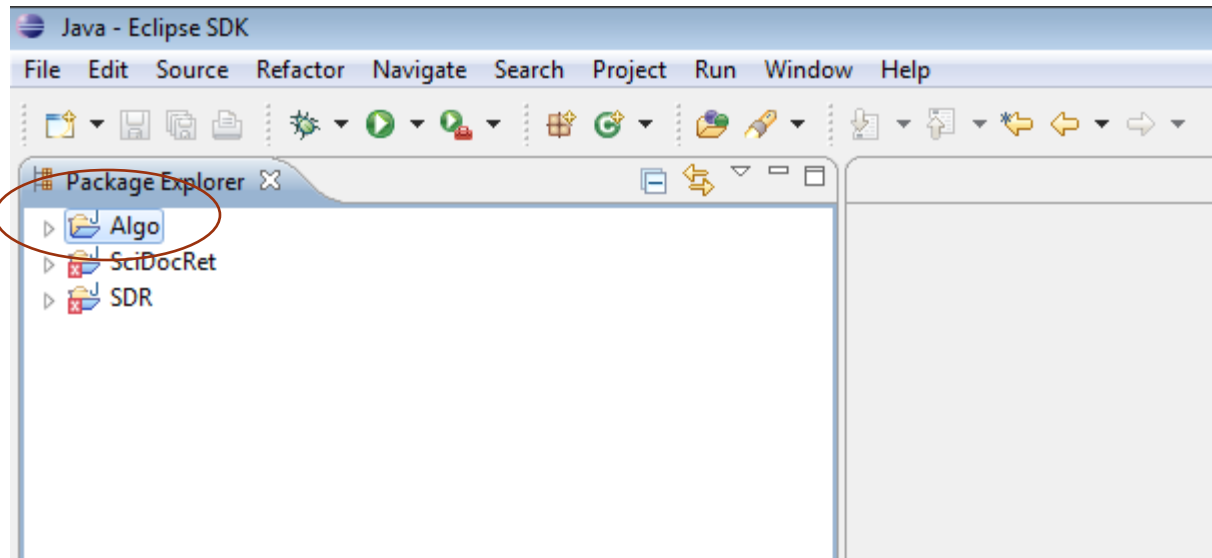


- Pada Project name, ketikkan **Algo** →



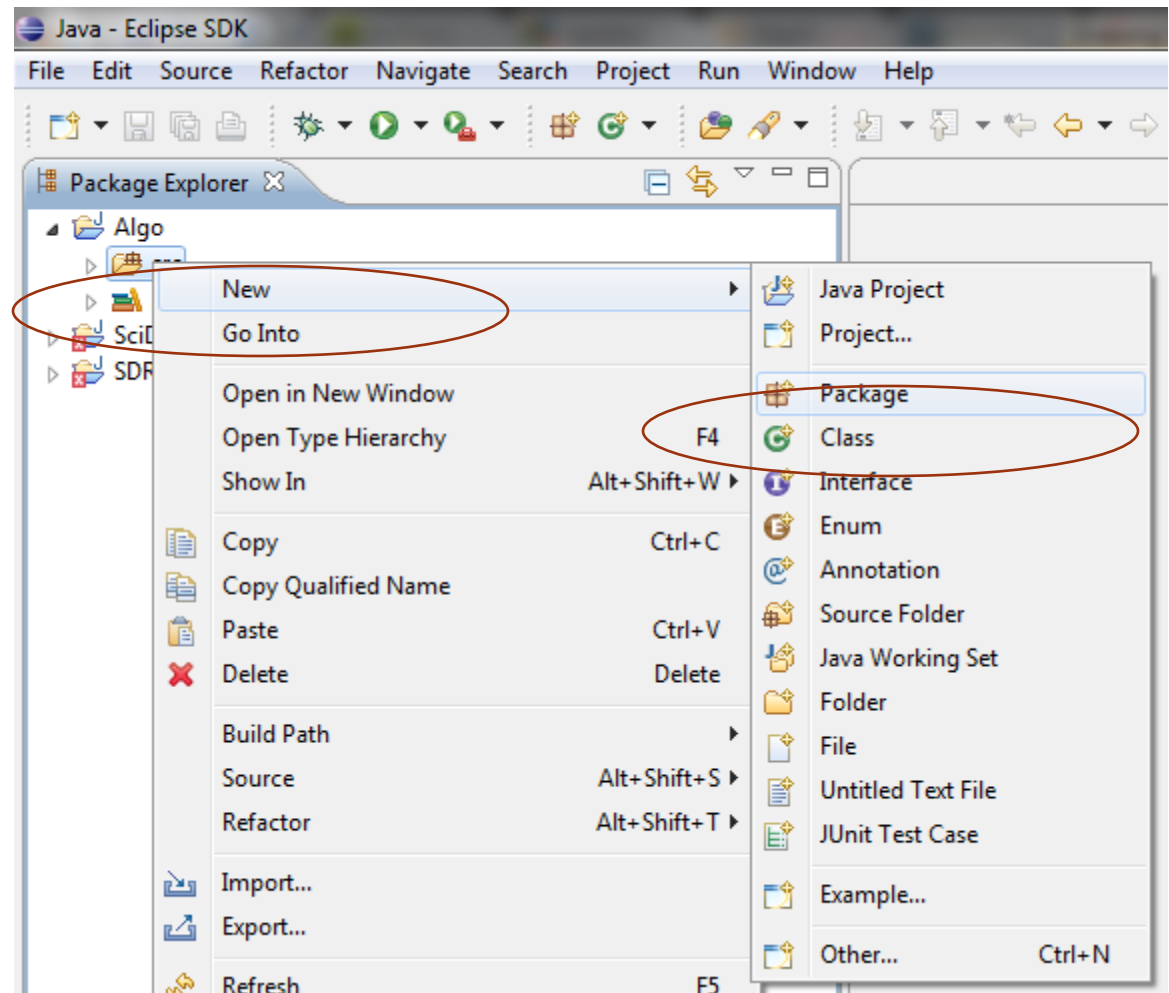
New Java Project

- New Java Project = **Algo**



Create new package

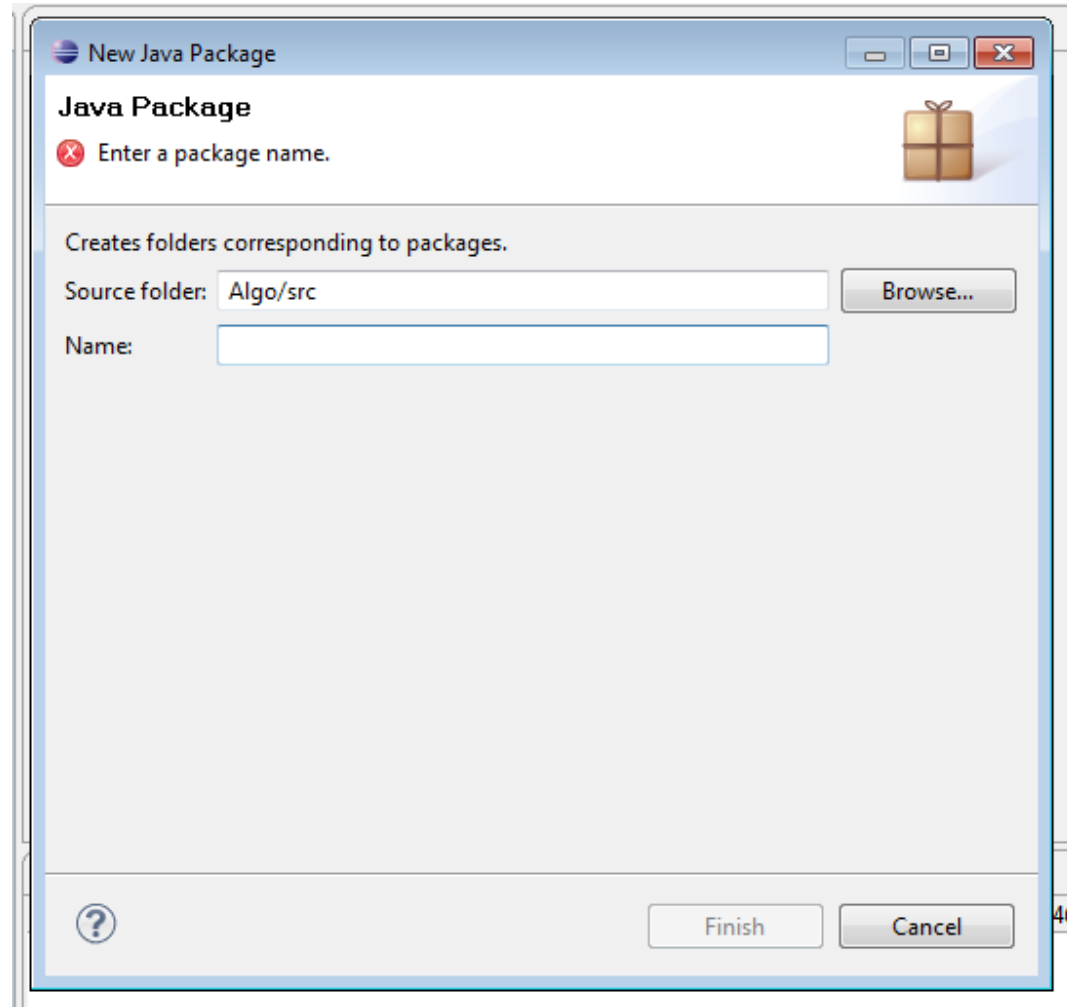
- Pada scr di Algo, klik kanan dan pilih
- Scr | New | Package



Create new package

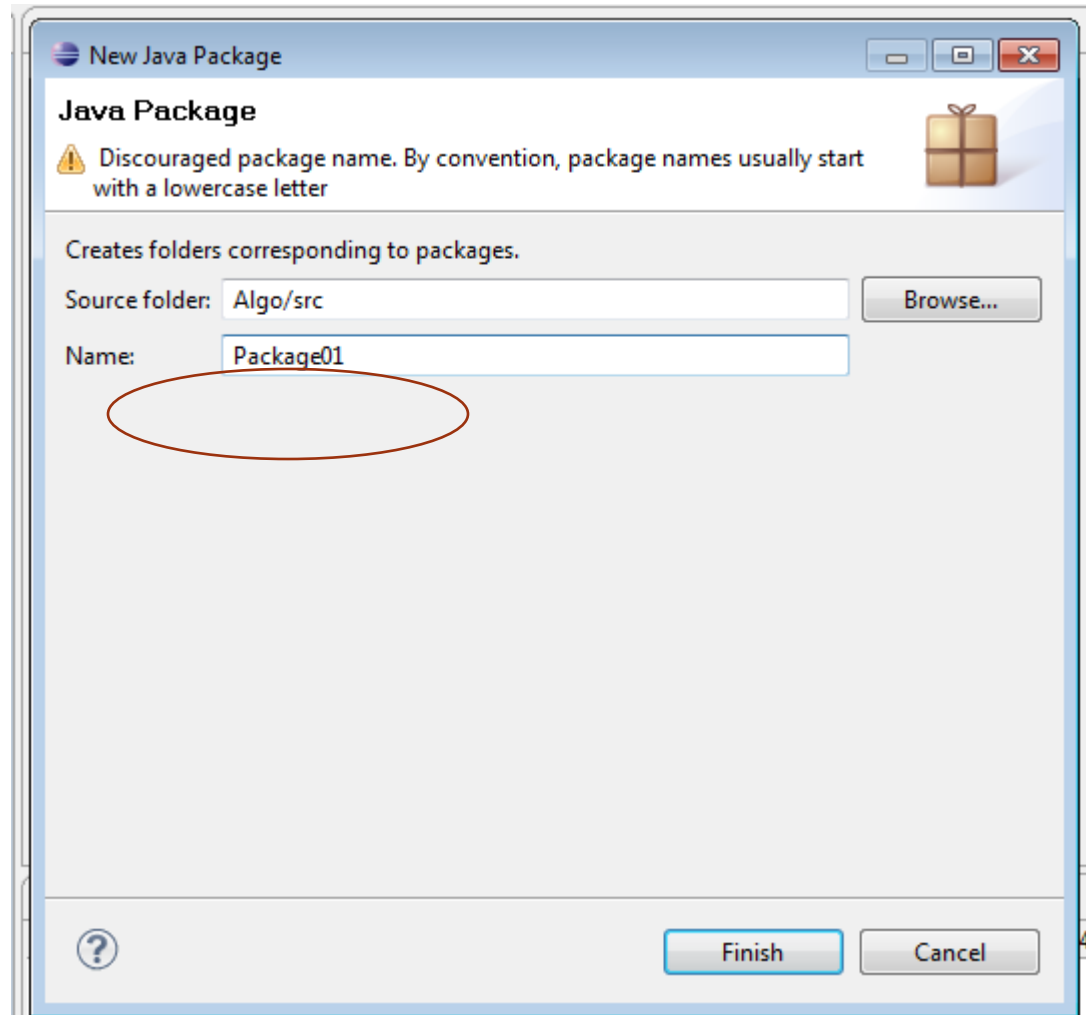
- **Definition:** A *package* is a grouping of related types providing access protection and name space management. Note that *types* refers to classes, interfaces, enumerations, and annotation types. Enumerations and annotation types are special kinds of classes and interfaces, respectively, so *types* are often referred to in this lesson simply as *classes and interfaces*.
- A package is a namespace that organizes a set of related classes and interfaces. Conceptually you can think of packages as being similar to different folders on your computer.
- You might keep HTML pages in one folder, images in another, and scripts or applications in yet another. Because software written in the Java programming language can be composed of hundreds or *thousands* of individual classes, it makes sense to keep things organized by placing related classes and interfaces into packages.

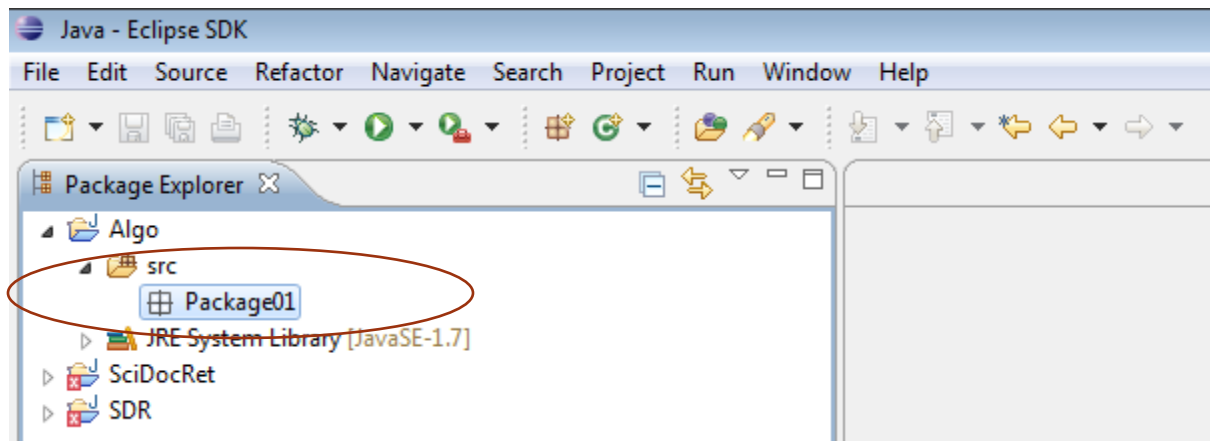
Create new package



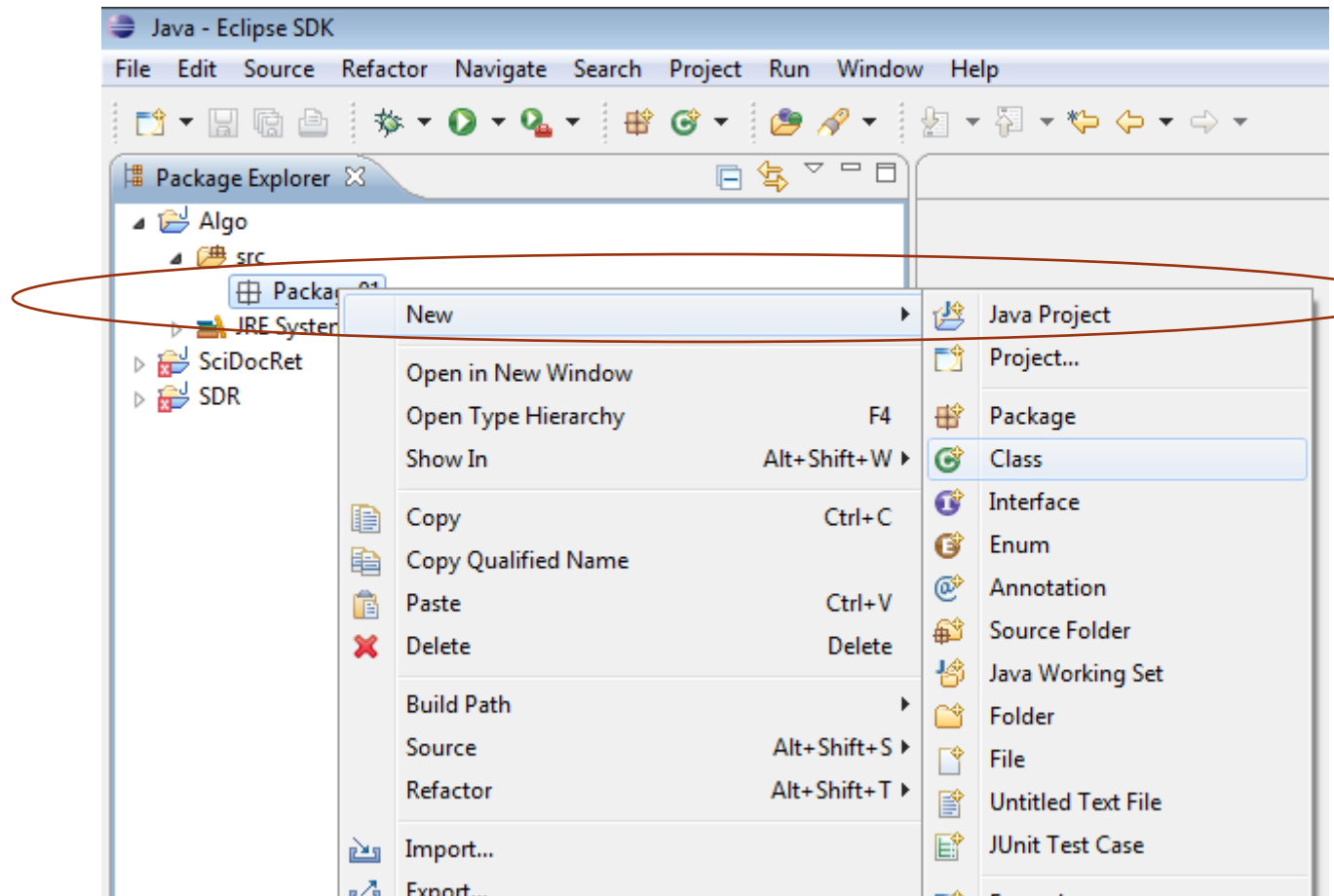
new package = Package01

- Type Name = **Package01**



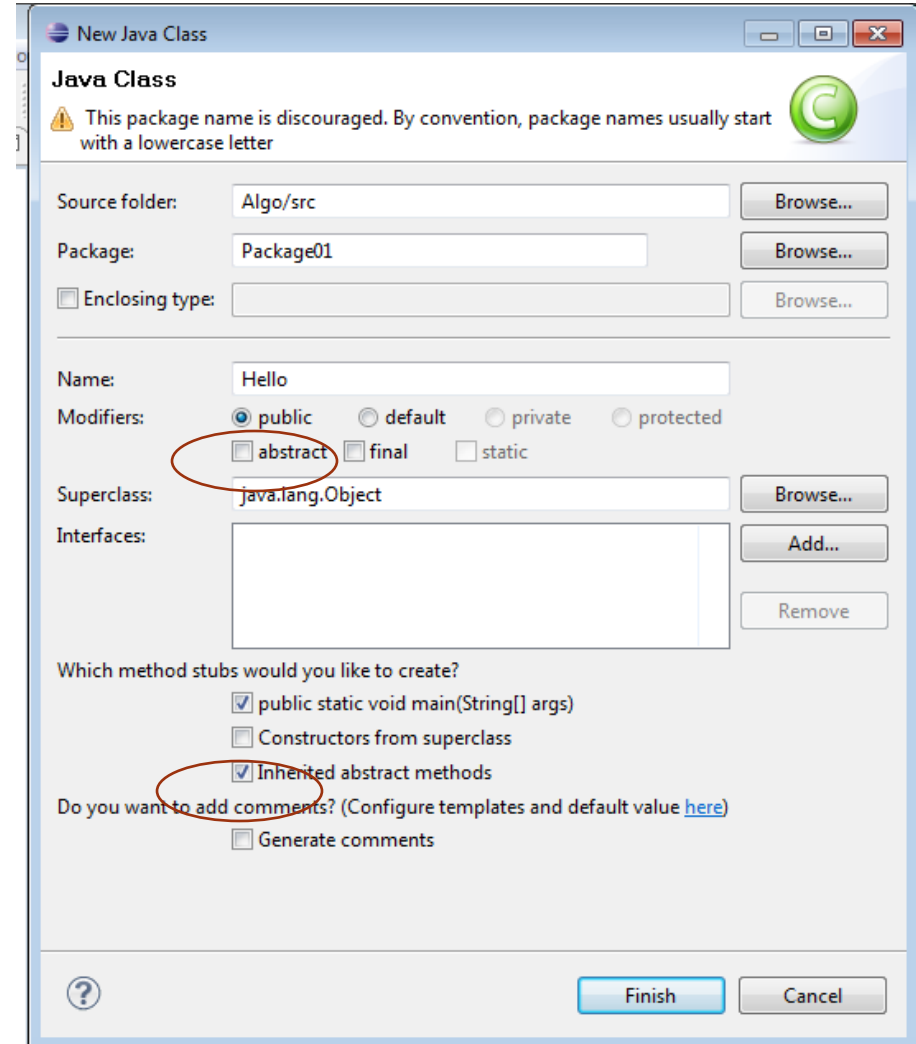


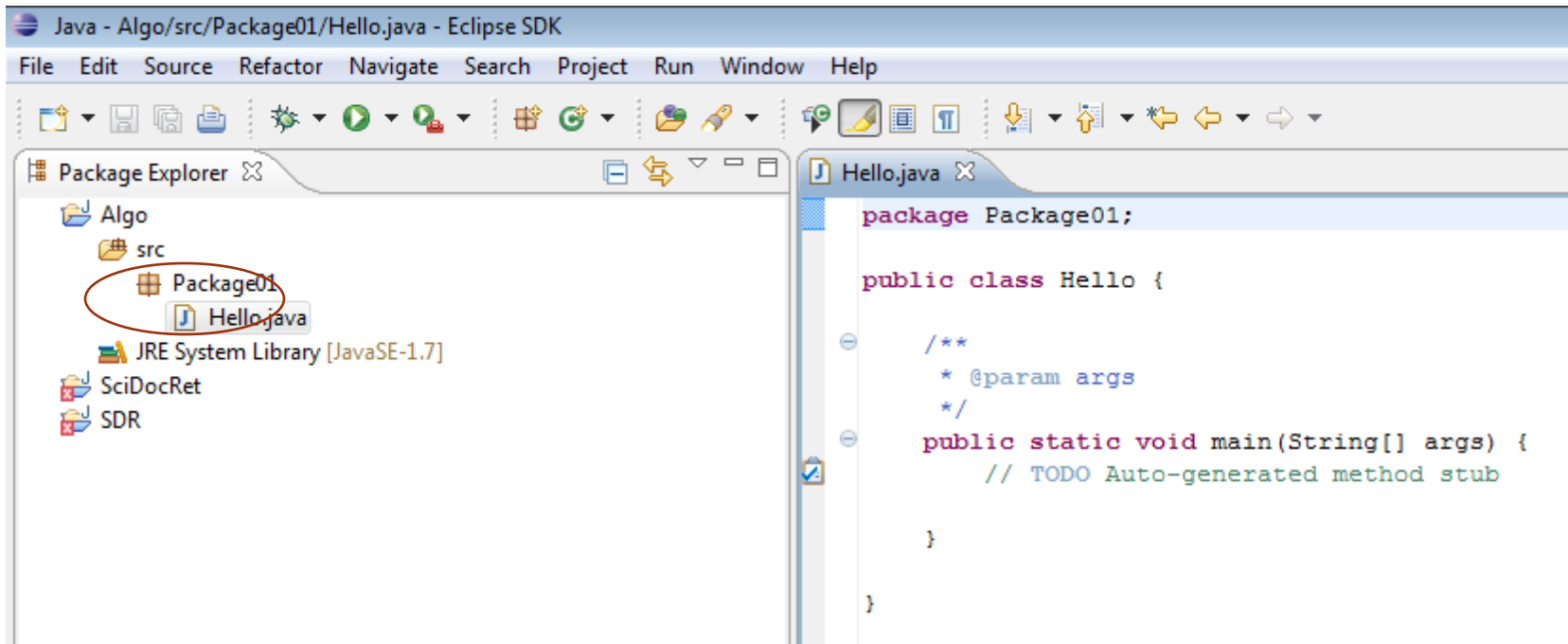
Create new class



New class = hello

- New Class = Hello





```
package Package01;
```

```
public class Hello {
```

```
/**
```

```
 * @param args
```

```
 */
```

```
public static void main(String[] args) {
```

```
 // TODO Auto-generated method stub
```

```
}
```

```
}
```



```
package Package01;
```

```
public class Hello {
```

```
    /**
```

```
     * @param args
```

```
     */
```

```
    public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        System.out.print("Hello");
```

```
        System.out.print("-----");
```

```
        System.out.print("Nama : Jokowi");
```

```
        System.out.print("NIM : 12142001");
```

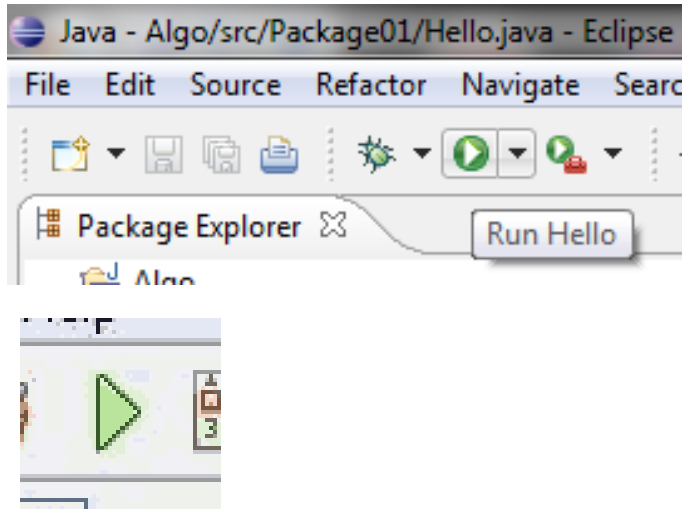
```
        System.out.print("Kelas: TIIA");
```

```
        System.out.print("-----");
```

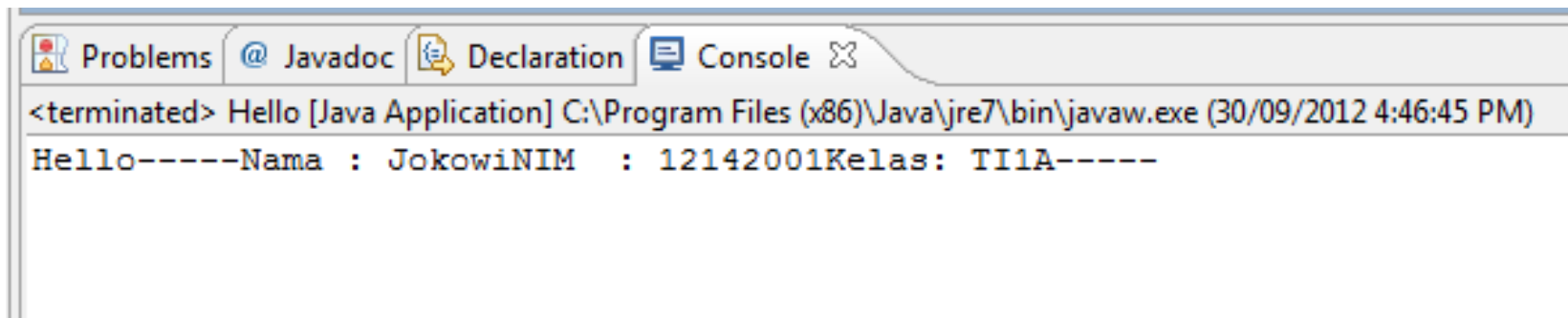
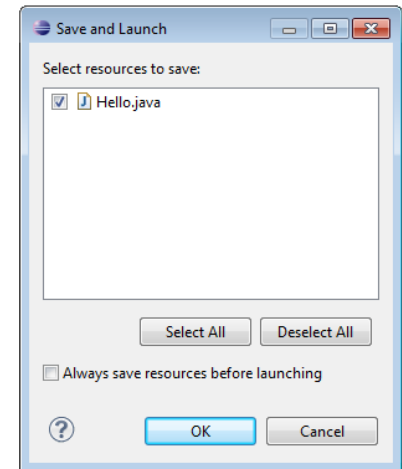
```
    }
```

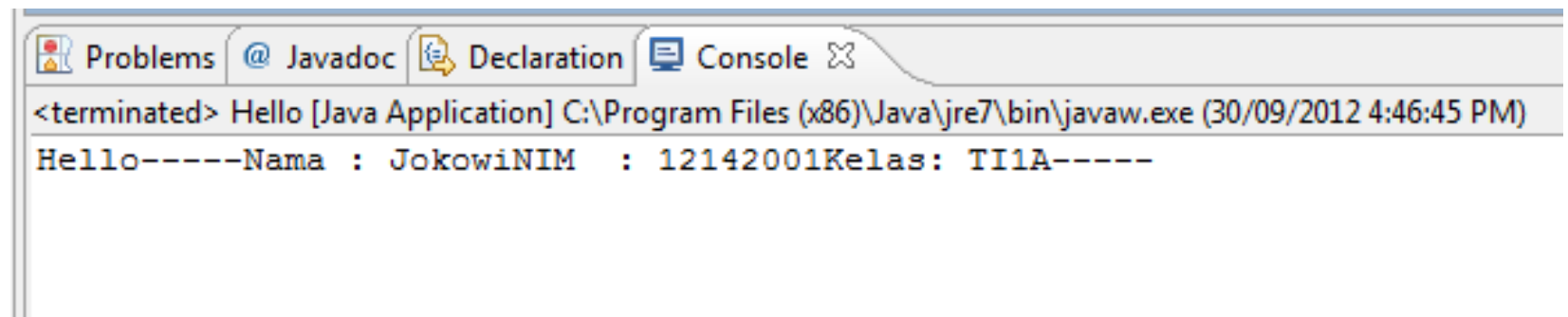
```
}
```

Running the program



- Click the **green arrow** at the top of the window to run the program.





The image shows a screenshot of an IDE's console window. The window title bar includes tabs for 'Problems', 'Javadoc', 'Declaration', and 'Console'. The console output displays the text: '<terminated> Hello [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (30/09/2012 4:46:45 PM)' followed by 'Hello-----Nama : JokowiNIM : 12142001Kelas: TI1A-----'.

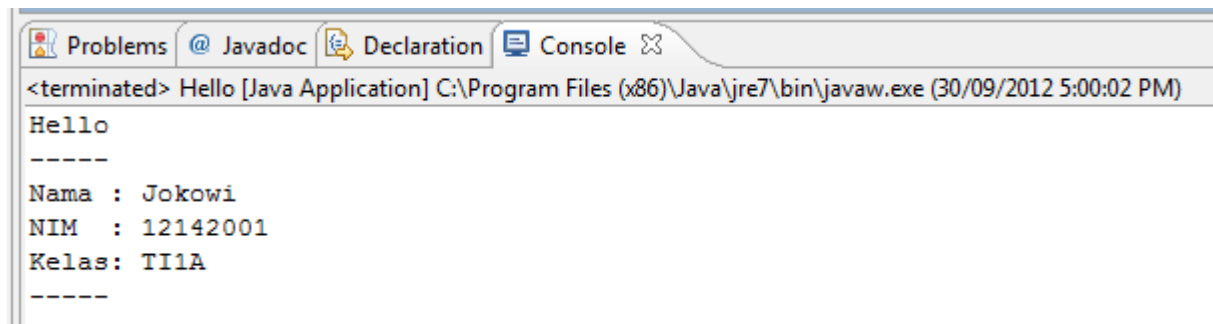
```
<terminated> Hello [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (30/09/2012 4:46:45 PM)
Hello-----Nama : JokowiNIM : 12142001Kelas: TI1A-----
```

```
package Package01;

public class Hello {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.print("Hello \n");
        System.out.print("----- \n");
        System.out.print("Nama :Jokowi \n");
        System.out.print("NIM : 12142001\n");
        System.out.print("Kelas:TIIA \n");
        System.out.print("----- \n");
    }

}
```



The image shows a screenshot of an IDE's console window. The window title bar includes tabs for 'Problems', 'Javadoc', 'Declaration', and 'Console'. The console output is as follows:

```
<terminated> Hello [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (30/09/2012 5:00:02 PM)
Hello
-----
Nama : Jokowi
NIM  : 12142001
Kelas: TI1A
-----
```

Notes

- Nama file harus sama dengan nama kelas program. Misal pada kode diatas nama kelasnya adalah **Hello**, maka nama file harus **Hello.java**
- Hanya boleh terdapat satu kelas public pada sebuah file
- Kelas yang menjadi program harus memiliki metode **public static void main(String[] args)**
- Terminal pada java menggunakan tanda ; (titik koma).

Escape Sequences

Escape Sequence	Name	Description
\a	Bell (alert)	Makes a sound from the computer
\b	Backspace	Takes the cursor back
\t	Horizontal Tab	Takes the cursor to the next tab stop
\n	New line	Takes the cursor to the beginning of the next line
\v	Vertical Tab	Performs a vertical tab
\f	Form feed	
\r	Carriage return	Causes a carriage return
\"	Double Quote	Displays a quotation mark (")
\'	Apostrophe	Displays an apostrophe (')
\?	Question mark	Displays a question mark
\\	Backslash	Displays a backslash (\)
\0	Null	Displays a null character