

# PRAKTIKUM 1

## DASAR PEMROGRAMAN JAVA

### 1. TUJUAN

- a. Mahasiswa dapat melakukan instalasi dan konfigurasi path beberapa program yang digunakan untuk membuat program Java
- b. Mahasiswa dapat membuat program Java sederhana yang didukung dari mata kuliah Dasar Pemrograman sebelumnya
- c. Mahasiswa dapat mengatasi permasalahan pada suatu program Java sederhana dan dapat menganalisisnya.

### 2. DASAR TEORI

#### 2.1 Asal Mula Java

Java dibuat dan diperkenalkan pertama kali oleh sebuah tim *Sun Microsystems* yang dipimpin oleh Patrick Naughton dan James Gosling pada tahun 1991 dengan *code name* Oak. Tahun 1995 Sun merubah nama *Oak* tersebut menjadi *Java*. Ide pertama kali kenapa Java dibuat adalah karena adanya motivasi untuk membuat sebuah bahasa pemrograman yang bersifat *portable* dan *platform independent* (tidak tergantung mesin dan sistem operasi) yang dapat digunakan untuk membuat peranti lunak yang dapat ditanamkan (*embedded*) pada berbagai macam peralatan elektronik elektronik konsumen biasa, seperti *microwave*, *remote control*, telepon, *card reader* dan sebagainya. Seperti yang Anda ketahui, peralatan elektronik ini menjadi sulit dan mahal. Karena alasan inilah maka muncullah suatu ide yang kemudian mendasari dan melahirkan apa yang sekarang kita kenal sebagai Java.

Sun merilis versi awal java secara resmi pada awal 1996 yang kemudian terus berkembang hingga muncul JDK 1.1 kemudian JDK 1.2 sehingga mulai versi ini java disebut java 2. Pada tahun 1998-1999 lahirlah teknologi J2EE (Java 2 Enterprise Edition). Terakhir Teknologi Java melahirkan J2ME (Java 2 Micro Edition) yang sudah diadopsi oleh Nokia, Siemens, SonyEricsson, Motorola untuk menghasilkan aplikasi mobile baik games maupun software lainnya yang dapat dijalankan diperalatan mobile seperti handphone.

#### 2.2 Kelebihan dan Karakteristik Java

Berikut ini adalah rangkuman kelebihan dari bahasa Java dibandingkan dengan bahasa pemrograman lainnya :

- a. Bersifat *portable* dan *platform independent*. Program Java yang Anda tulis akan dapat dieksekusi di platform manapun tanpa memerlukan kompilasi ulang (*portable*) asalkan Java Virtual Machine untuk platform tersebut tersedia.
- b. Memiliki *garbage collection* yang dapat mendealokasi memori secara otomatis. Anda tidak perlu secara eksplisit membebaskan suatu lokasi memori yang dipakai karena ini akan dilakukan secara otomatis oleh Java
- c. Menghilangkan pewarisan berganda yang terdapat pada C++. Walaupun kelihatannya lebih sebagai suatu kekurangan, namun banyak para ahli yang mengakui bahasa konsep pewarisan berganda lebih banyak mengakibatkan kerugian dari pada keuntungan. Java telah didesain sedemikian rupa sehingga Anda tidak akan memerlukan teknik ini dalam pembuatan program apa pun.
- d. Mengurangi pointer aritmetik. Pengaksesan lokasi memori secara langsung dengan menggunakan pointer memungkinkan program untuk melakukan suatu tindakan yang tidak seharusnya atau tidak boleh dilakukan. Untuk mengurangi dan menghilangkan kemungkinan kesalahan seperti ini, penggunaan pointer pada Java telah dibatasi dengan menggunakan *reference*
- e. Memiliki array sejati
- f. Mengurangi kerancuan antara pemberian nilai pada statemen kondisional. Contoh penggunaan tanda '=' dan '==' pada kondisi if.

- g. Perpustakaan kelas yang lengkap, Java terkenal dengan kelengkapan library / perpustakaan (kumpulan program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya.
- h. Bergaya C++, memiliki sintaks seperti bahasa pemrograman [C++] sehingga menarik banyak pemrogram C++ untuk pindah ke Java.

Selain itu para desainer Java sendiri telah merangkum beberapa karakteristik dari bahasa Java yang telah dikembangkan, dimana karakteristik inilah yang membuat Java menjadi seperti apa yang kita kenal sekarang ini sehingga dapat dikatakan sebagai ciri khas dari Java itu sendiri :

- a. **Berorientasi objek**, Java telah menerapkan konsep pemrograman berorientasi objek yang modern dalam implementasinya.
- b. **Robust**, Java mendorong pemrograman yang bebas dari kesalahan dengan bersifat *strongly typed* dan memiliki *run-time checking*.
- c. **Portable**, program Java dapat dieksekusi di platform manapun selama tersedia Java Virtual Machine untuk platform tersebut.
- d. **Multithreading**, Java mendukung penggunaan multithreading yang telah terintegrasi secara langsung dalam bahasa Java.
- e. **Dinamis**, program Java dapat melakukan suatu tindakan yang ditentukan pada saat eksekusi program dan bukan pada saat kompilasi.
- f. **Sederhana**, Java menggunakan bahasa yang sederhana dan mudah dipelajari.
- g. **Terdistribusi**, Java didesain untuk berjalan pada lingkungan yang terdistribusi seperti halnya internet.
- h. **Aman**, aplikasi yang dibuat dengan bahasa Java dapat dipastikan keamanannya terutama untuk aplikasi internet.
- i. **Netral secara arsitektur**, Java tidak terikat pada suatu mesin atau sistem operasi tertentu.
- j. **Interpreted**, aplikasi Java dapat dieksekusi pada platform yang berbeda-beda dengan melakukan interpretasi pada bytecode.

J2SDK adalah seperangkat tool (kit) yang berisi kompiler dan interpreter java yang direlease oleh sun dan bersifat free.

**Tabel Program pada J2SDK**

Nama Program	Keterangan
javac	Kompiler
java	Interpreter
jdb	Debugger
javap	Disassembler
appletviewer	Penampil applet
javadoc	Pembangkit dokumentasi
javah	Pembangkit berkas header C

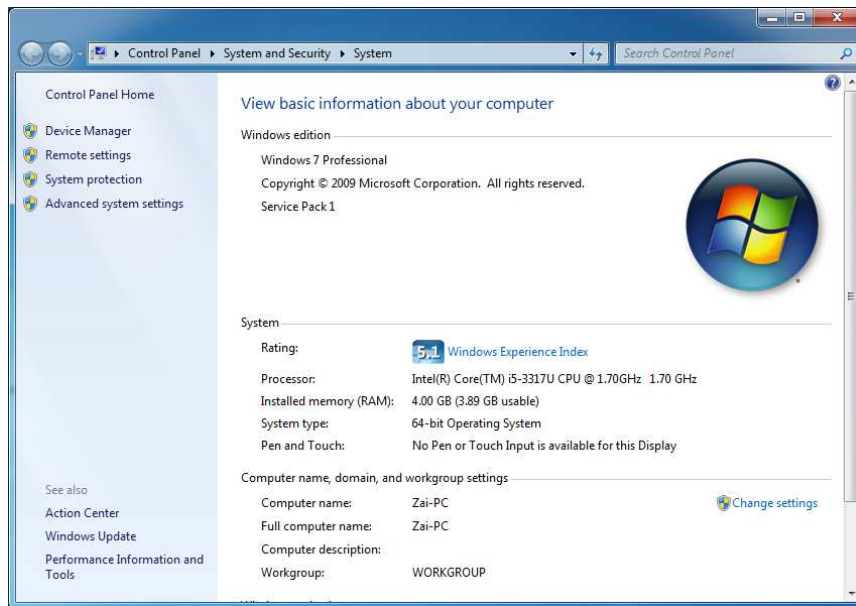
Beberapa editor yang bisa digunakan untuk pemrograman java adalah:

- a. Notepad, editor standard yang disediakan oleh windows. Tidak dilengkapi dengan utilitas jumlah baris (berpengaruh pada pelacakan baris program yang menyebabkan error kompilasi).
- b. Netbeans, salah satu editor yang dilengkapi dengan pelacakan baris program yang menyebabkan error kompilasi.
- c. Editor-editor lain: Eclipse, Gel, JBuilder, JCreator dan lain-lain.

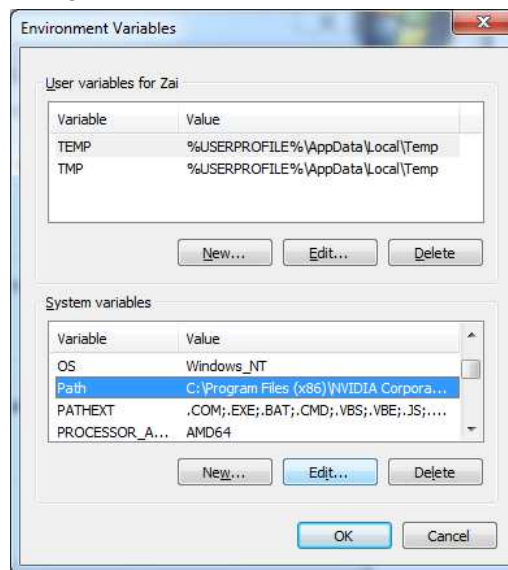
### 2.3 Instalasi JDK dan Netbeans

Untuk membuat program Java perlu dilakukan instalasi program JDK. Sebaiknya digunakan JDK terbaru yang dapat diambil di <http://www.oracle.com/technetwork/java/javase/downloads/java-se-jdk-7-download-432154.html>. Pada praktikum ini digunakan editor Netbeans yang dapat diambil di <http://netbeans.org/downloads/>. Pastikan pilih versi terbaru dan stabil. Selain itu juga pilih installer yang paling lengkap tetapi dengan konsekuensi ukuran file cukup besar. Instalasai JDK dilakukan terlebih dahulu, setelah itu baru install Netbeans. Instalasi JDK dan Netbeans tidak

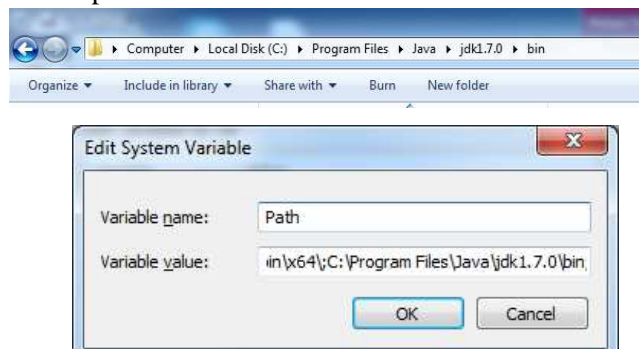
membutuhkan pengetahuan khusus karena proses instalasi dilakukan setting-an default. Hanya saja setelah proses instalasi keduanya diperlukan pengaturan CLASSPATH agar program java yang akan dibuat dapat di-compile dan dijalankan melalui cmd atau command prompt. Untuk setting CLASSPATH pada windows 7, pilih **Computer** klik kanan dan pilih **Properties** sehingga tampil seperti di bawah ini.



Pilih **Advanced system settings**. Pada bagian tab **Advanced** tekan tombol **Environment Variables**. Pilih **Path** pada bagian **System Variables** dan tekan **Edit**.



Tambahkan letak folder bin hasil instalasi JDK pada bagian Variable value dengan cara copy address pada windows explorer.



Saat peletaan classpath jdk, gunakan ; dan tanpa spasi dengan claspath lain yang sudah ada. Untuk melakukan pengecekan apakah setting classpath sudah berhasil buka command prompt dan jalankan perintah **javac**. Apabila tampilan seperti di bawah ini maka setting classpath berhasil.

```

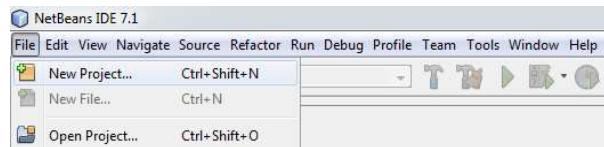
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.


C:\Users\Zai>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 used
  -classpath <path> Specify where to find user class files and annotations
  -cp <path>        Specify where to find user class files and annotations
  -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 compilation is done.
  -processor <class1>[,<class2>[,<class3>...]] Names of the annotation processors to run; bypasses default discovery process
  -processorpath <path> Specify where to find annotation processors
  -d <directory>    Specify where to place generated class files
  -s <directory>    Specify where to place generated source files
  -implicit:<none,class> 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
  -version          Version information
  -help            Print a synopsis of standard options
  -AkeyI=valueI    Options to pass to annotation processors
  -X<flag>         Print a synopsis of nonstandard options
  -X<flag>         Pass <flag> directly to the runtime system
  -Xerror          Terminate compilation if warnings occur
  @<filename>      Read options and filenames from file

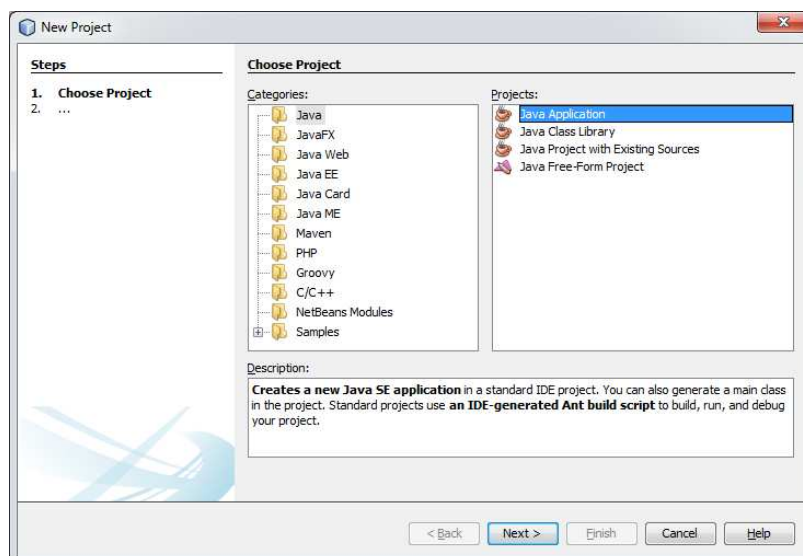
C:\Users\Zai>
  
```

## 2.4 Membuat Project

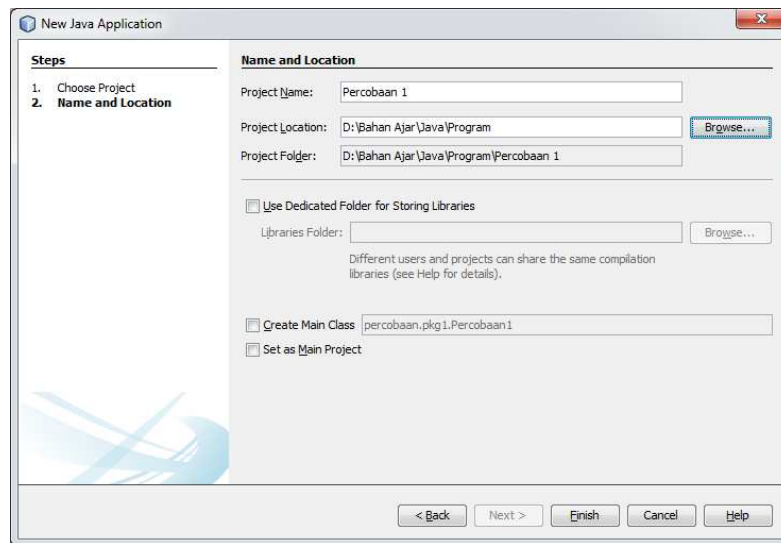
Untuk membuat project, jalankan Netbeans pilih File dan New Project



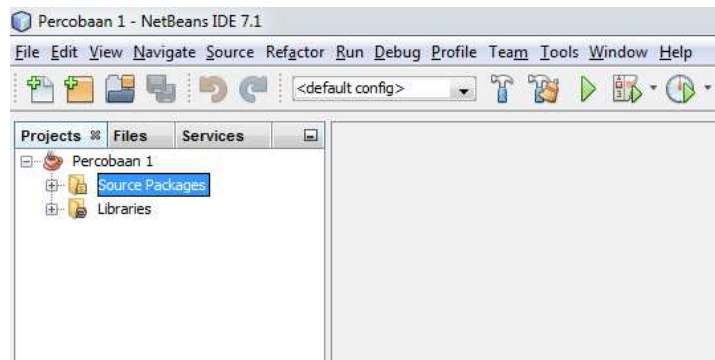
Atau tombol  dan muncul halaman seperti di bawah ini.



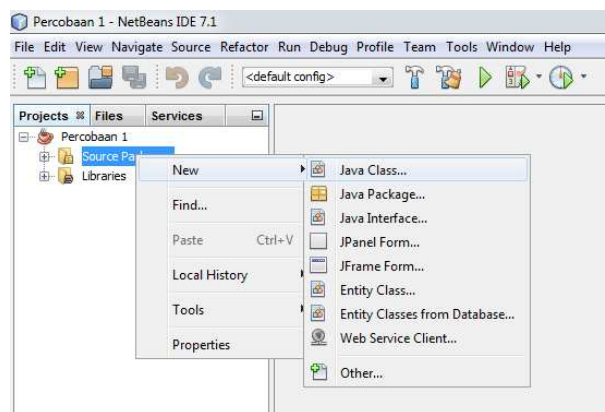
Pada bagian Categories pilih **Java** dan bagian Projects pilih **Java Application** dan tekan **Next**.



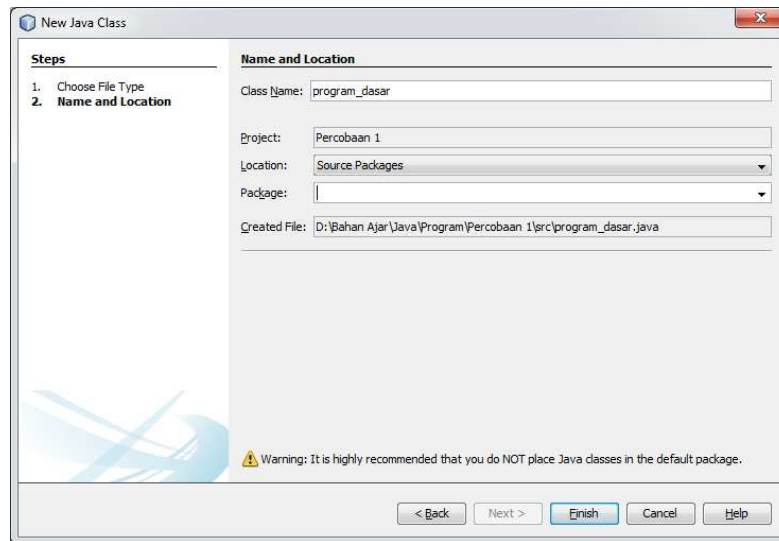
Tuliskan nama project dan tempat penyimpanan. Untuk pemilihan penyimpanan lain tekan Browse. Uncek Create Main Class dan tekan Finish. Hasilnya seperti di bawah ini.



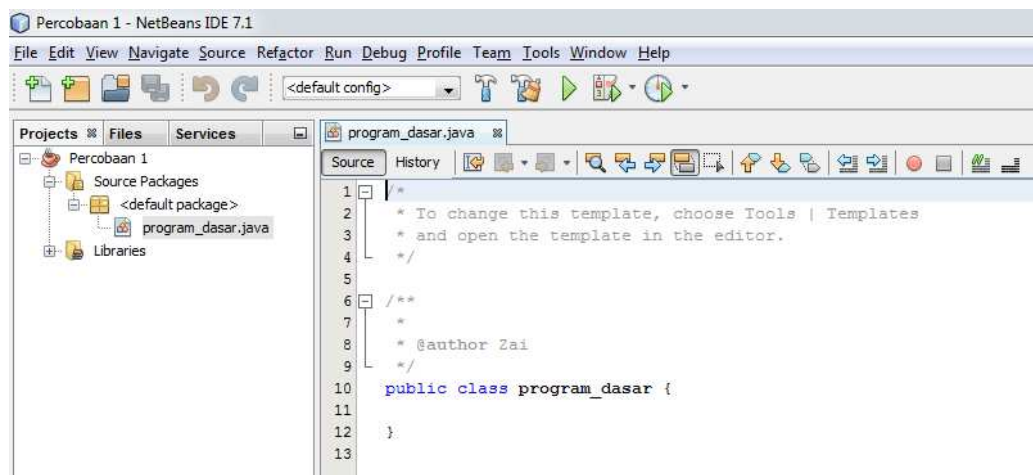
Untuk membuat class atau file .java, pada bagian Source Packages klik kanan pilih New dan Java Class.



Muncul halaman seperti di bawah ini.




Tulis nama class dan bagian Package sementara dikosongi terlebih dahulu dan tekan Finish. Hasilnya seperti di bawah ini.

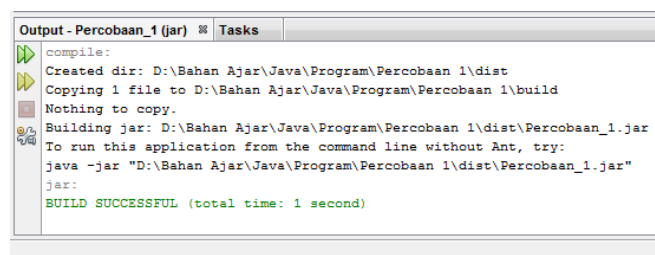


## 2.5 Program Sederhana

Langkah awal dibuat program seperti di bawah ini.

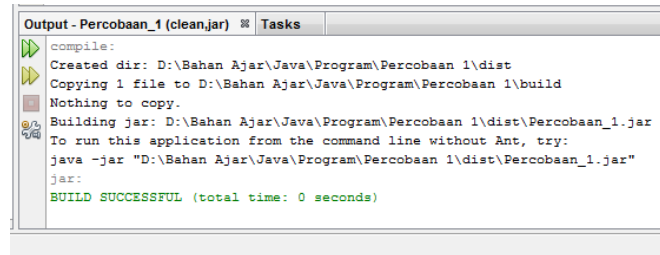
```
public class program_dasar {
    public static void main(String args[]){
        System.out.println("Belajar Pemrograman Java");
    }
}
```


Main program merupakan bagian program yang pertama kali di jalankan. Perintah `System.out.println` digunakan untuk menampilkan suatu string, char, int atau tipe yang lain pada command windows. Untuk build project tekan tombol  atau F11 dan hasilnya seperti di bawah ini.

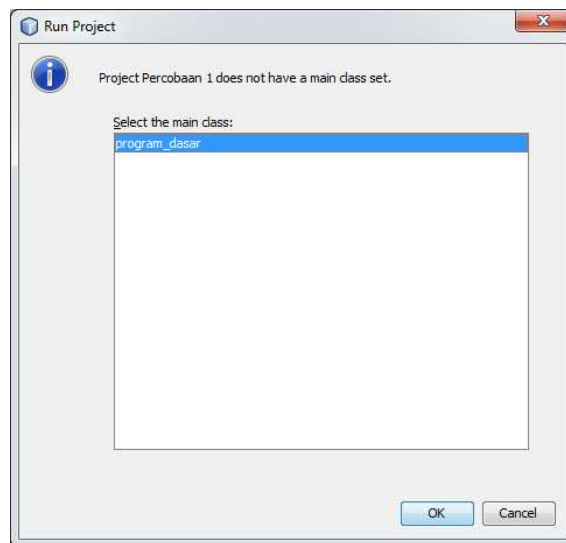


Apabila muncul keterangan BUILD SUCCESSFUL maka build berhasil dilakukan dan menuju langkah berikutnya.

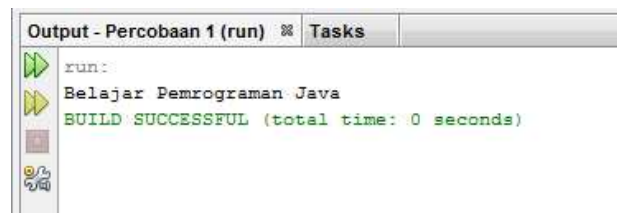
Selanjutnya tekan Clean and Build Project  atau Sift+F11 dan hasilnya seperti di bawah ini.



Dengan keterangan BUILD SUCCESSFUL Clean and Build Project berhasil dilakukan. Setelah itu baru run program dengan menekan tombol Run Project  atau F6 untuk menjalankan program dengan tampilan seperti di bawah ini.



Tekan OK.



Kompilasi menggunakan command prompt.

Masuk pada folder src project yang akan dikompilasi. Ketik javac dan diikuti file .java yang akan dikompilasi.

```
D:\>cd "Bahan Ajar\Java\Program\percobaan 1\src"  
D:\Bahan Ajar\Java\Program\percobaan 1\src>javac program_dasar.java
```

Pada folder src akan dihasilkan file .class

Ketikkan java dan nama class (tanpa .java)

```
D:\Bahan Ajar\Java\Program\percobaan 1\src>java program_dasar  
Belajar Pemrograman Java
```

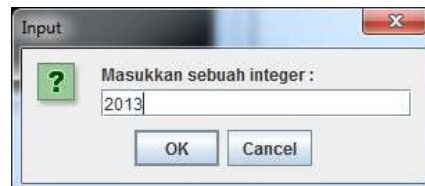
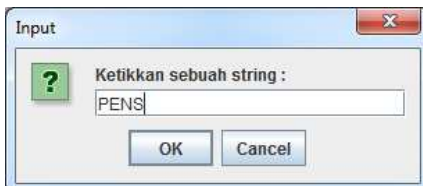
### Baca input keyboard melalui console

```
import java.util.Scanner;
public class baca_input_console {
    //membuat single shared Scanner untuk keyboard input
    private static Scanner scanner = new Scanner( System.in );
    public static void main(String args[]){
        System.out.print("Ketikkan sebuah string : ");
        String input_string = scanner.nextLine();
        System.out.println("String yang anda ketikkan adalah :
"+input_string);
        System.out.print("Ketikkan sebuah integer : ");
        int input_int = scanner.nextInt();
        System.out.println("Integer yang anda ketikkan adalah : "+input_int);
    }
}
```

```
Ketikkan sebuah string : PENS
String yang anda ketikkan adalah : PENS
Ketikkan sebuah integer : 2013
Integer yang anda ketikkan adalah : 2013
```

### Baca input keyboard melalui GUI

```
import javax.swing.JOptionPane;
public class baca_input_gui {
    public static void main(String args[]){
        String input_string = JOptionPane.showInputDialog("Ketikkan sebuah
string :");
        System.out.println("String yang anda ketikkan adalah :
"+input_string);
        String input = JOptionPane.showInputDialog("Masukkan sebuah integer
:");
        //konversi string ke integer
        int input_int = Integer.parseInt(input);
        System.out.println("Integer yang anda ketikkan adalah : "+input_int);
    }
}
```



```
String yang anda ketikkan adalah : PENS
Integer yang anda ketikkan adalah : 2013
```

### 3. TUGAS PENDAHULUAN

1. Jelaskan perbedaan antara JRE dengan JDK ?
2. Buat program untuk konversi suhu dari Celcius ke Fahrenheit. Dengan nilai celcius inputan dari keyboard.

$$Fahrenheit = \frac{Celcius \times 9}{5} + 32$$

3. Buatlah program untuk menghitung luas lingkaran dengan jari-jari diinputkan melalui keyboard.