

## BAB 3

# Mengenali Lingkup Pemrograman Anda

### 3.1 Tujuan

Dalam bagian ini, kita akan membahas tentang bagaimana menulis, mengkompilasi dan menjalankan program Java. Terdapat dua cara dalam melakukannya, yang pertama menggunakan console dan text editor. Sedangkan cara kedua adalah dengan menggunakan NetBeans yang merupakan **Integrated Development Environment** atau yang disebut **IDE**.

Pada akhir pembahasan, diharapkan pembaca dapat :

- Membuat program Java menggunakan text editor dan console dalam Linux (Ubuntu Dapper)
- Membedakan antara syntax-errors dan runtime-errors
- Membuat program Java menggunakan NetBeans

### 3.2 Pendahuluan

IDE adalah sebuah peralatan dalam pemrograman yang diintegrasikan ke dalam aplikasi software yang mendukung pembangunan GUI, teks atau penulisan kode, compiler dan debugger.

Tutorial ini menggunakan Ubuntu Dapper sebagai sistem operasinya. Sebelum menggunakan tutorial ini, pastikan bahwa telah menginstal Java dan NetBeans dalam sistem yang Anda gunakan. Untuk instruksi bagaimana cara menginstal Java dan NetBeans, dapat dilihat pada **Appendix A**. Untuk versi Windows XP dalam bagian ini, dapat dilihat pada **Appendix B**.

Sebelum membahas lebih terperinci, pada awalnya akan kita lihat program Java pertama yang akan Anda tulis.

### 3.3 Program Java Pertama

```
public class Hello
{
    /**
     * My first java program
     */
    public static void main(String[] args) {

        //Menampilkan kata "Hello world" dilayar
        System.out.println("Hello world!");

    }
}
```

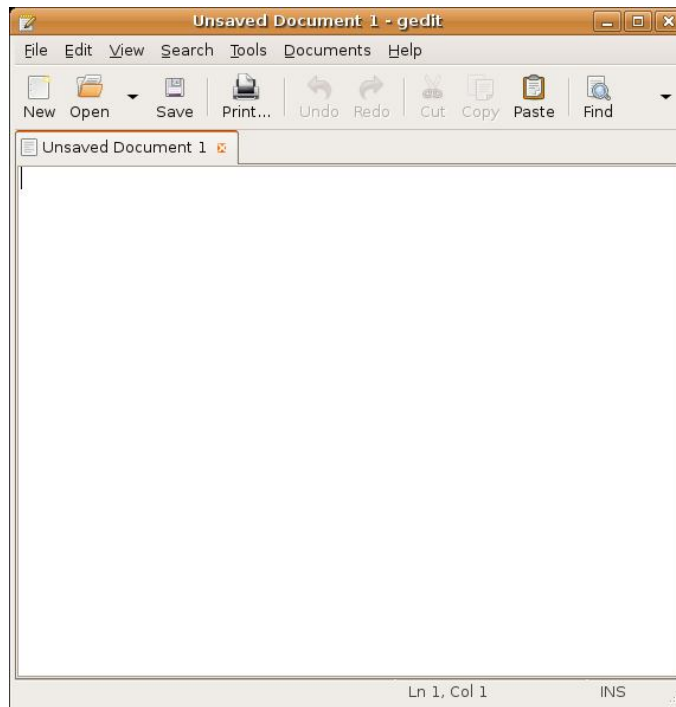
Sebelum menjelaskan apa arti dari program tersebut, cobalah untuk menulis program ini di dalam file dan kemudian menjalankannya.

### 3.4 Menggunakan Text Editor dan Console

Dalam contoh ini, kita akan menggunakan text editor untuk mengedit program Java. Anda juga akan membuka terminal window untuk mengkompilasi dan mengeksekusi program Java Anda.

#### Langkah 1: Memulai Text Editor

Untuk memulai text editor di Linux , klik pada Applications->Accessories->Text Editor.



Gambar 1: Aplikasi Text Editor di Linux

**Langkah 2: Membuka Terminal**

Untuk membuka terminal di Linux, klik pada Applications-> Accessories-> Terminal.

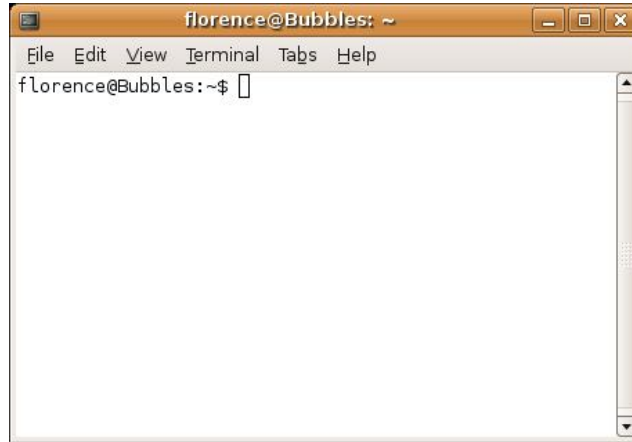
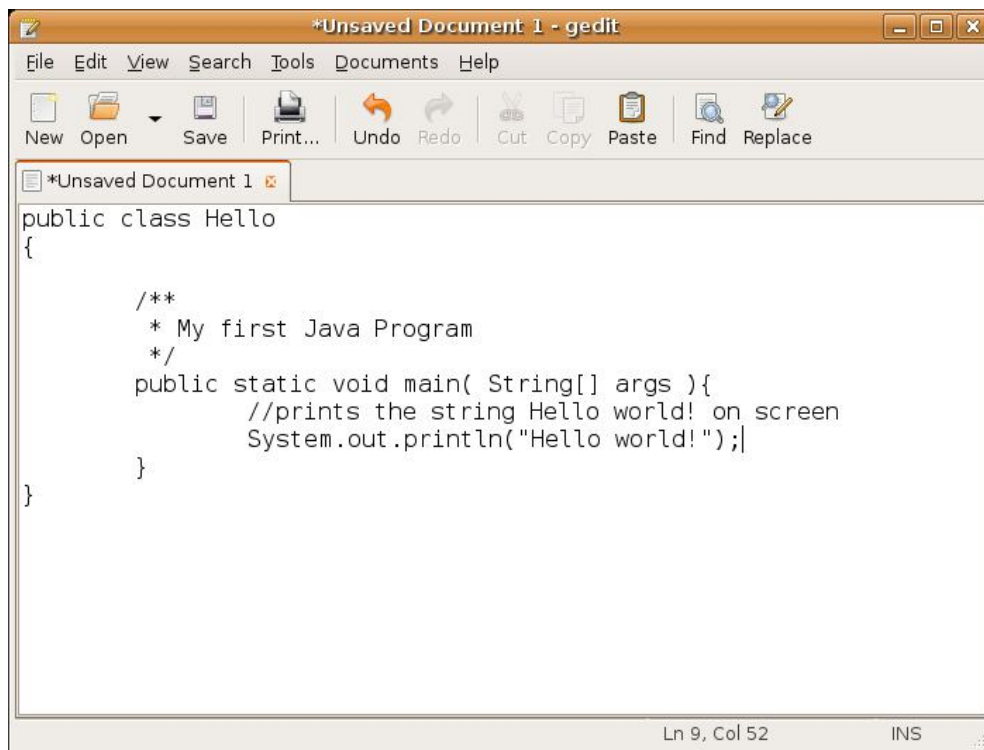


Figure 2: Terminal pada Linux

**Langkah 3: Tulislah listing program Java Anda dalam text editor.**

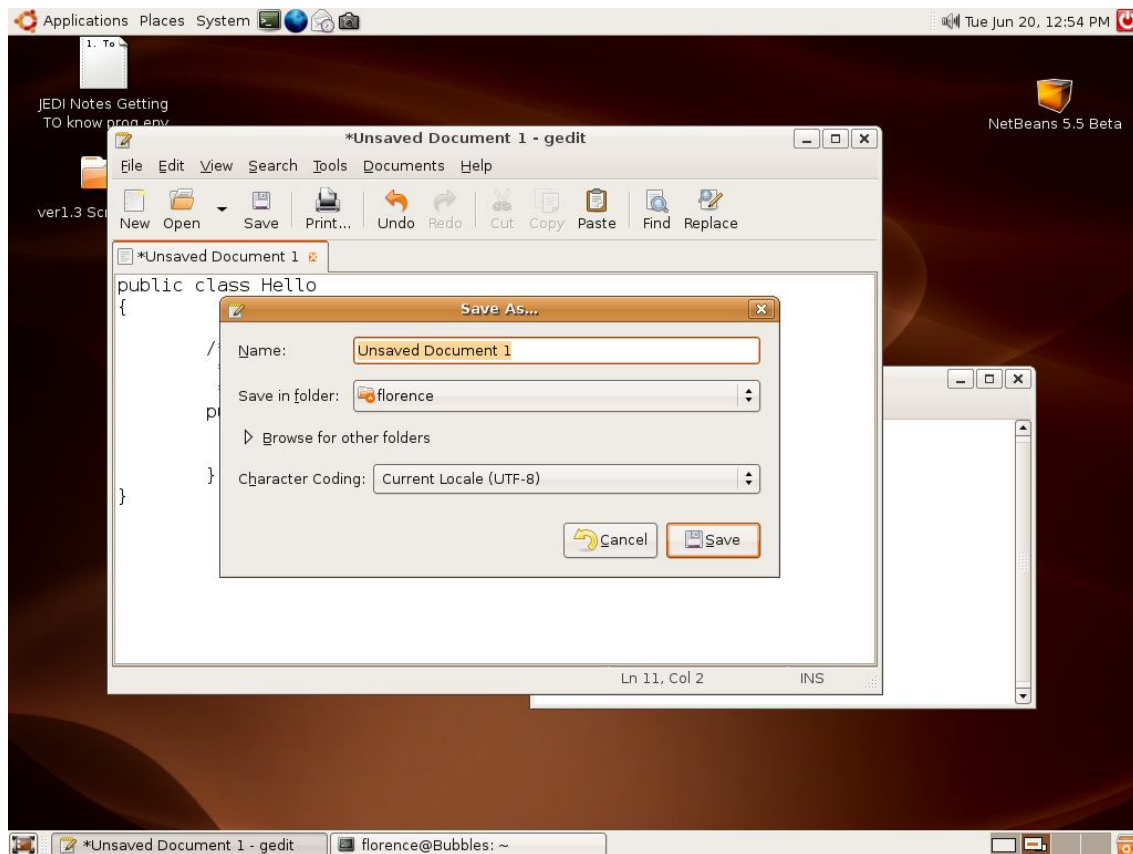
Gambar 3: Menulis Kode Program Menggunakan Text Editor

#### Langkah 4: Simpanlah program Java Anda

Program ini akan disimpan dalam file yang bernama "Hello.java", dan disimpan dalam sebuah folder yang bernama MYJAVAPROGRAMS.

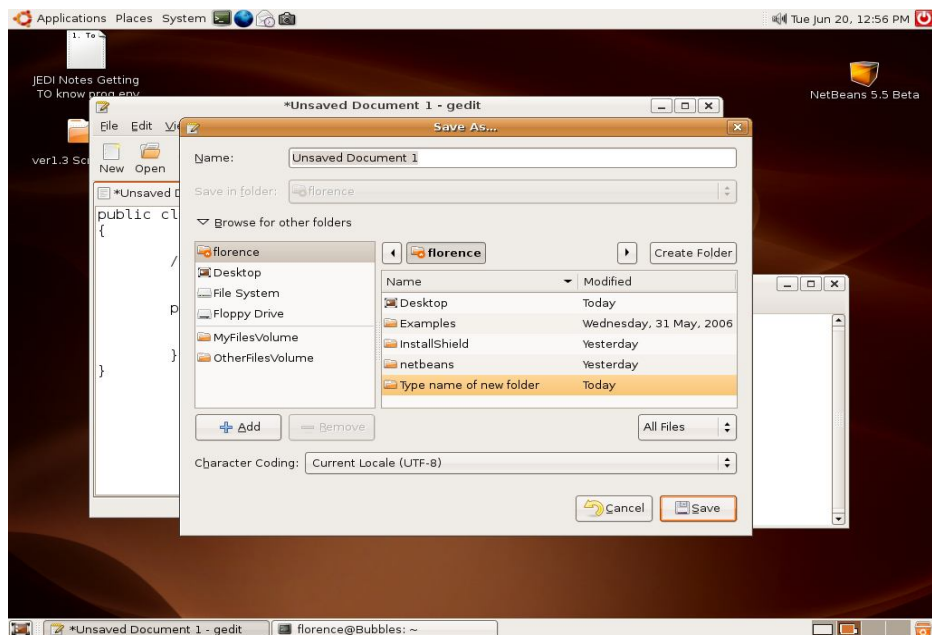
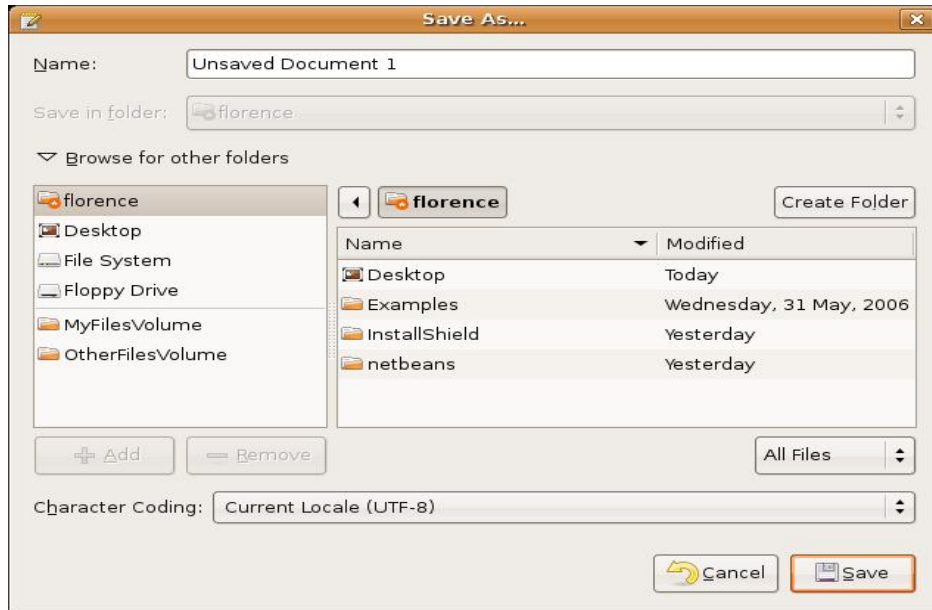
Untuk membuka **Save** dialog box, klik pada File menu yang terdapat pada menubar dan kemudian klik save.

Setelah melakukan instruksi tersebut, dialog box akan tampil seperti gambar dibawah ini.

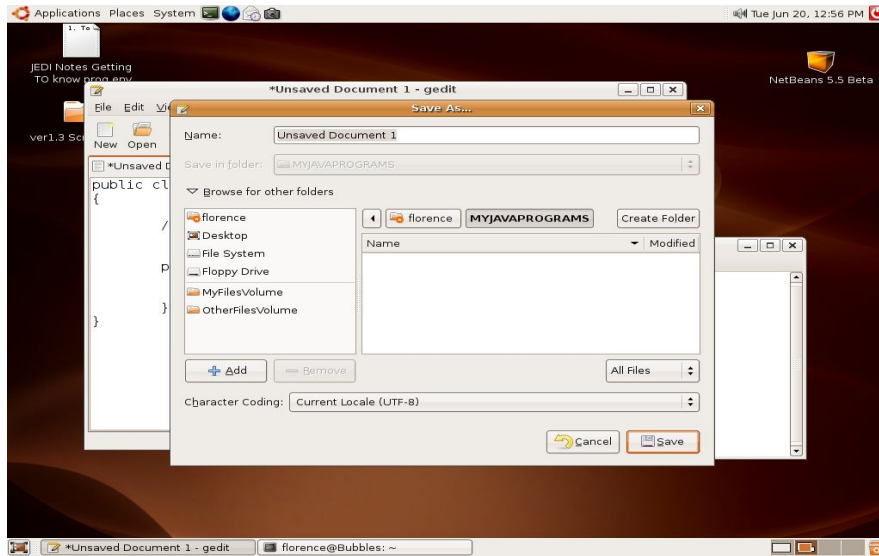


Gambar 4: Menyimpan File

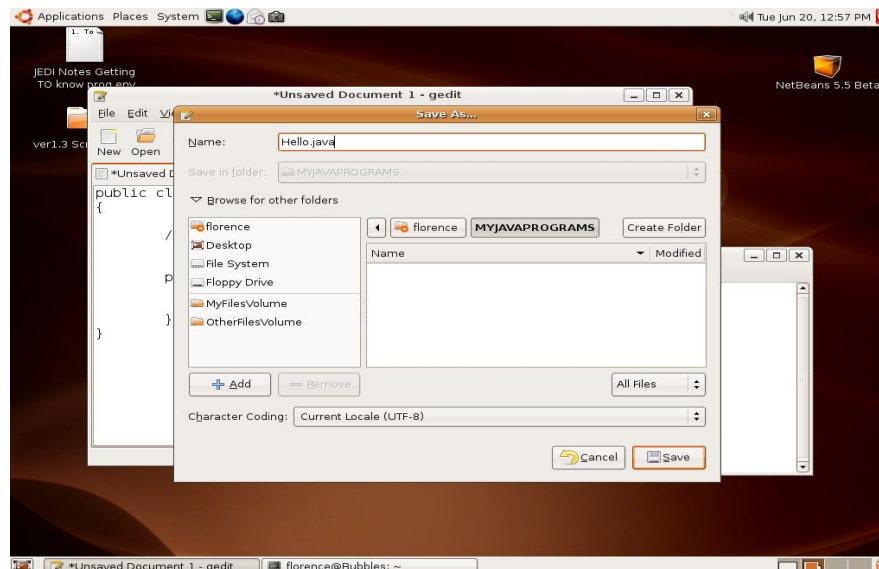
Klik pada tombol browse, kemudian klik tombol Create Folder.



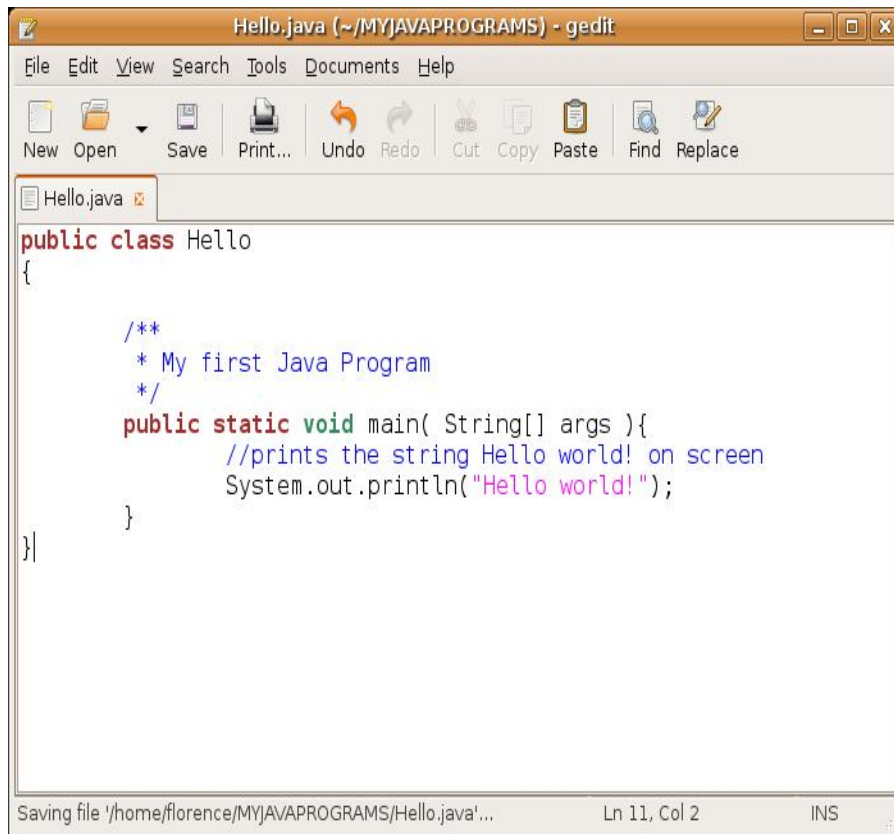
Nama folder baru, MYJAVAPROGRAMS. Sekarang, klik pada folder MYJAVAPROGRAMS untuk mengetahui isi folder yang ada didalamnya. Setelah Anda klik pada folder tersebut, Anda akan melihat gambar yang ditampilkan seperti berikut ini. Untuk saat ini folder tersebut harus kosong sampai terbuatnya folder baru dan kita akan menyimpan semuanya didalam folder tersebut.



Sekarang, dalam textbox Selection, ketiklah nama file dari program Anda, yaitu "Hello.java", kemudian klik pada tombol SAVE.



Sekarang Anda telah menyimpan file Anda, ingatlah cara bagaimana nama frame dapat berubah dari "Untitled Document 1 (modified) – gedit" menjadi "Hello.java (~ /MYJAVAPROGRAMS) – gedit". Jika Anda ingin melakukan perubahan pada file Anda, Anda hanya perlu mengeditnya, kemudian menyimpannya kembali dengan mengklik pada File->Save.



```
public class Hello
{
    /**
     * My first Java Program
     */
    public static void main( String[] args ){
        //prints the string Hello world! on screen
        System.out.println("Hello world!");
    }
}
```

Gambar 5: Jendela Baru setelah Penyimpanan

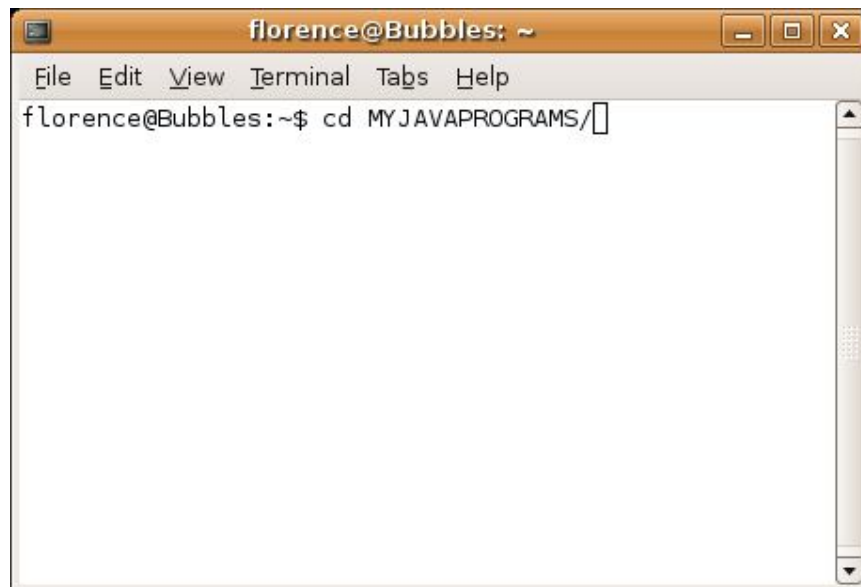
### Langkah 5: Mengkompilasi program Anda

Langkah berikut ini adalah mengkompilasi program Anda. Masuklah ke Terminal Window yang telah kita buka sebelumnya.

Ketika Anda membuka terminal window, terminal tersebut menampilkan jendela yang disebut sebagai **home folder** Anda. Untuk melihat isi dari folder tersebut, ketik **ls** dan kemudian tekan ENTER. Apa yang akan Anda temukan adalah daftar file-file dan folder di dalamnya.

Disinilah Anda dapat melihat folder dengan nama "MYJAVAPROGRAMS" yang telah kita buat sebelumnya, dan merupakan tempat dimana kita menyimpan program Hello.java kita. Kemudian masuklah ke dalam direktori tersebut.


Untuk masuk ke dalam sebuah direktori, ketiklah perintah : **cd [directory name]**. Perintah "cd" digunakan untuk merubah direktori. Dalam hal ini, nama dari direktori kita adalah MYJAVAPROGRAM, maka Anda harus mengetik : **cd MYJAVAPROGRAMS**.



Gambar 6: Merubah Direktori



Setelah berada di dalam folder yang berisi program Java Anda, sudah saatnya untuk memulai meng-compile program Java tersebut. Sebagai catatan, bahwa file yang Anda maksud harus ada didalam folder yang Anda pilih. Periksalah dengan menjalankan perintah "ls" lagi untuk melihat bahwa file Anda ada didalam folder tersebut.



```
florence@Bubbles: ~/MYJAVAPROGRAMS
File Edit View Terminal Tabs Help
florence@Bubbles:~$ cd MYJAVAPROGRAMS/
florence@Bubbles:~/MYJAVAPROGRAMS$ ls
Hello.java
florence@Bubbles:~/MYJAVAPROGRAMS$
```

Gambar 7: Daftar File didalam Direktori yang Baru

Untuk mengkompilasi program Java, gunakanlah perintah : **javac [filename]**. Dalam hal ini, Anda mengetikkan perintah : **javac Hello.Java**.



```
florence@Bubbles: ~/MYJAVAPROGRAMS
File Edit View Terminal Tabs Help
florence@Bubbles:~$ cd MYJAVAPROGRAMS/
florence@Bubbles:~/MYJAVAPROGRAMS$ ls
Hello.java
florence@Bubbles:~/MYJAVAPROGRAMS$ javac Hello.java
florence@Bubbles:~/MYJAVAPROGRAMS$ ls
Hello.class Hello.java
florence@Bubbles:~/MYJAVAPROGRAMS$
```

Gambar 8: Mengkompilasi File Java


Selama pengkompilasian, javac menambah file ke dalam disk yang disebut **[filename].class**, atau dalam hal ini, **Hello.class**, yang merupakan kode sebenarnya.

#### Langkah 6 : Menjalakan Program

Selama proses pengkompilasian, javac menambah file baru kedalam disk yang disebut **[filename].class**, dalam hal ini, **Hello.class**, yang merupakan kode sebenarnya saat ini, dengan perkiraan tidak ada permasalahan pada saat proses compile (kita akan menyelidiki dan membahas permasalahan yang ditemukan selama proses compile, pada bagian berikutnya), berarti telah siap untuk menjalankan program Anda.

Untuk menjalankan program Java Anda, dengan mengetikkan perintah : **java [filename without the extension]**, maka dalam hal ini, Anda mengetikkan perintah : **java Hello**.

Sekarang Anda dapat melihat di layar bahwa Anda telah menjalankan program Java pertama Anda, yang menampilkan pesan, "Hello world!".

A screenshot of a terminal window titled "florence@Bubbles: ~/MYJAVAPROGRAMS". The terminal shows the following sequence of commands and output:

```
florence@Bubbles:~$ cd MYJAVAPROGRAMS/
florence@Bubbles:~/MYJAVAPROGRAMS$ ls
Hello.java
florence@Bubbles:~/MYJAVAPROGRAMS$ javac Hello.java
florence@Bubbles:~/MYJAVAPROGRAMS$ ls
Hello.class Hello.java
florence@Bubbles:~/MYJAVAPROGRAMS$ java Hello
Hello world!
florence@Bubbles:~/MYJAVAPROGRAMS$
```

Gambar 9: Menjalakan File Class

### 3.4.1 Errors

Apa yang telah ditunjukkan sejauh ini adalah program Java dimana kita tidak menemukan berbagai masalah dalam proses kompilasi maupun pada saat menjalankannya. Bagaimanapun juga, hal ini tidak selalu terjadi. Seperti yang telah kita diskusikan pada bagian pertama dari latihan ini, biasanya kita juga akan menemukan permasalahan selama proses berlangsung.

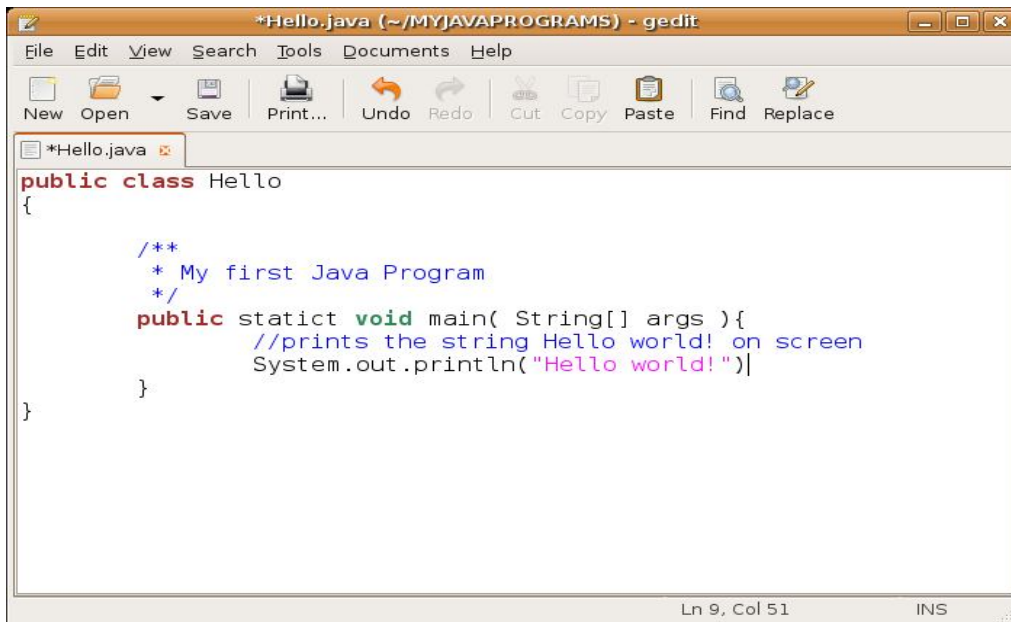
*Seperti yang telah didiskusikan sebelumnya, ada 2 macam error. Pertama adalah compile-time error atau yang disebut juga sebagai syntax error. Kedua adalah runtime errors.*

#### 3.4.1.1 Syntax Errors

Syntax errors biasanya terjadi karena kesalahan penulisan. Mungkin Anda kekurangan sebuah perintah di Java atau lupa untuk menulis tanda titik-koma pada akhir pernyataan. Java mencoba untuk mengisolasi error tersebut dengan cara menunjukkan baris dari kode dan terlebih dahulu menunjuk karakter yang salah dalam baris tersebut. Bagaimanapun juga, error belum tentu berada pada titik yang ditunjuk.

Kesalahan umum lainnya adalah dalam kapitalisasi, ejaan, penggunaan dari karakter khusus yang tidak benar, dan penghilangan dari pemberian tanda baca yang sebenarnya.

Mari kita mengambil contoh, pada program Hello.java, dimana dengan sengaja kita menghilangkan titik-koma pada akhir pernyataan dan juga mencoba untuk mengetikkan ejaan yang salah pada sebuah perintah.




```
public class Hello
{
    /**
     * My first Java Program
     */
    public static void main( String[] args ){
        //prints the string Hello world! on screen
        System.out.println("Hello world!")
    }
}
```

Gambar 10: Listing Program yang Memiliki Error

Lihatlah pesan error yang ditampilkan setelah peng-compile-an program dijalankan. Pesan error yang pertama memberitahu bahwa di program Anda terdapat error pada baris 6. Hal itu menunjuk pada kata setelah **statict**, dimana seharusnya dieja sebagai static.

Pesan error yang kedua memberitahukan bahwa pada program tersebut kehilangan titik-koma setelah pernyataan Anda.



```
florence@Bubbles: ~/MYJAVAPROGRAMS
File Edit View Terminal Tabs Help
florence@Bubbles:~/MYJAVAPROGRAMS$ javac Hello.java
Hello.java:7: <identifier> expected
    public statict void main( String[] args ){
           ^
Hello.java:11: ';' expected
}
^
2 errors
florence@Bubbles:~/MYJAVAPROGRAMS$
```

Gambar 11: Pengkompilasian Listing Program yang Memiliki Error

Jika Anda menemukan banyak pesan error, cobalah untuk mengoreksi kesalahan yang pertama pada daftar error satu per satu, dan mencoba untuk mengkompilasinya kembali. Dengan melakukan hal tersebut maka, dapat mengurangi total error yang ada.

### 3.4.1.2 Run-time Errors

Run-time error merupakan error yang tidak akan ditampilkan sampai Anda menjalankan program Anda. Bahkan program yang dikompilasi dengan sukses dapat menampilkan jawaban yang salah jika programmer belum berpikir sampai struktur dan proses logis dari program tersebut.

## 3.5 Menggunakan NetBeans

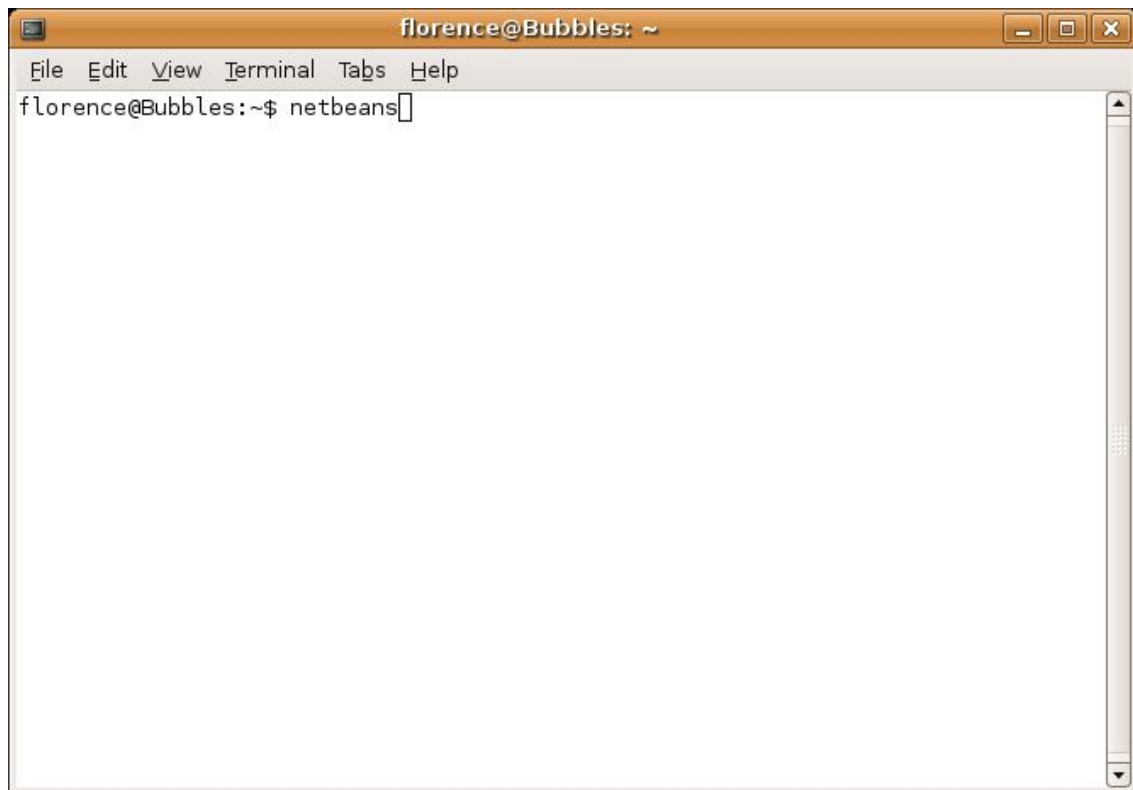
Sekarang kita mencoba mengerjakan program kita dengan jalan yang lebih rumit, mari kita lihat bagaimana mengerjakan semua proses yang telah dijelaskan pada bagian sebelumnya dengan menggunakan sebuah aplikasi.

Pada bagian pelajaran ini, kita akan menggunakan **NetBeans**, yang merupakan **Integrated Development Environment** atau **IDE**. Suatu IDE adalah lingkup pemrograman yang diintegrasikan kedalam suatu aplikasi perangkat lunak yang menyediakan pembangun GUI, suatu text atau kode editor, suatu compiler atau interpreter dan suatu debugger.

### Langkah 1 : NetBeans

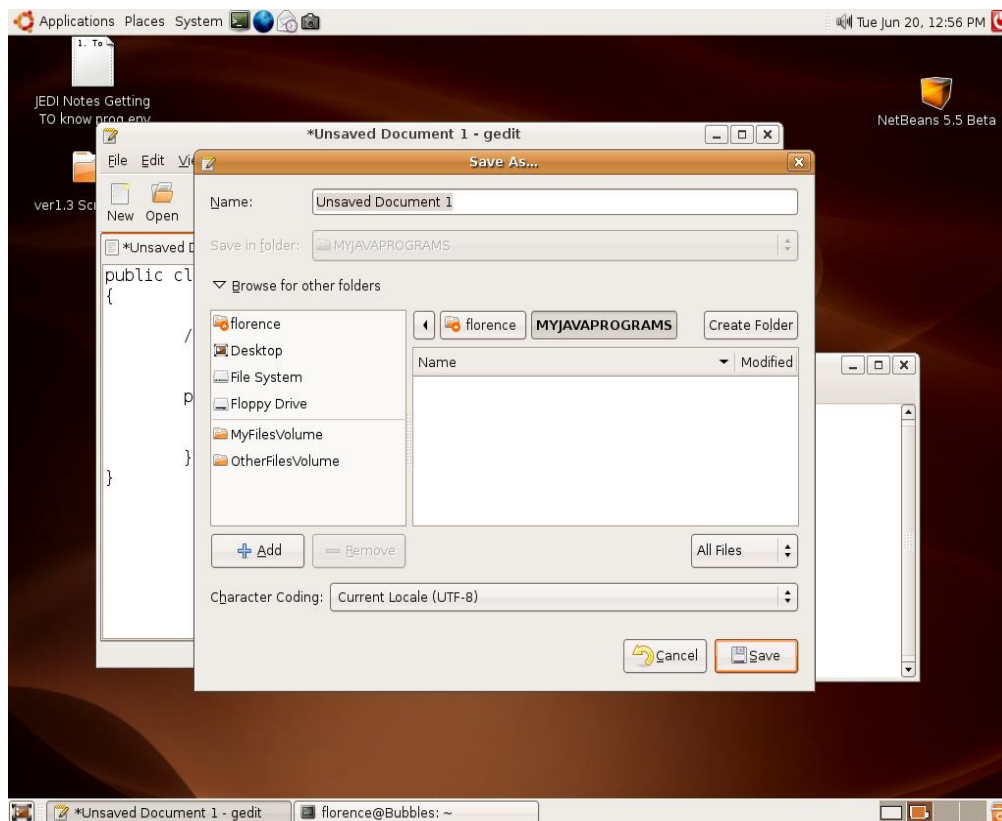
Ada dua cara untuk menjalankan NetBeans. Pertama menulis perintah menggunakan terminal, atau hanya dengan mengklik pada tombol shortcut yang dapat ditemukan di desktop.

Untuk menjalankan NetBeans menggunakan command-line. Bukalah terminal (lihat langkahnya pada bagaimana menjalankan terminal didiskusi sebelumnya), dan ketiklah : **netbeans**.



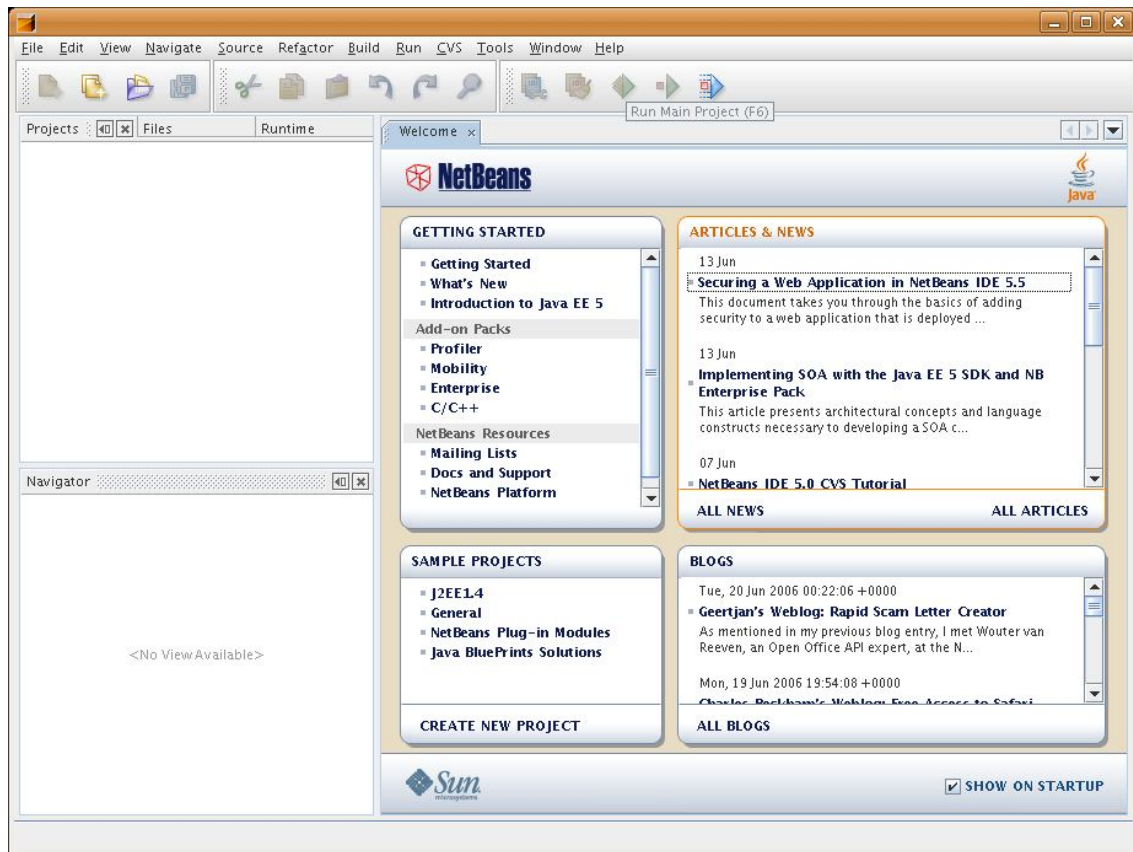
Gambar 12: Menjalankan NetBeans Menggunakan Command-Line

Cara kedua untuk menjalankan NetBeans, adalah dengan mengklik shortcut icon yang terdapat pada desktop Anda.



Gambar 13: Menjalankan NetBeans Menggunakan Shortcut Icon di Desktop

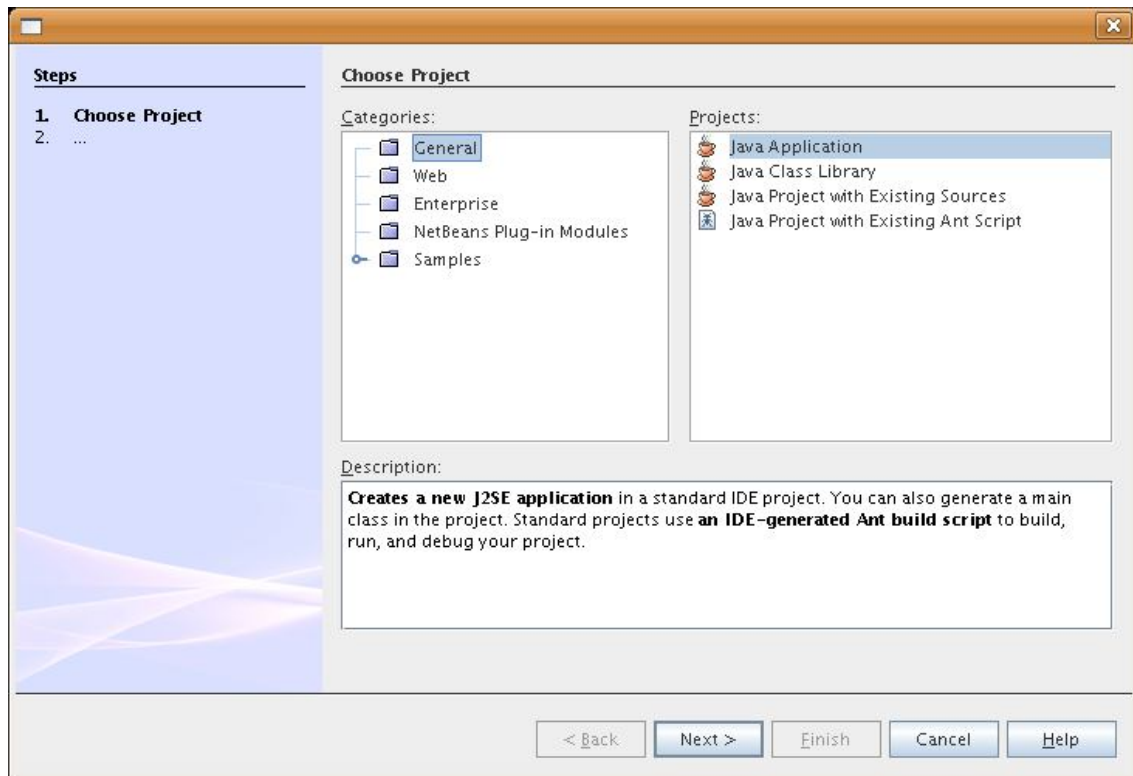
Setelah Anda membuka NetBeans IDE, Anda akan melihat graphical user interface (GUI) seperti yang ditunjukkan dibawah ini.



Gambar 14: Window Setelah Membuka NetBeans

## Langkah 2: Membuat sebuah Project

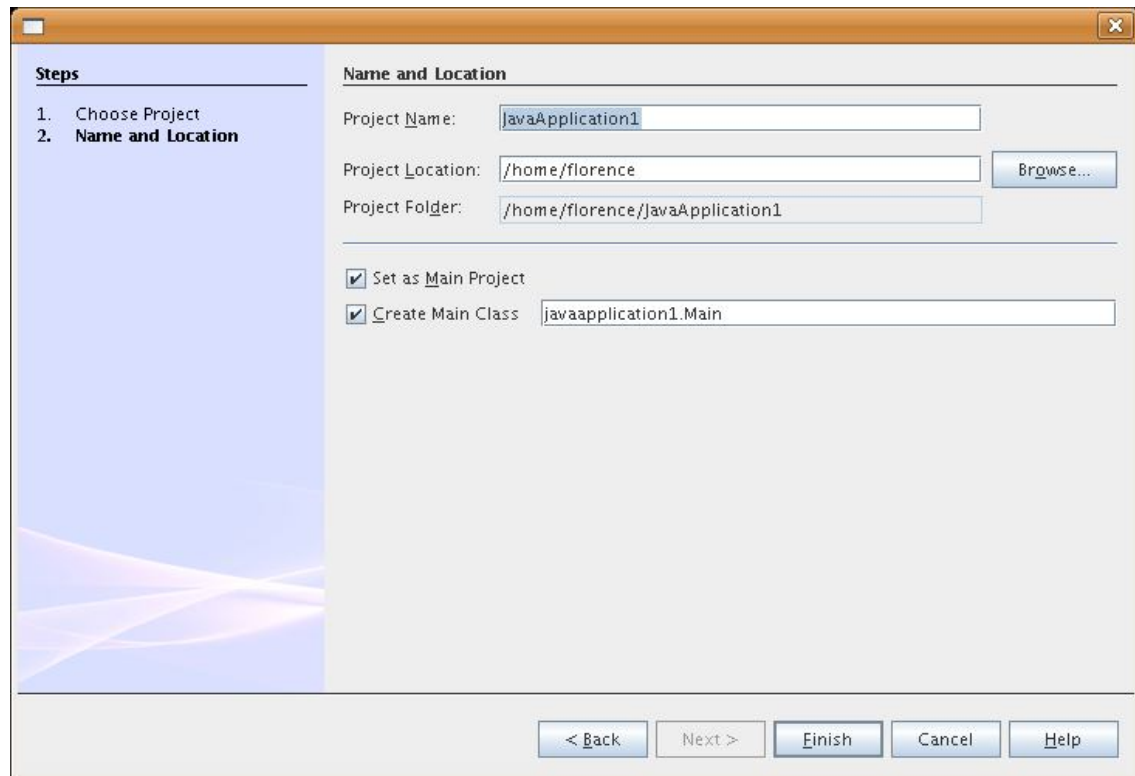
Pertama, mari kita buat sebuah project. Klik pada File->New Project. Setelah melakukannya, akan ditampilkan kotak dialog project yang baru. Kemudian klik pada Java Application dan klik tombol NEXT.



Gambar 15: Memilih Tipe Project

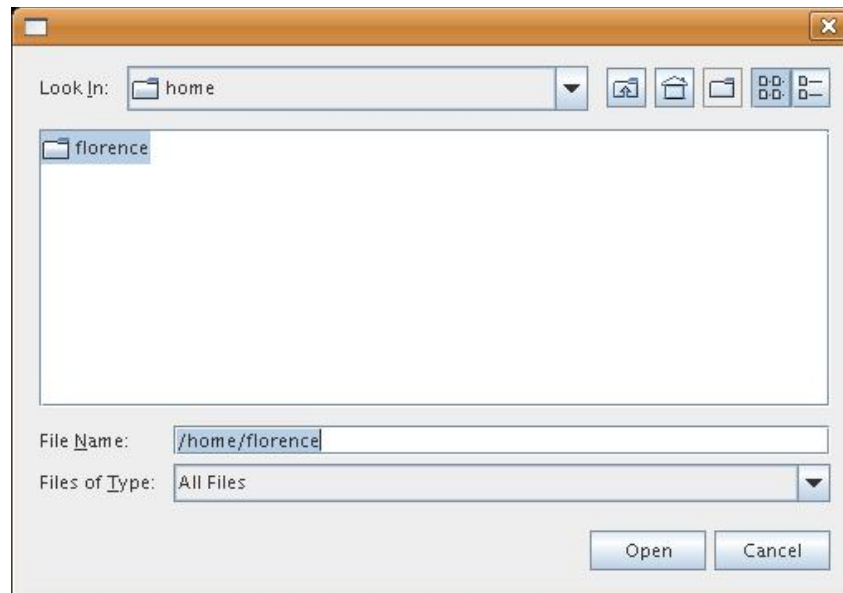


Sekarang, dialog aplikasi yang baru akan ditampilkan.



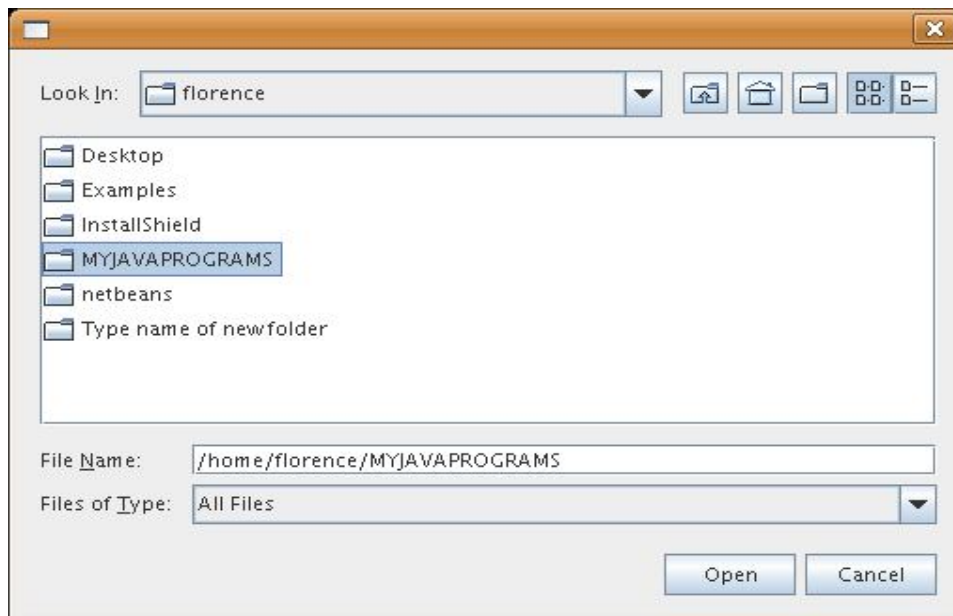
Gambar 16: Mengatur Informasi Project

Cobalah untuk mengubah lokasi aplikasi, dengan mengklik pada tombol BROWSE. Kemudian dialog lokasi project akan muncul. Klik dua kali pada home folder Anda.



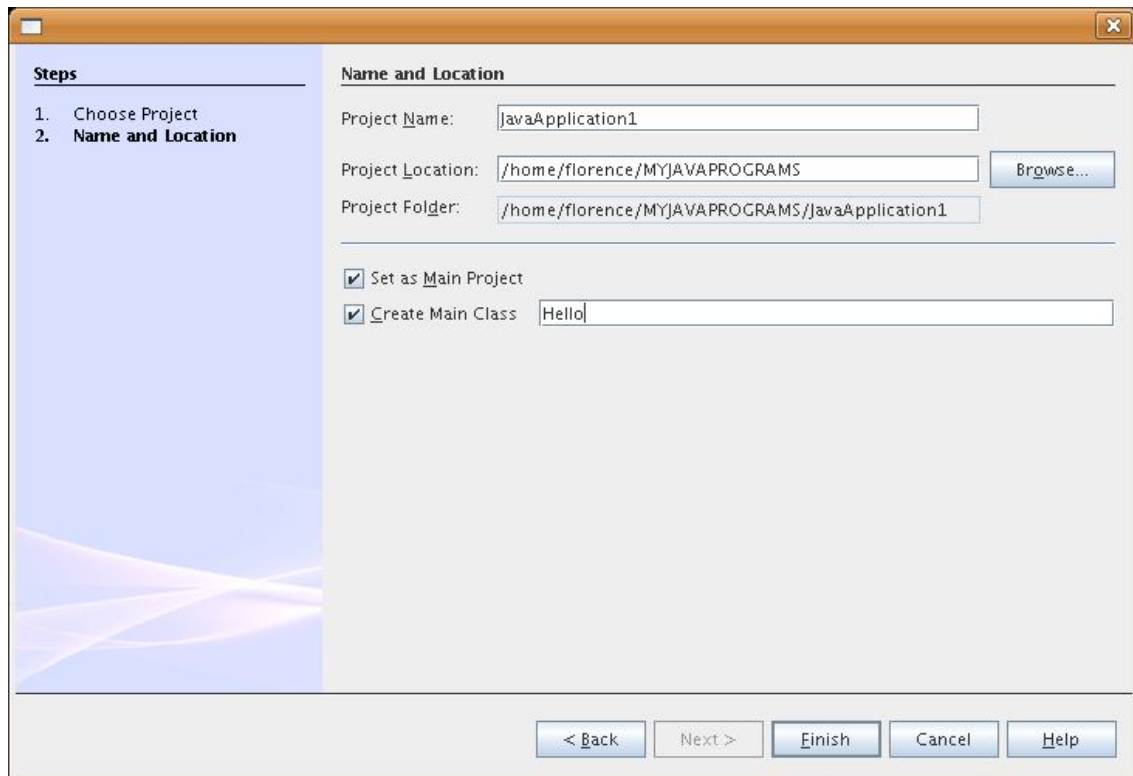
Gambar 17: Mengatur Lokasi Project

Kemudian isi dari folder root akan ditampilkan. Lalu klik dua kali pada folder MYJAVAPROGRAMS dan klik tombol OPEN.



Lihatlah Project Location dan Project Folder telah berubah menjadi /home/florence/MYJAVAPROGRAMS.

Yang terakhir, pada textfield Create Main Class, tulislah Hello sebagai nama kelas utama, dan kemudian klik tombol FINISH.

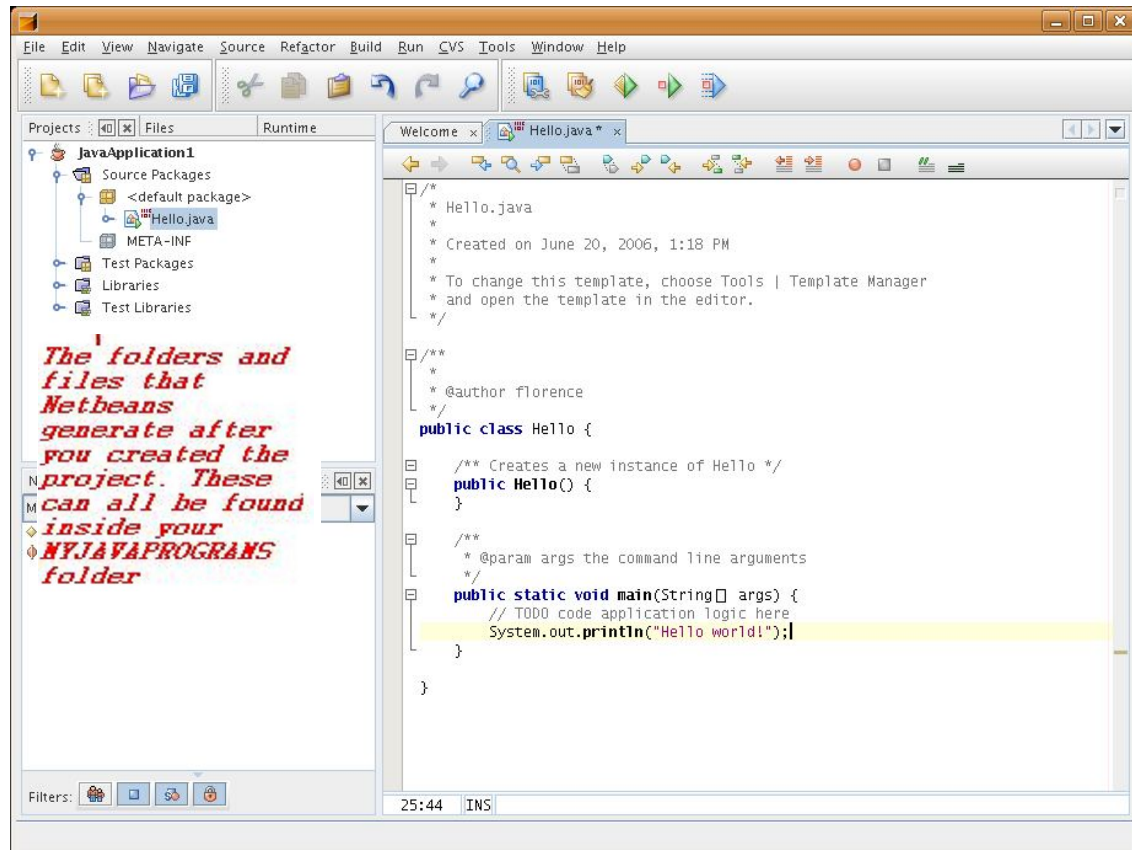


Gambar 18: Window setelah Mengatur Lokasi Project ke MYJAVAPROGRAMS/Mengatur kelas utama project menjadi Hello

### Langkah 3 : Menulis dalam Program Anda

Sebelum menulis program Anda, pertama akan dijelaskan mengenai jendela utama setelah membuat project.

Seperti ditunjukkan di bawah, secara otomatis NetBeans membuat kode dasar untuk program Java Anda. Anda dapat menambah pernyataan-pernyataan Anda pada kode yang telah di-generate. Pada sisi kiri jendela, Anda dapat melihat daftar folder dan file-file yang dihasilkan NetBeans setelah membuat sebuah project. Semua ini dapat Anda temukan dalam folder MYJAVAPROGRAMS Anda, dimana Anda menetapkan lokasi project tersebut.

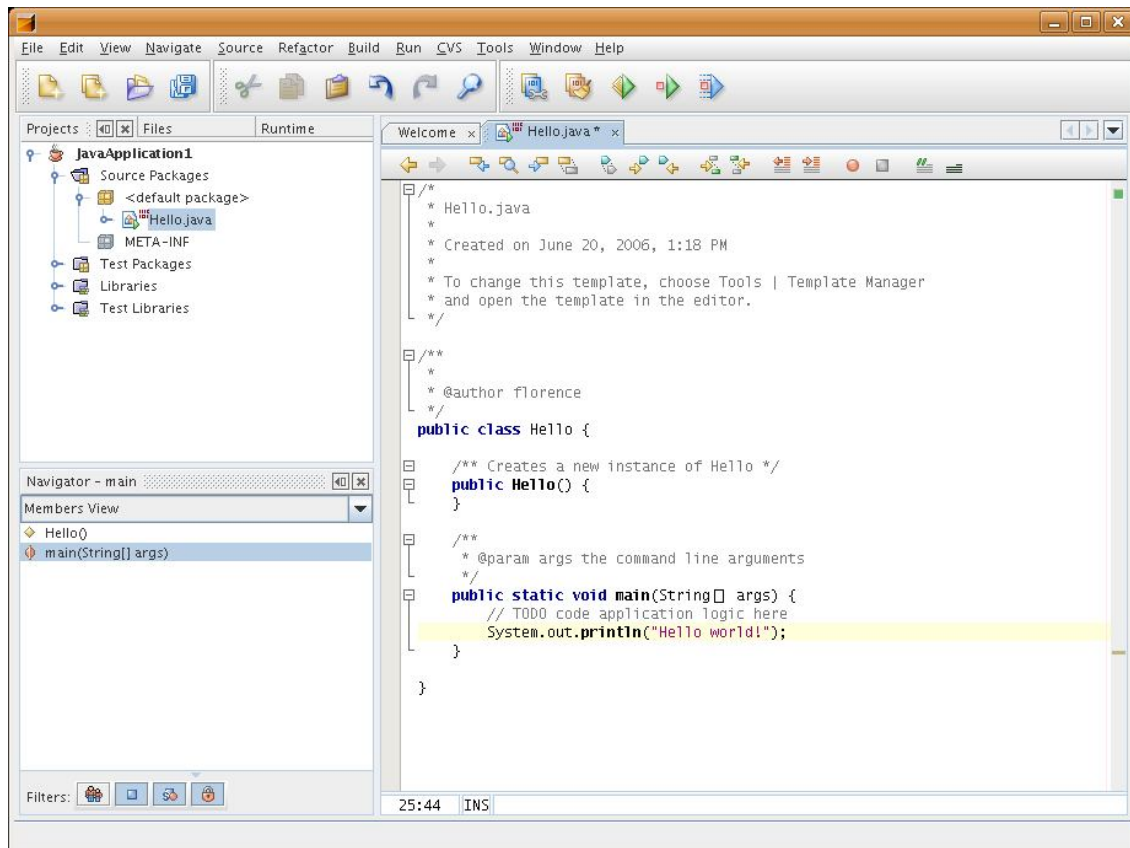


Gambar 19: Tampilan dari Project yang telah Dibuat

Sekarang, cobalah memodifikasi kode yang dihasilkan oleh NetBeans. Untuk saat ini abaikan bagian-bagian lain dari program, detail dari kode tersebut akan dijelaskan pada bagian berikutnya. Masukkan kode :

```
System.out.println("Hello world!");
```

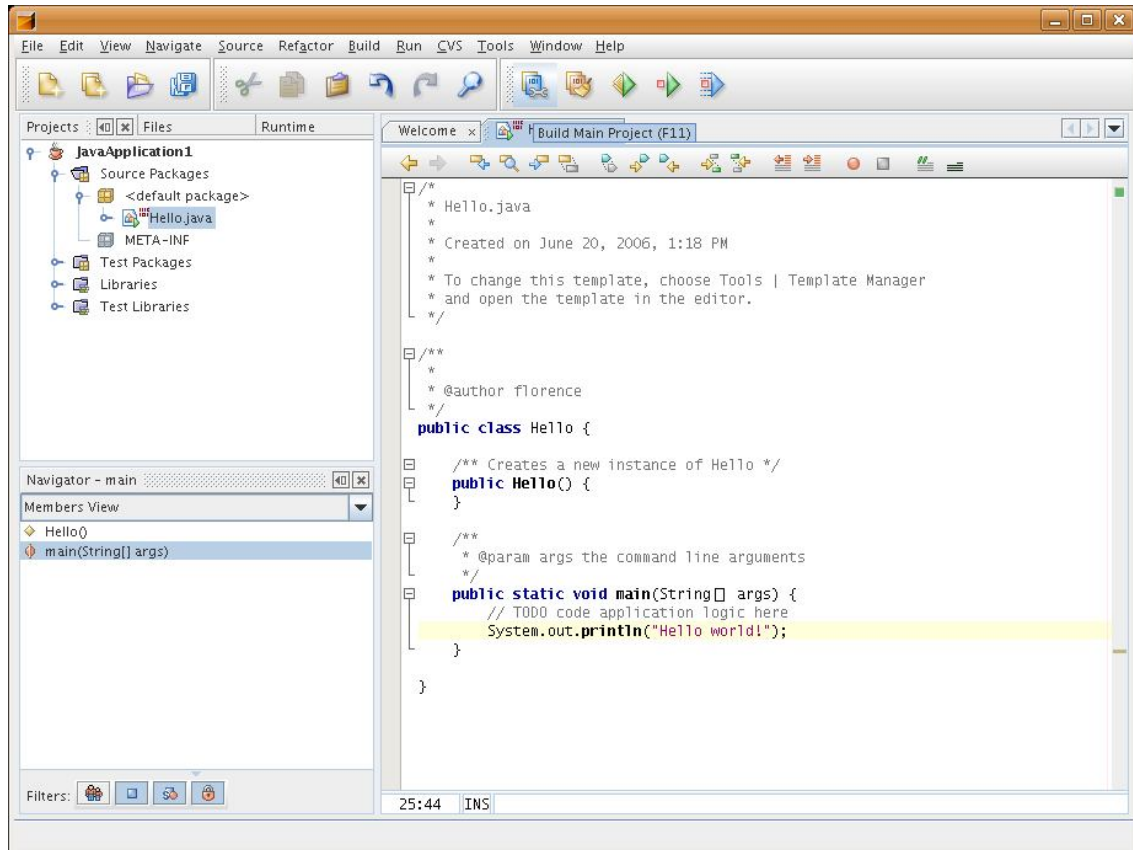
Setelah pernyataan, //TODO code application logic here.



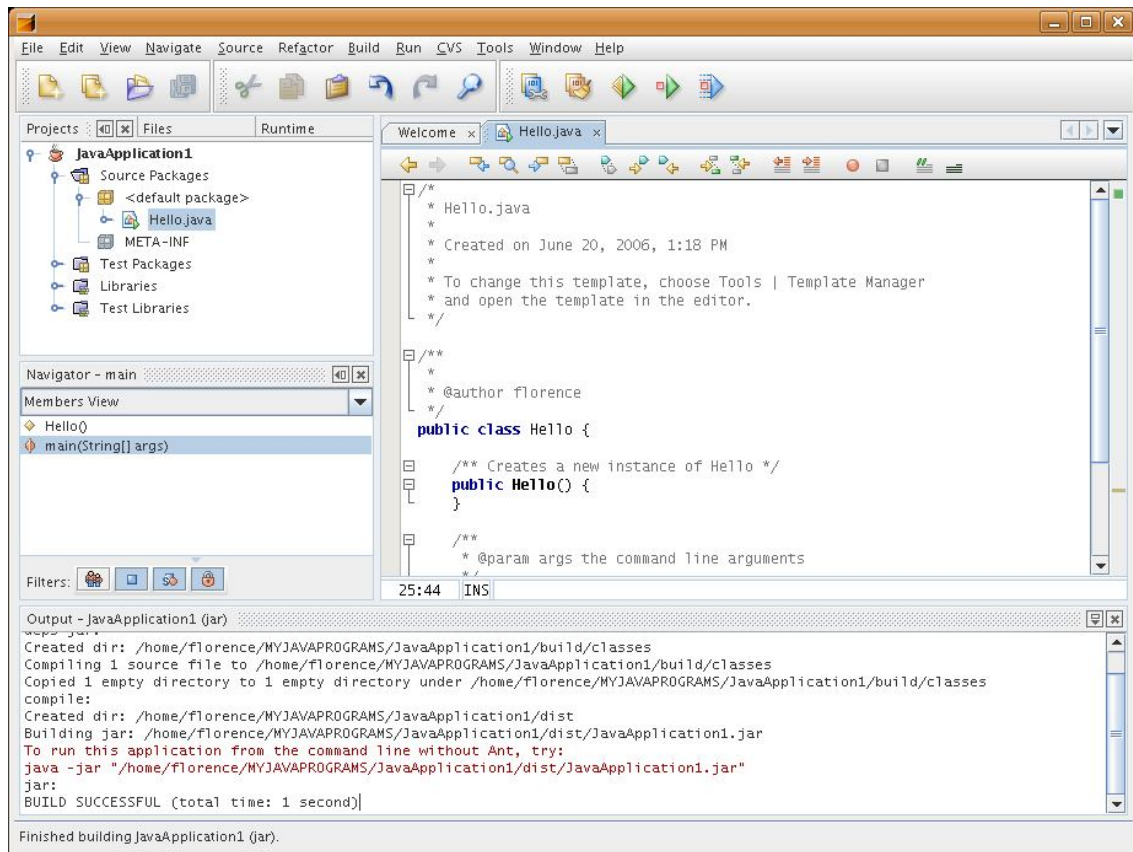
Gambar20: Memasukkan Kode

#### Langkah 4 : Mengkompilasi Program Anda

Sekarang, untuk mengkompilasi program Anda, klik pada Build->Build Main Project. Atau, Anda juga dapat menggunakan tombol shortcut untuk mengkompilasi kode Anda.



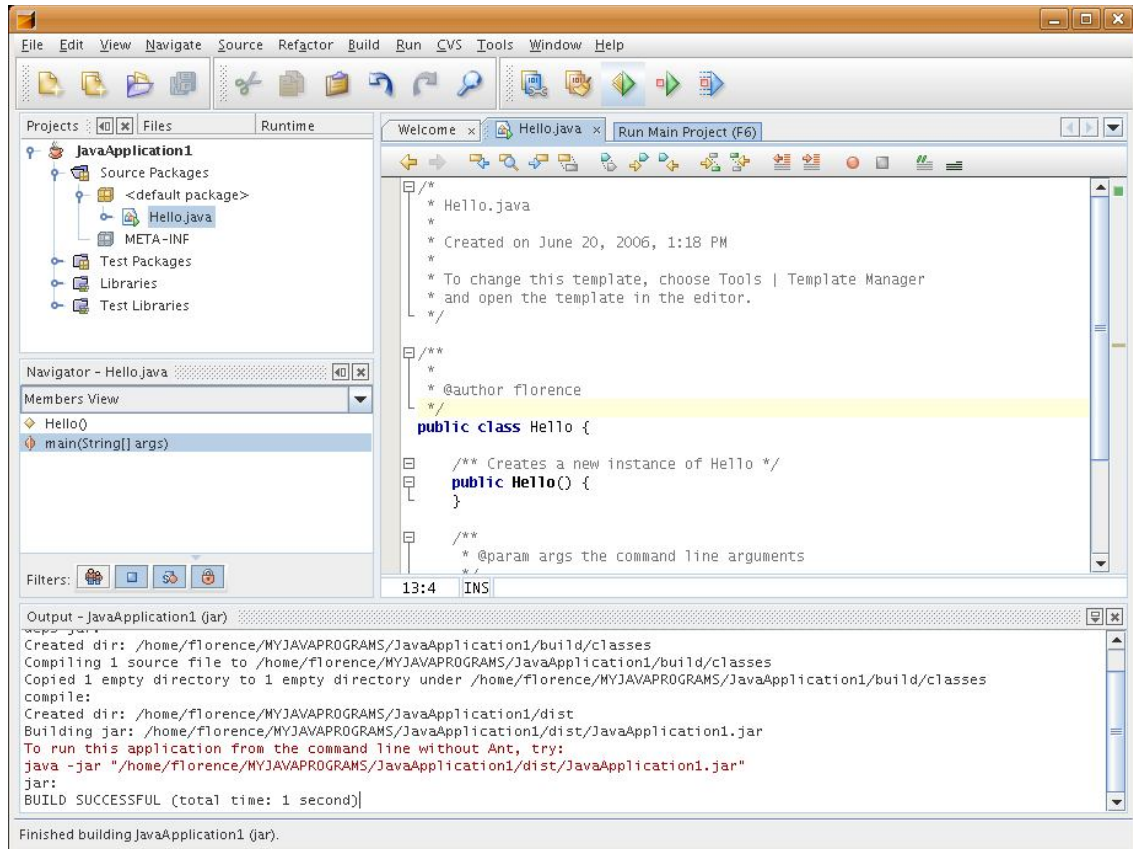
Jika tidak terdapat error pada program Anda, Anda dapat melihat pesan sukses build pada jendela yang dihasilkan.



Gambar 21: Tampilan Setelah sukses melakukan compile

## Langkah 5 : Menjalankan Program Anda

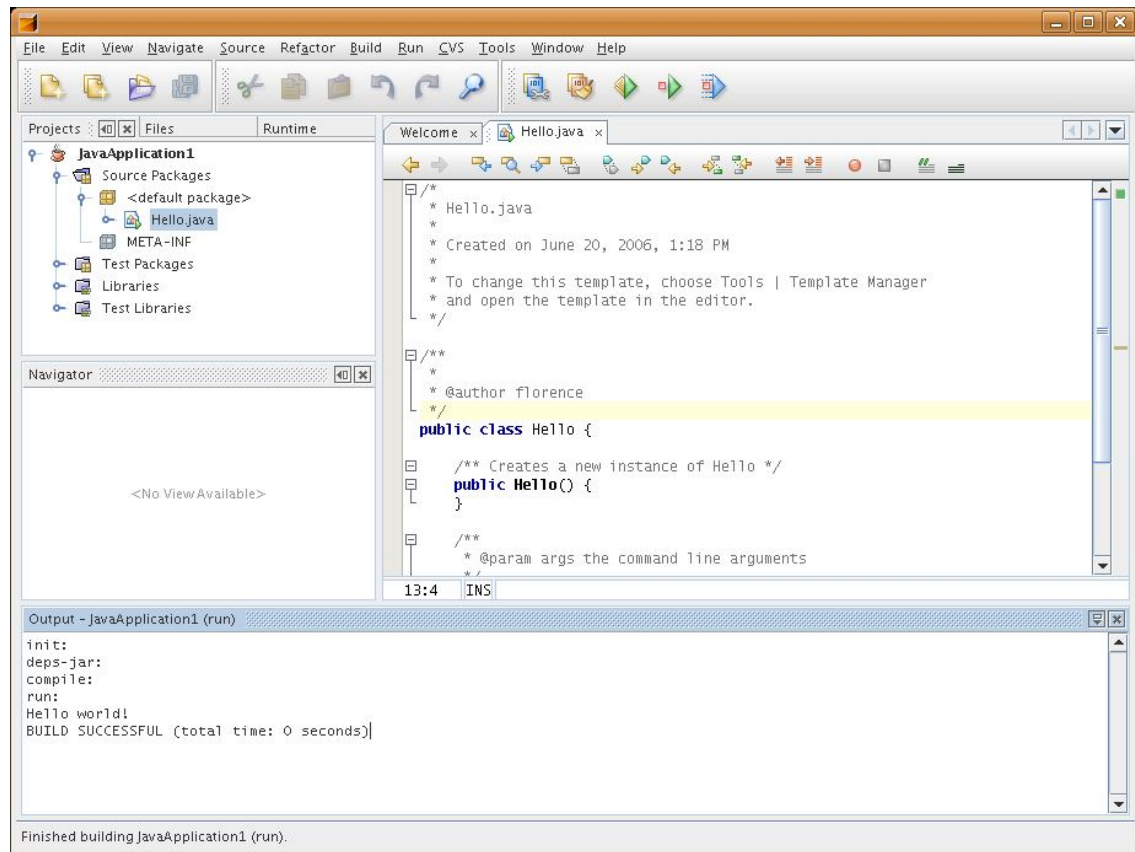
Untuk menjalankan program Anda, klik pada Run->Run Main Project. Atau Anda juga dapat menggunakan tombol shortcut untuk menjalankan program Anda.



Gambar 22: Menjalankan dengan NetBeans



Hasil dari program Anda ditampilkan pada jendela.



Gambar 23: Tampilan Setelah Sukses Menjalankan Program

## 3.6 Latihan

### 3.6.1 Hello World!

Menggunakan NetBeans, membuat class dengan nama : [NamaAnda]. Hasil dari program yang harus tampil di layar :

```
Welcome to Java Programming [NamaAnda]!!!
```

### 3.6.2 The Tree

Menggunakan NetBeans, membuat class dengan nama : TheTree. Hasil dari program yang harus tampil dilayar :

```
I think that I shall never see,  
a poem as lovely as a tree.  
A tree whose hungry mouth is pressed  
Against the Earth's sweet flowing breast.
```