

### JURUSAN TEKNIK INFORMATIKA

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS SANATA DHARMA YOGYAKARTA**

**MODUL 1**

**Pengenalan Pemakaian Netbean dan Program Sederhana**

# A. TUJUAN

1. Mahasiswa mengenal lingkungan pengembangan program Java.
2. Mahasiswa mampu mengetik, mengkompilasi dan menjalankan contoh program dalam Java.

# B. LANDASAN TEORI

**B.1 Fase *Edit***

Fase 1 merupakan fase penyuntingan (*edit*) file program Java menggunakan program *editor*. Dalam fase ini, program Java (yang sering juga disebut *source code*) diketik menggunakan *editor*, dilakukan koreksi-koreksi sejauh diperlukan, dan selanjutnya disimpan dalam media simpanan sekunder, seperti hard disk maupun flash disk. *Source code* Java tersebut diberi nama dengan ekstensi **.java** yang menunjukkan bahwa file tersebut berisi *source code* Java. Salah satu program editor dalam lingkungan sistem operasi Windows yang dapat digunakan untuk membuat program Java adalah **notepad**.

Untuk mengembangkan suatu sistem informasi bagi organisasi, tersedia pula ***integrated development environment (IDE)***, yang diproduksi oleh beberapa *software supplier*. IDE menyediakan alat bantu (*tools*) yang mendukung proses pengembangan perangkat lunak, termasuk ***editor*** untuk mengetik dan mengedit program serta ***debugger*** untuk menemukan *error* dalam program.

Beberapa IDE yang populer diantaranya adalah NetBeans ([www.netbeans.org](http://www.netbeans.org)), jEdit ([www.jedit.org](http://www.jedit.org)), Eclipse ([www.eclipse.org](http://www.eclipse.org)), JBuilder ([www.borland.com](http://www.borland.com)), JCreator ([www.jcreator.com](http://www.jcreator.com)), BlueJ ([www.blueJ.org](http://www.blueJ.org)) dan jGRASP ([www.jgrasp.org](http://www.jgrasp.org)).

**B.2.2 Fase *Compile***

Dalam fase 2, *programmer* menggunakan perintah **javac** (*Java compiler*) untuk mengkompilasi suatu program. Sebagai contoh, untuk mengkompilasi program Welcome.java , kita perlu mengetikkan perintah berikut dalam window terminal sistem yang dipakai (misal *command prompt*)

javac Welcome.java

Jika program telah selesai dikompilasi, maka *Java Compiler* akan menghasilkan file bertipe **.class** yang berisi program versi kompilasi**.** Dalam contoh kita, maka akan dihasilkan file Welcome.class. *Java Compiler* mengubah *Java source code* menjadi **bytecodes** yang berisi perintah-perintah yang akan dijalankan selama fase eksekusi (*execute*). Bytecodes akan dieksekusi oleh **Java Virtual Machine (JVM)** yang merupakan bagian dari JDK (Java Development Kit) dan dasar dari *java platform.* Sebuah *Virtual Machine* (VM) merupakan sebuah perangkat lunak aplikasi yang mensimulasikan sebuah komputer tetapi menyembunyikan detail sistem operasi dan perangkat keras dari program yang berinteraksi dengan VM. Jika VM yang sama digunakan pada banyak komputer dengan platform yang berbeda, maka aplikasi tersebut dapat digunakan pada semua platform. Misal program Java yang telah kita kompilasi pada sistem operasi Windows, kita jalankan pada komputer dengan sistem operasi Linux dan menggunakan JVM, maka program tetap akan berjalan dengan baik.

Sebagai contoh, untuk mengeksekusi aplikasi Java yang telah kita kompilasi (Welcome.class) kita tinggal mengetikkan perintah pada terminal sistem (command prompt)

java Welcome

kemudian JVM akan mulai bekerja memulai fase 3.

**B.2.3 Fase *Loading* Program ke dalam Memori**

Dalam fase 3 ini, program akan ditempatkan dalam memori sebelum proses *loading* dimulai. *Class loader*  akan mengambil file .class yang berisi *bytecodes* dari program dan mengirimkan ke memori. Disamping file .class dari program kita, *class loader*  juga akan memanggil file .class dari Java yang digunakan oleh program kita. File .class tersebut dapat diambil dari sistem kita atau dari jaringan / internet.

**B.2.4 Fase *Bytecode Verification***

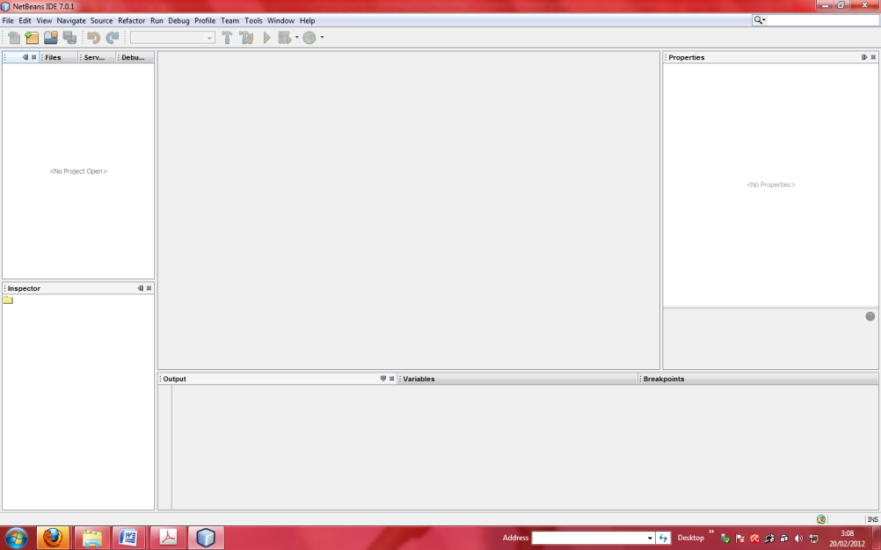
Dalam fase 4, setelah kelas di-*load,*  *bytecode verifier*  akan menguji *bytecodes* untuk meyakinkan bahwa *bytecodes* benar dan tidak melanggar batas-batas keamanan Java.

**B.2.5 Fase *Execution***

Dalam fase 5, JVM akan mengeksekusi *bytecodes* program, kemudian melakukan perintah seperti yang diberikan program. JVM akan melakukan proses translasi *bytecodes* ke dalam bahasa mesin yang bersesuaian dengan platform komputer yang digunakan untuk mengeksekusi.

**B.2. Pengenalan IDE NetBeans**

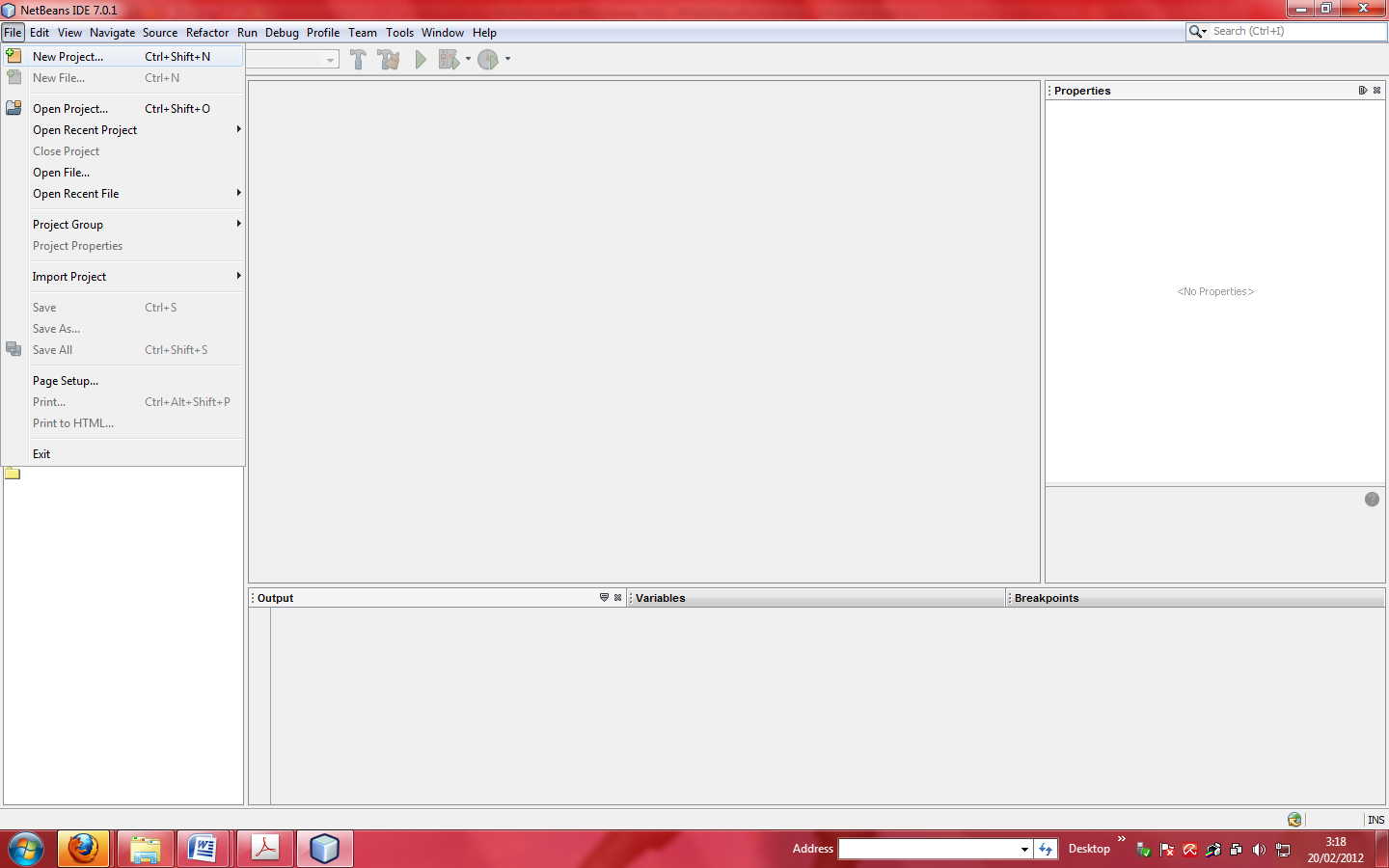
NetBeans merupakan sebuah IDE (integrated development environment) yang menyediakan fitur-fitur untuk memudahkan programmer dalam memprogram Java. NetBeans dapat didownload secara free di situs <http://www.netbeans.org>. Dengan NetBeans, maka proses penulisan source code, kompilasi, eksekusi, debugging (mencari kesalahan program) menjadi sangat mudah. Gambar 1.1 memperlihatkan tampilan awal dari NetBeans yang berisi berbagai layar.

****

**Gambar 1.1. Lingkungan NetBeans**

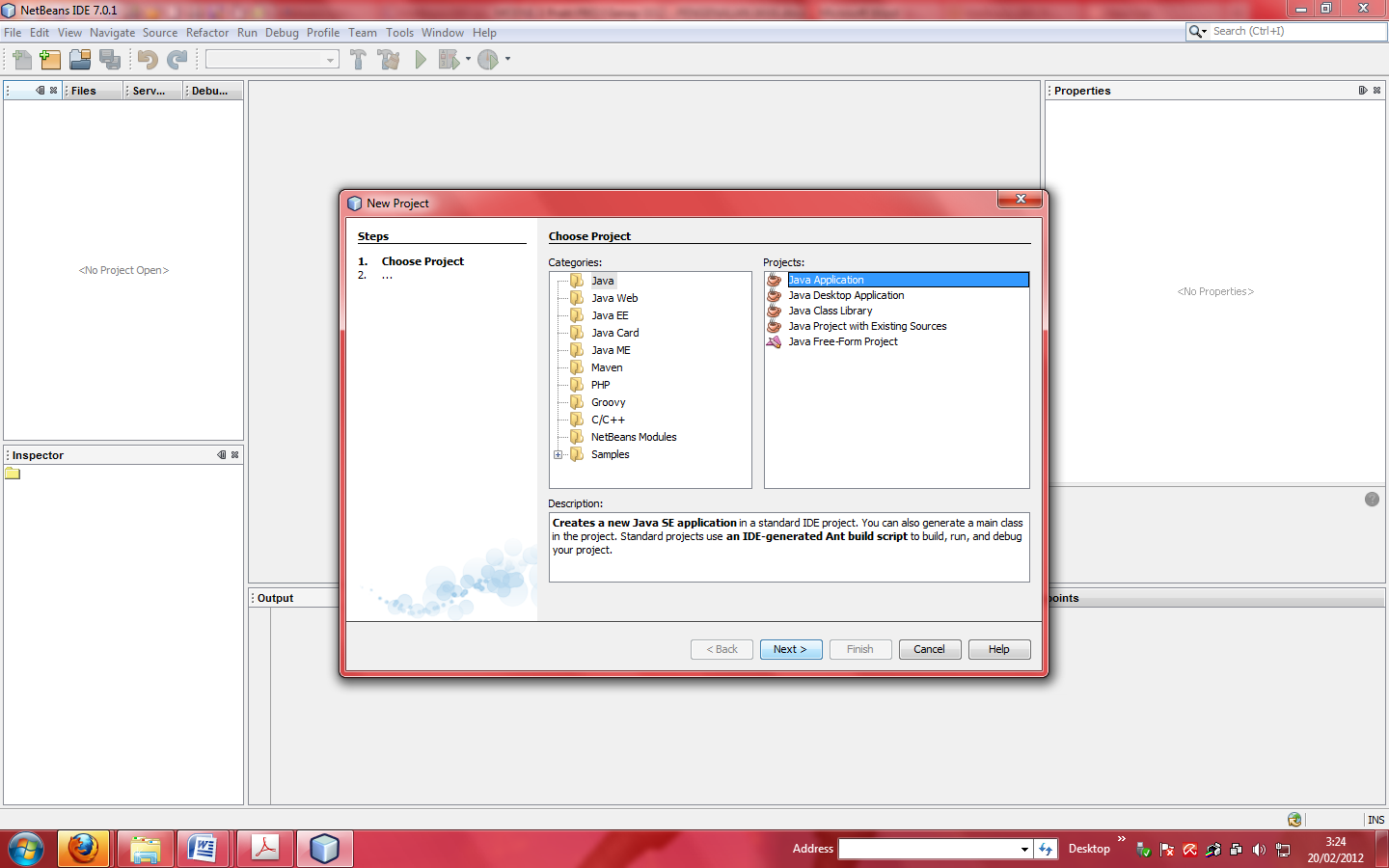
Dalam NetBeans, semua langkah pemrograman dilakukan dalam sebuah project. Sebuah project akan berisi source code dan semua lingkungan pemrograman yang diperlukan untuk mengkompilasi dan mengeksekusi source code. Sebuah project dapat berisi lebih dari satu source code program. Berikut ini adalah langkah-langkah untuk memulai project :

1. Buka NetBeans IDE melalui Start -> NetBeans -> NetBeans IDE
2. Setelah IDE NetBeans terbuka, pilihlah File -> New Project (Ctrl-Shift-N) seperti tampak pada gambar 1.2.

**

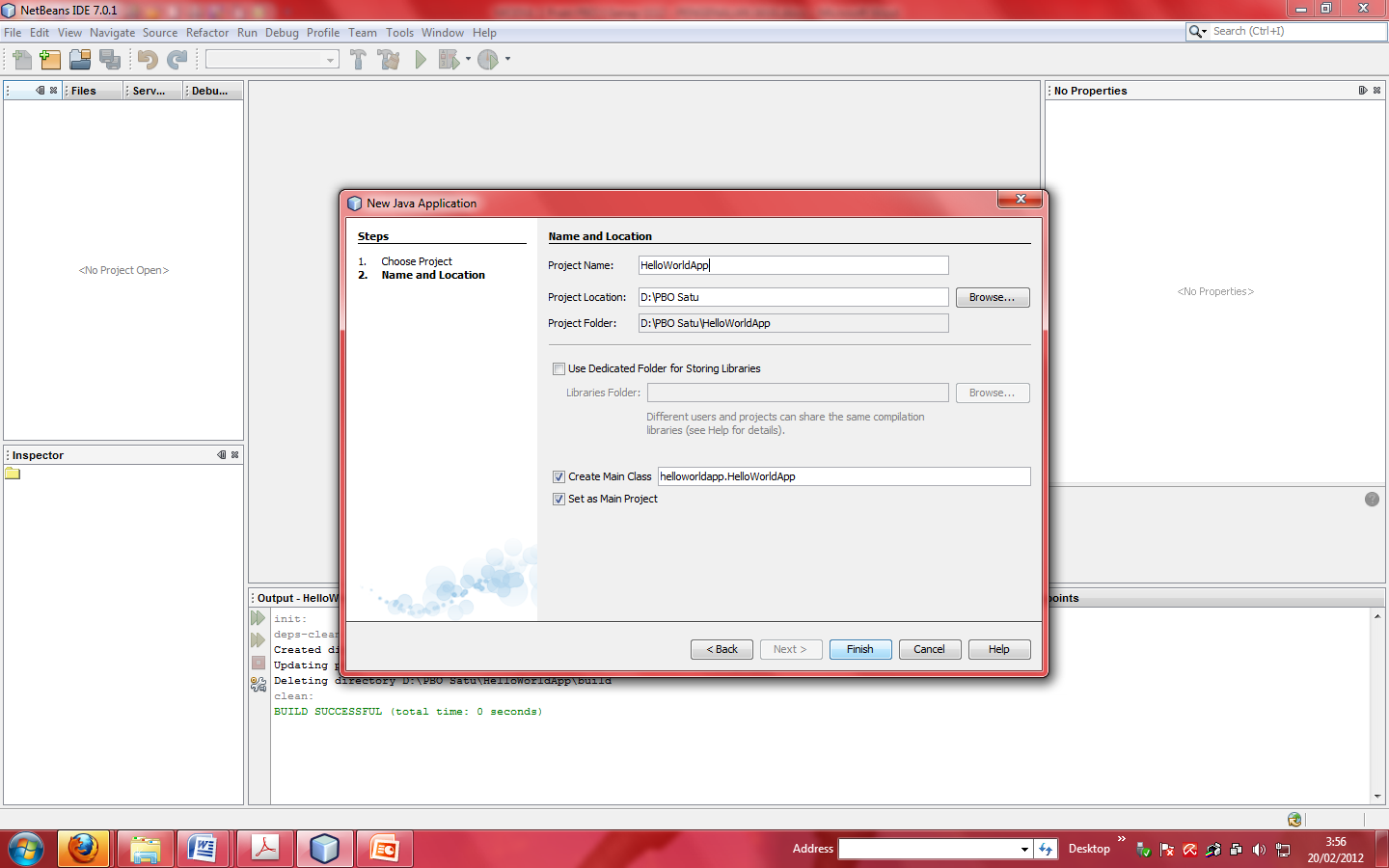
**Gambar 1.2. Memilih menu Project Baru**

1. Berikutnya, pada jendela New Project, pilihlah Categories Java -> Projects Java Application, seperti tampak pada gambar 1.3. Setelah itu klik Next.

**

**Gambar 1.3. Membuat Java Application Baru**

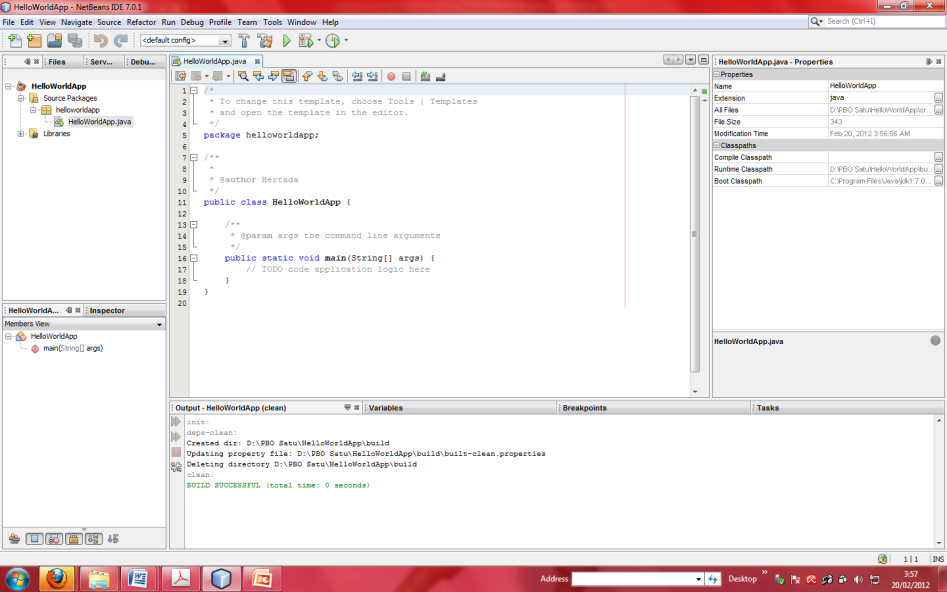
1. Pada jendela New Java Application, pilihlah
   1. Project Name : HelloWorldApp.
   2. Project Location : (nama folder tempat anda akan menyimpan project).
   3. Pilihlah Create Main Class seperti tampak pada gambar 1.4
   4. Pilihlah Set as Main Project.
   5. Klik Finish.

**

**Gambar 1.4. Membuat Java Application Baru**

Akan terbentuk sebuah project pada IDE. Komponen-komponen pada layar IDE seperti tampak pada gambar 1.5. adalah sebagai berikut :

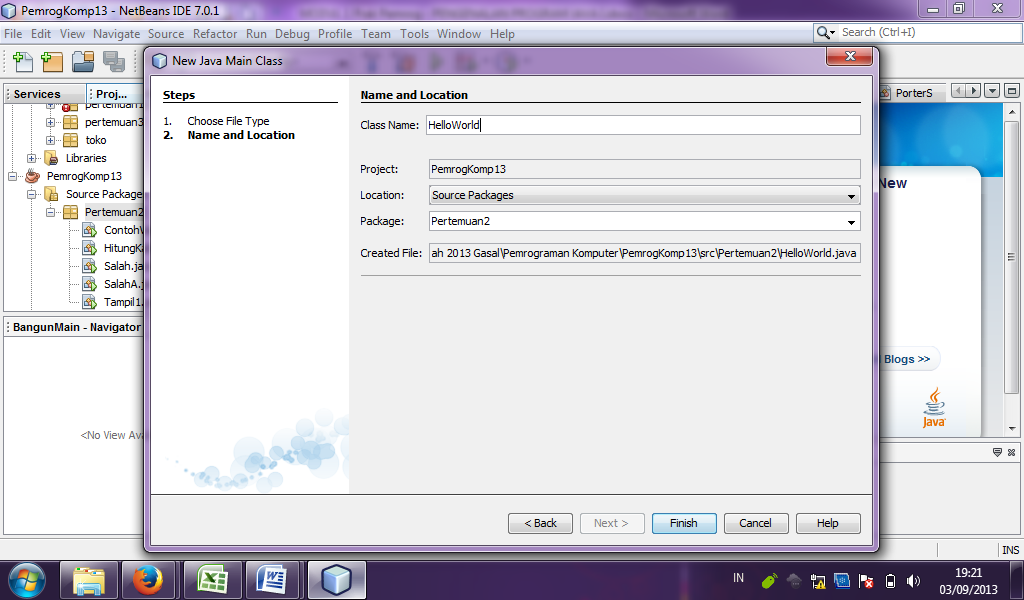
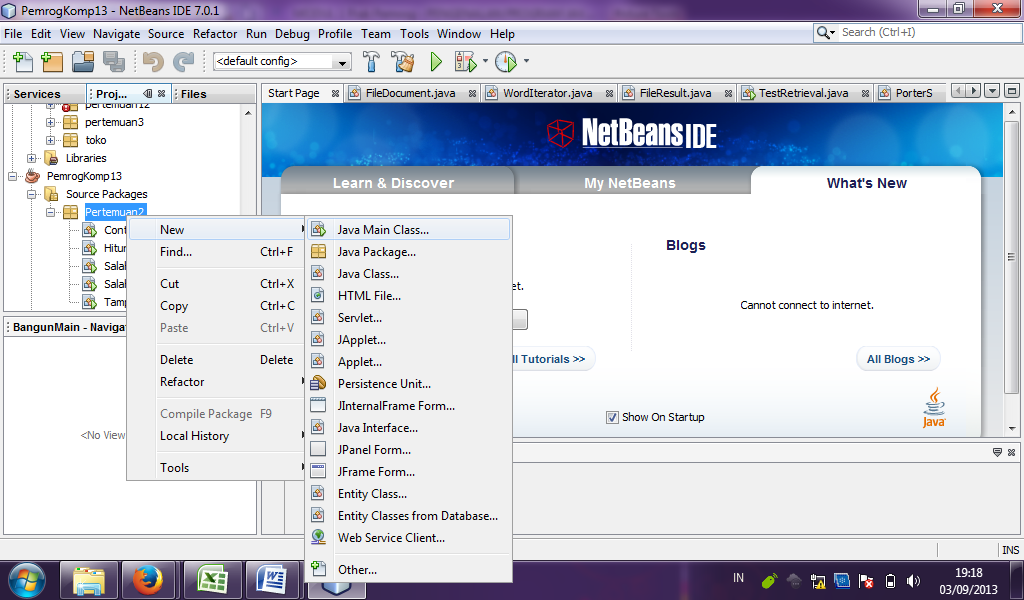
* Layar Project, berisi tampilan bentuk pohon yang menggambarkan komponen dari sebuah project berupa file berisi source code, banyak library yang digunakan oleh kode program kita.
* Layar Editor, berisi kode program dari program kelas HelloWorldApp kita.
* Layar Navigator, yang digunakan untuk mengarahkan secara cepat setiap elemen pada kelas yang kita pilih.
* Layar Output, yang berisi keluaran program yang dijalankan, atau keterangan program yang dicetak.
* Layar Properties, berisi atribut/data, method dan keterangan dari kelas yang kita buat.



**Gambar 1.5. Membuat Java Application Baru**

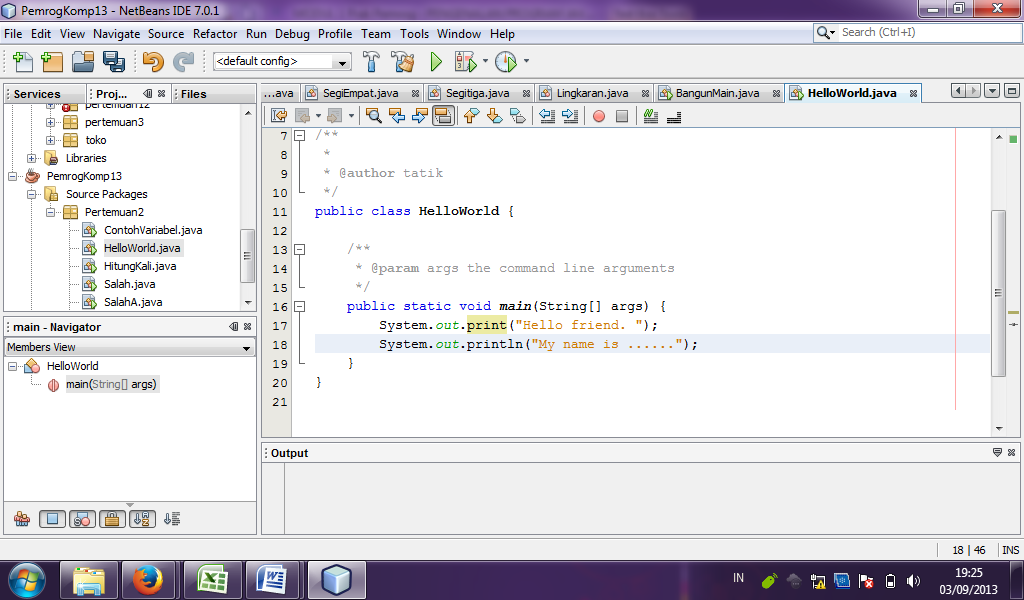
Setelah project terbentuk, tampak pada jendela project sebuah kelas **main** yang bernama HelloWorldApp. Kelas main merupakan sebuah kelas yang berisi method main (public static void main(String[] args). Jika sebuah kelas di-run, maka eksekusi akan dimulai dari baris pertama method main. Jika sebuah kelas tidak memiliki method main, maka kelas tersebut tidak dapat di-run. Method-method dalam kelas yang tidak memiliki method main akan dijalankan dari method main atau dari method lain.

Untuk lebih jelasnya, kita buat kelas baru bernama HelloWorldApp.java (gambar 1.6), dan kita isikan ke dalam kelas tersebut perintah-perintah seperti pada gambar 1.7



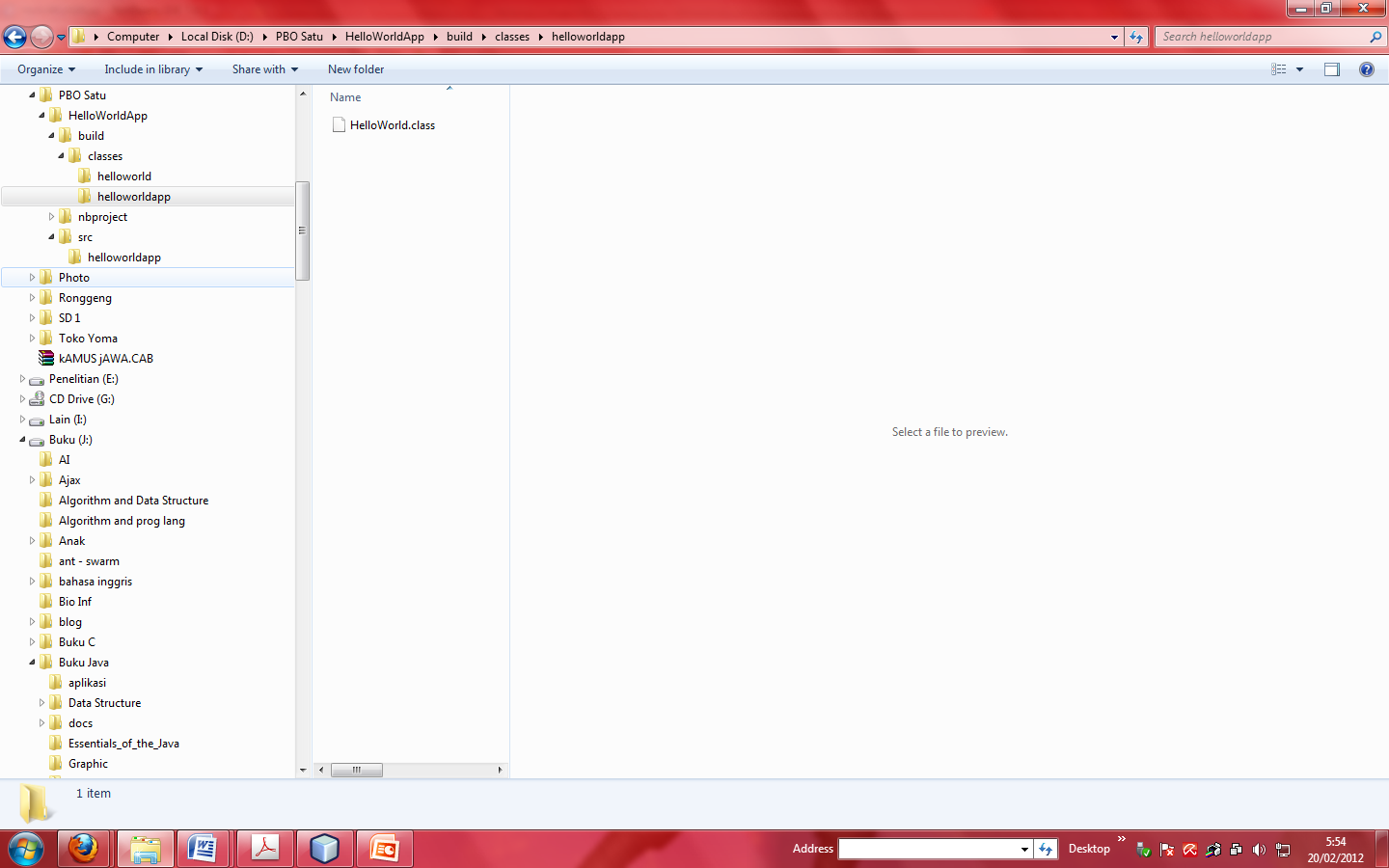
Klik kanan mouse diatas package

**Gambar 1.6. Membuat kelas baru.**



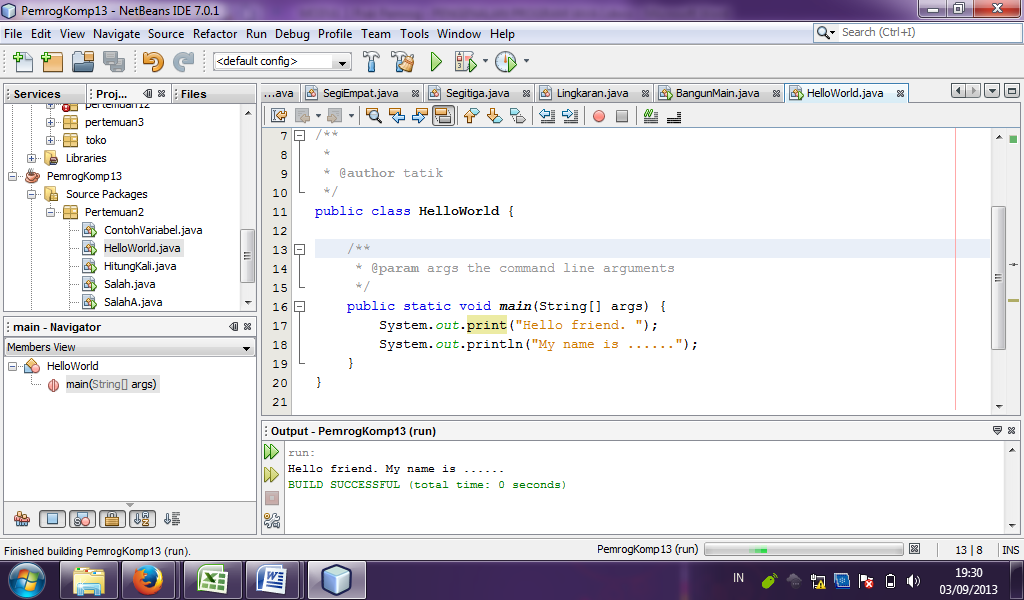
**Gambar 1.7. Membuat kelas HelloWorld.java.**

Simpanlah file HelloWorld.java dengan memilih File->Save atau Ctrl-S. Dan kini kita sudah memiliki sebuah kelas bernama HelloWorld.java. Proses Save pada NetBeans diset otomatis untuk melakukan proses Compile terhadap file yang kita simpan, sehingga kita tidak perlu melakukan fase compile secara terpisah. Tampak pada gambar 1.8, isi dari folder project kita setelah proses Save file HelloWorld.



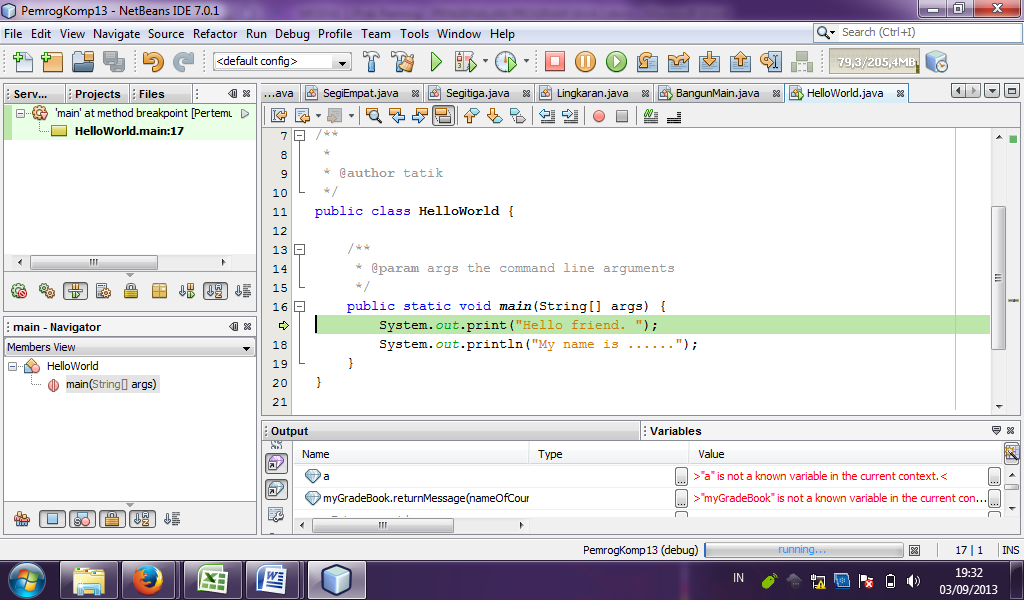
**Gambar 1.8. Isi folder project.**

Simpan HelloWorld.java dan kemudian kita lakukan proses run program dengan cara pilih Run->Run Main Project (F6). Karena pada project hanya terdapat satu kelas main, yaitu HelloWorld.java, maka otomatis method main yang dijalankan adalah method main pada kelas HelloWorld.java. Sehingga hasil dari run program tersebut adalah tampak pada gambar 1.9.



**Gambar 1.9. Output program HelloWorldApp.java.**

Untuk mengetahui jalannya program, kita dapat melakukan proses debug program dengan menggunakan kunci F7 dan F8. Posisi perintah dimana program sedang dijalankan, tampak dari arsiran pada layar, seperti tampak pada cuplikan debug gambar 1.10.



**Gambar 1.10. Cuplikan debug program HelloWorldApp.java.**

# C. LANGKAH PRAKTIKUM PERTEMUAN 1

1. Kerjakan contoh program HelloWorld diatas.
2. Buat project bernama pemrogramanSatu
3. Buat package baru dengan cara klik kanan diatas project pemrogramanSatu, kemudian tuliskan nama package : pertemuanSatu.
4. Buat kelas Tampil1 di dalam package pertemuanSatu. Tuliskan perintah berikut ini dalam kelas Tampil1 :

public class Tampil1 {

public static void main(String[] args) {

System.out.println("Pertemuan 1");

System.out.println("Kuliah Pemrograman Komputer");

}

}

* Simpan file kelas dengan urutan perintah : klik menu File -> klik submenu Save atau dengan perintah ctrl+s.
* Compile dan run program dengan urutan perintah : klik menu Run -> klik submenu Run File atau dengan perintah shift+F6. Kemudian lihat hasilnya pada layar output.

Untuk nomor 5 - 9, kerjakan masing-masing dalam satu file.

1. Program kali

public class HitungKali1 {

public static void main(String[] args) {

int a,b;

int hasil;

a=3;

b=4;

hasil=a\*b;

}

}

1. Program kali dan menampilkan hasil :

public class HitungKali2 {

public static void main(String[] args) {

int a,b;

int hasil;

a=3;

b=4;

hasil=a\*b;

System.out.println(hasil);

}

}

1. Program kali dan menampilkan hasil dengan layout :

public class HitungKali3 {

public static void main(String[] args) {

int a,b;

int hasil;

a=3;

b=4;

hasil=a\*b;

System.out.println("Hasil dari perkalian 3 dan 4 adalah :"+hasil);

}

}

1. Program kali dan menampilkan hasil dengan layout :

public class HitungKali4 {

public static void main(String[] args) {

int a,b;

int hasil;

a=3;

b=4;

hasil=a\*b;

System.out.println("Hasil dari perkalian "+a+" dan "+b+" adalah :"+hasil);

}

}

1. Program hitung kali dengan perintah print :

public class HitungKali5 {

public static void main(String[] args) {

int a,b;

int hasil;

a=3;

b=4;

hasil=a\*b;

System.out.print("Nilai dari a="+a);

System.out.println(" dan nilai dari b="+b);

System.out.println("Hasil dari perkalian "+a+" dan "+b+" adalah :"+hasil);

}

}

Untuk nomor 10-12, temukan kesalahan dalam program atau kejanggalan dalam program :

1. Program salahA :

public class SalahA {

public static void main(String[] args) {

float a;

a='d';

System.out.println(a);

}

}

1. Program salahB :

public class SalahB {

public static void main(String[] args)

int x,y,b;

a=3

x=4;

y=ax+b;

System.out.println(“Hasil dari persamaan y adalah :”+y);

}

}

1. Program salahC :

public class SalahC {

int hasilPangkat, nilai;

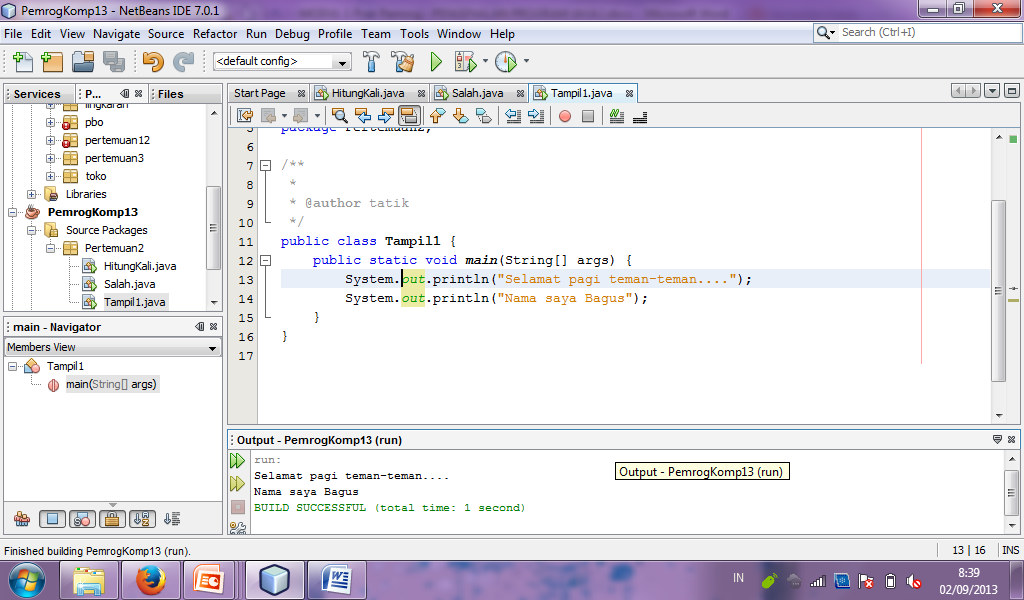
nilai=3;

hasilpangkat=nilai \* nilai ;

System.out.print(“Hasil dari pangkat adalah :”+hasilPangkat);

}

1. Buatlah secara mandiri program baru dalam kelas Tampil2, yang ketika di run akan menampilkan : (nama silahkan diganti dengan nama masing-masing)



1. **DAFTAR PUSTAKA**

Deitel, H.M. and P.J. Deitel, 2004. *JavaTM**How to Program.* 6th edition. Prentice Hall.

|  |
| --- |
|  |
|  |

☺ Selamat Berlatih ☺