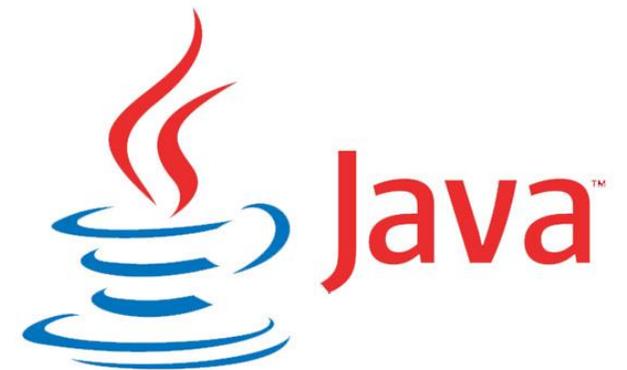


Pengantar Pemrograman dengan Bahasa Java

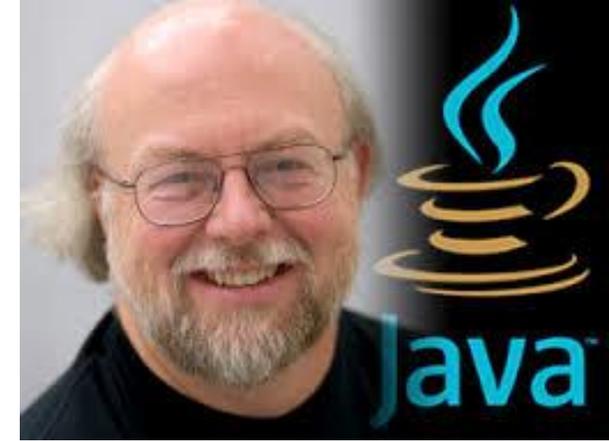
IF2123 Aljabar Geometri

Oleh: Rinaldi Munir



Sekolah Teknik Elektro dan Informatika
ITB

Sejarah Bahasa Java



James Gosling

- Bahasa java dibuat oleh James Gosling saat masih bergabung di *Sun Microsystems* dan dirilis tahun 1995.
- Bahasa Java dapat dijalankan pada berbagai komputer dan *platform* sistem operasi.
- Slogan Java: *Write once, run anywhere!* (Tulis sekali, jalankan di manapun)
- Java adalah bahasa pemrograman bersifat umum (*general purpose*)
- Sintaks Bahasa Java diadopsi dari Bahasa C dan C++ tetapi lebih sederhana
- Nama “java” diambil dari jenis kopi yang diminum oleh James Gosling saat itu.

- Java termasuk Bahasa pemrograman yang populer untuk mengembangkan aplikasi, termasuk aplikasi berbasis web.

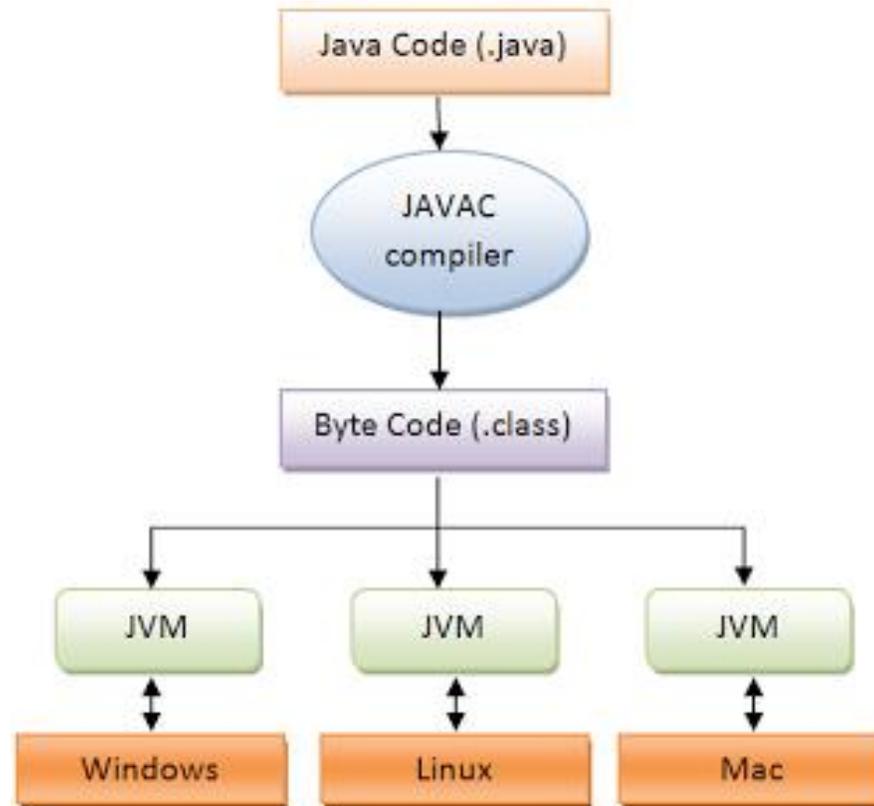
Language Rank	Types	Spectrum Ranking	Spectrum Ranking
1. Java	🌐 📱 🖥️	100.0	100.0
2. C	📱 🖥️ 🖨️	99.9	99.3
3. C++	📱 🖥️ 🖨️	99.4	95.5
4. Python	🌐 🖥️	96.5	93.5
5. C#	🌐 📱 🖥️	91.3	92.4
6. R	🖥️	84.8	84.8
7. PHP	🌐	84.5	84.5
8. JavaScript	🌐 📱	83.0	78.9
9. Ruby	🌐 🖥️	76.2	74.3
10. Matlab	🖥️	72.4	72.8

Gambar 1. Sepuluh (10) bahasa pemrograman top 2015:

Teknologi Java = Bahasa pemrograman + platform

Java Sebagai Bahasa Pemrograman

- Bahasa java memiliki karakteristik: *sederhana, berorientasi objek, interpreted, terdistribusi, tangguh, portable, memiliki kinerja tinggi, aman, dinamis*. (Baca di: <http://java.sun.com/docs/white/langenv/>)
- *Compiler* java mengubah kode program menjadi bahasa *intermediate* yang disebut *java bytecode*. Kemudian *interpreter* Java bernama JVM (*Java Virtual Machine*) melakukan interpretasi *bytecode* setiap *bytecode* tersebut dijalankan.



Gambar 2. Proses kompilasi dan interpretasi program Java
(Sumber gambar: <http://belajarjava-19.blogspot.co.id>)

Java Sebagai Sebuah *Platform*

- *Platform* adalah lingkungan perangkat keras dan perangkat lunak untuk menjalankan program.
- Java adalah *platform* perangkat lunak untuk menjalankan program java.
- *Platform* java terdiri dari dua komponen:
 1. *Java Virtual Machine* (JVM)
 2. *Java Application Programming Interface* (Java API)
- JVM pada dasarnya adalah aplikasi sederhana yang ditulis dalam bahasa C untuk mengeksekusi program yang ditulis dalam bahasa Java.

- Cara kerja JVM: Pada saat eksekusi, JVM membaca *bytecode*, lalu mengubahnya ke bahasa mesin yang sesuai dengan komputer yang menjalankannya.
- Proses kompilasi bahasa java menghasilkan *bytecode* yang selalu sama untuk setiap sistem operasi atau jenis mesinnya, tetapi JVM akan mengubah *bytecode* menjadi bahasa mesin tujuannya.
- Java API merupakan *library* yang disediakan java untuk mengembangkan program java. Java API berisi sekumpulan komponen perangkat lunak yang memudahkan pemrogram java mengembangkan aplikasi.

Tiga edisi java yang dikeluarkan oleh *Sun Microsystem*:

1. *Java™ 2 Standard Edition (J2SE)*: edisi standard java yang dapat digunakan untuk aplikasi java secara umum.
2. *Java™ 2 Micro Edition (J2ME)*: edisi java untuk perangkat yang memiliki keterbatasan memori dan prosesor seperti perangkat *mobile* dan *wireless*.
3. *Java™ 2 Enterprise Edition (J2EE)*: edisi java untuk pengembangan aplikasi besar

Kakas Java

Untuk menulis program java, diperlukan beberapa kakas:

1. *Java™ 2 Standard Edition (J2SE)*

Unduh paket SDK (*Software Development Kit*) java terbaru dari situs resmi <http://java.sun.com>

2. Editor teks

Sembarang editor teks seperti *Notepad, Ultraedit, Wordpad, Vi,* atau *Joe*

File Edit View History Bookmarks Tools Help

Java - Wikipedia bahasa In... x Java SE Development Kit 8 ... x +

www.oracle.com/technetwork/java/javase/downloads/jdk Search

Most Visited Facebook (79) Getting Started Stats (3) Facebook Latest Headlines Facebook (3) Facebook Facebook Call for Paper

Java Magazine

JDK 8u60 Checksum

Java SE Development Kit 8u60

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	77.69 MB	jdk-8u60-linux-arm32-vfp-hflt.tar.gz
Linux ARM v8 Hard Float ABI	74.64 MB	jdk-8u60-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.66 MB	jdk-8u60-linux-i586.rpm
Linux x86	174.83 MB	jdk-8u60-linux-i586.tar.gz
Linux x64	152.67 MB	jdk-8u60-linux-x64.rpm
Linux x64	172.84 MB	jdk-8u60-linux-x64.tar.gz
Mac OS X x64	227.07 MB	jdk-8u60-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.67 MB	jdk-8u60-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.02 MB	jdk-8u60-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.18 MB	jdk-8u60-solaris-x64.tar.Z
Solaris x64	96.71 MB	jdk-8u60-solaris-x64.tar.gz
Windows x86	180.82 MB	jdk-8u60-windows-i586.exe
Windows x64	186.16 MB	jdk-8u60-windows-x64.exe

Java SE Development Kit 8u60 Demos and Samples Downloads

You must accept the [Oracle BSD License](#). to download this software.

Accept License Agreement Decline License Agreement

Product / File Description File Size Download

Forums

Java Magazine

Java.net

Developer Training

Tutorials

Java.com



October 25 - 29, 2015
San Francisco, U.S.

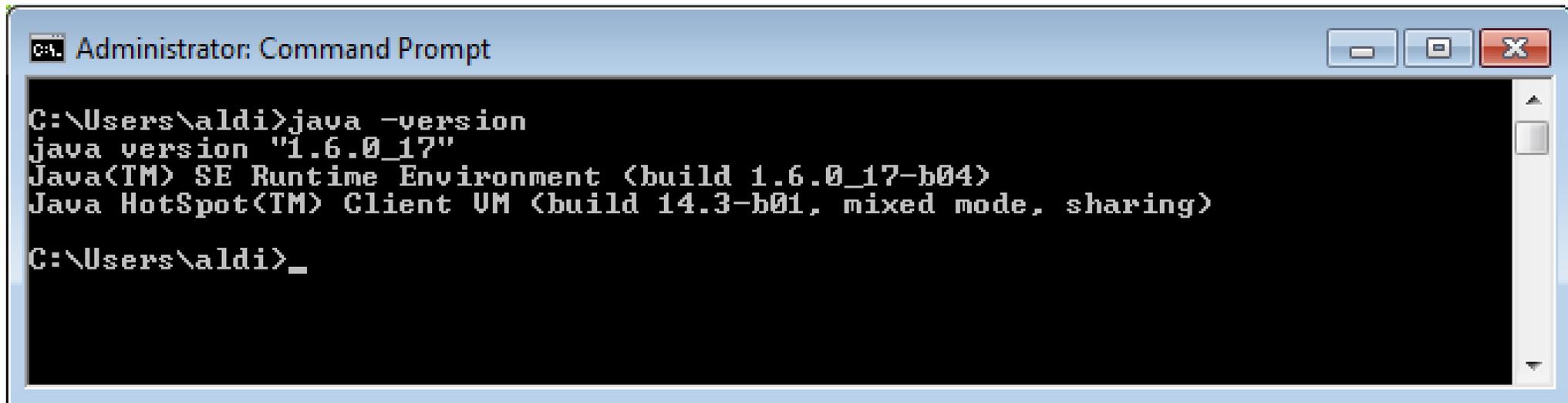
[Register Now](#)



Duke's Choice Award

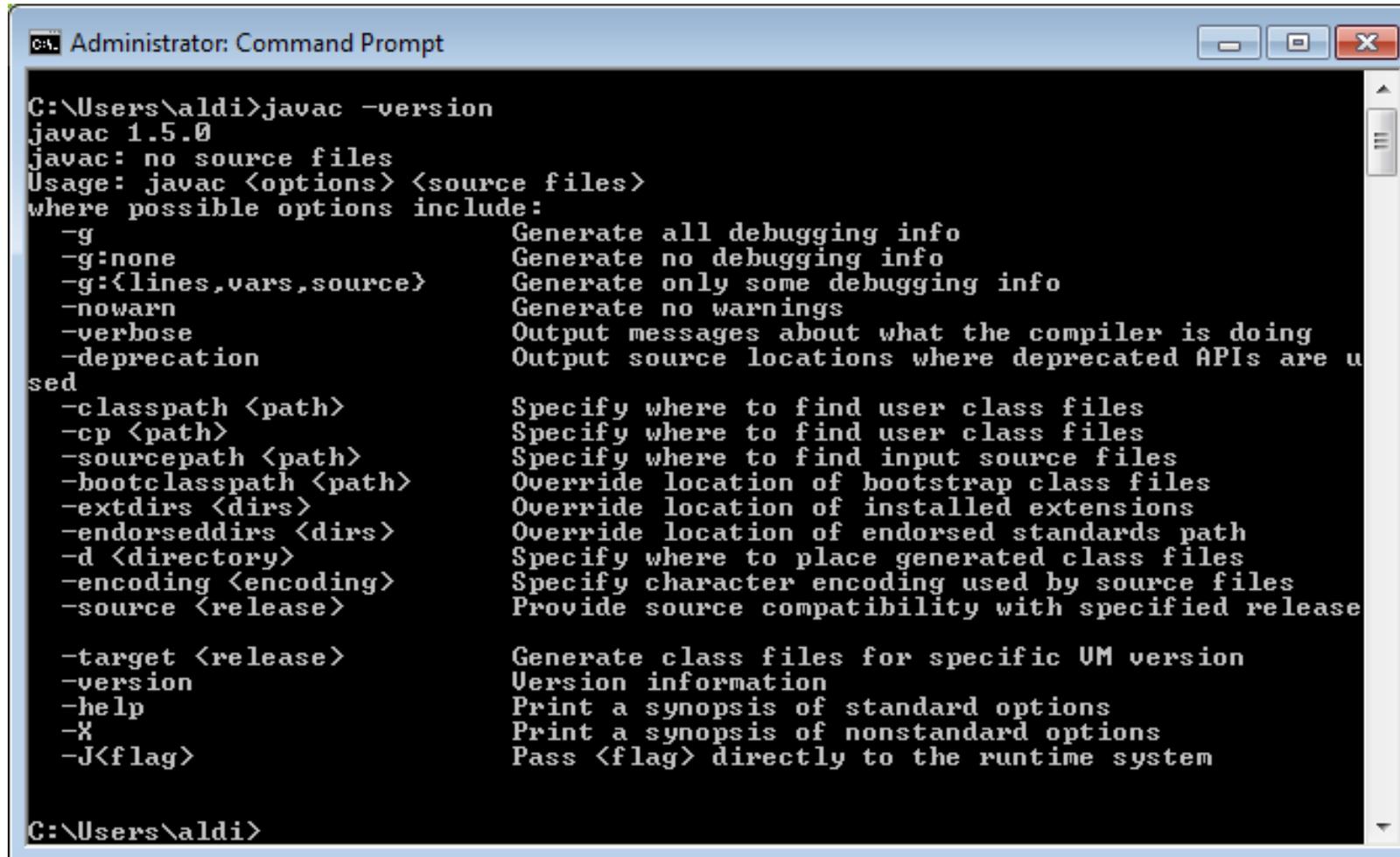
- Untuk pengembangan aplikasi visual (*visual programming*), anda membutuhkan kakas pengembangan java yang mengintegrasikan:
 - JDK
 - Editor teks
 - Editor antarmuka pengguna (GUI = *Graphical User Interface*)
 - Manajemen aplikasi
 - *Debugger*
- Contoh kakas pengembangan java: *Netbeans* dan *Eclipse*

- Instalasilah JDK ke komputer anda dan ikuti semua instruksi untuk menginstalasinya.
- Aturlah nilai *environment variable* PATH melalui Control Panel > System > Advanced > Environment Variables
- Untuk mengetahui versi JRE yang terinstal:



```
Administrator: Command Prompt
C:\Users\aldi>java -version
java version "1.6.0_17"
Java(TM) SE Runtime Environment (build 1.6.0_17-b04)
Java HotSpot(TM) Client VM (build 14.3-b01, mixed mode, sharing)
C:\Users\aldi>_
```

- Untuk mengetahui versi JDK yang terinstal:



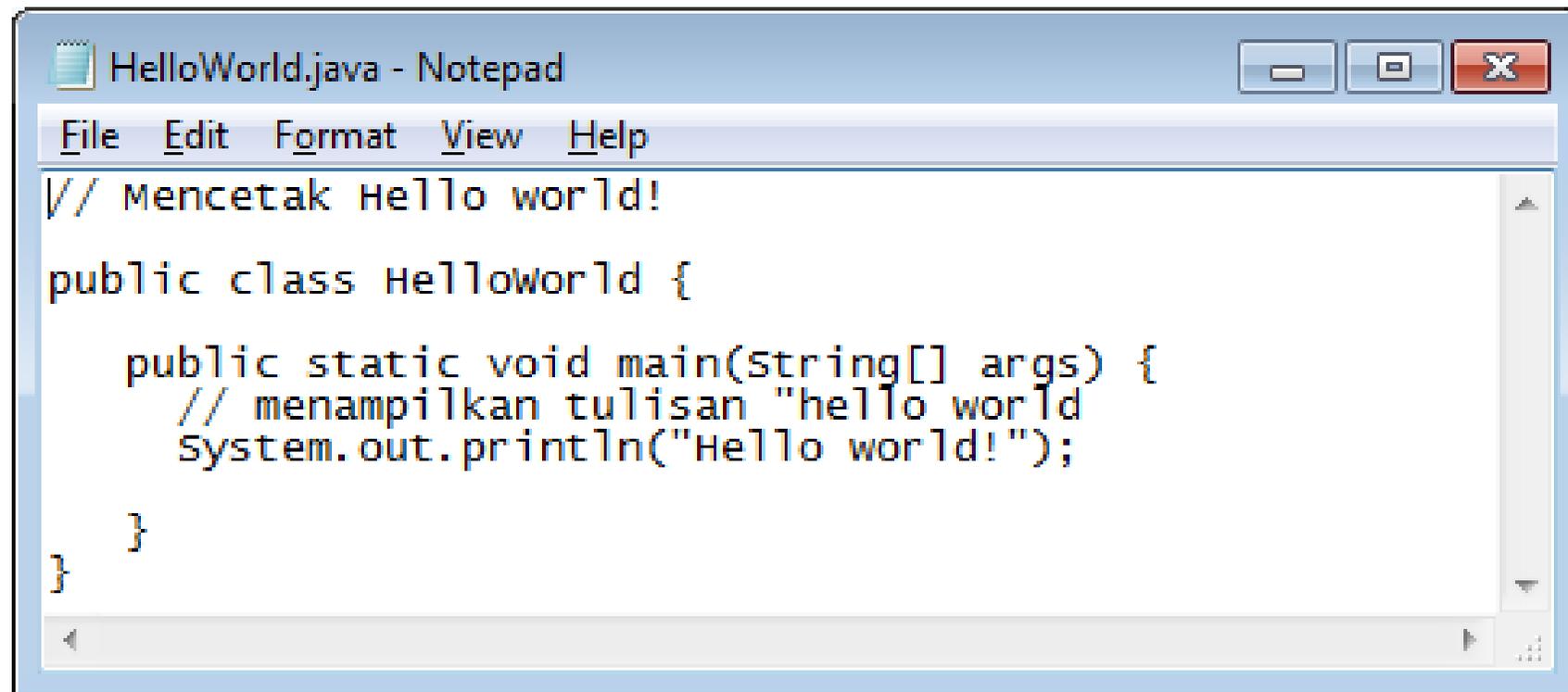
```
Administrator: Command Prompt
C:\Users\aldi>javac -version
javac 1.5.0
javac: no source files
Usage: javac <options> <source files>
where possible options include:
  -g                Generate all debugging info
  -g:none           Generate no debugging info
  -g:{lines,vars,source}  Generate only 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
  -cp <path>        Specify where to find user class files
  -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
  -d <directory>   Specify where to place generated class 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
  -X               Print a synopsis of nonstandard options
  -J<flag>         Pass <flag> directly to the runtime system

C:\Users\aldi>
```

Program javaku yang pertama

- Ketik program HelloWorld di bawah ini dengan editor teks, simpan dengan nama file HelloWorld.java (harus sama persis dengan nama class)

A screenshot of a Notepad window titled "HelloWorld.java - Notepad". The window has a menu bar with "File", "Edit", "Format", "View", and "Help". The text area contains the following Java code:

```
// Mencetak Hello world!  
  
public class HelloWorld {  
  
    public static void main(String[] args) {  
        // menampilkan tulisan "hello world"  
        System.out.println("Hello world!");  
    }  
}
```

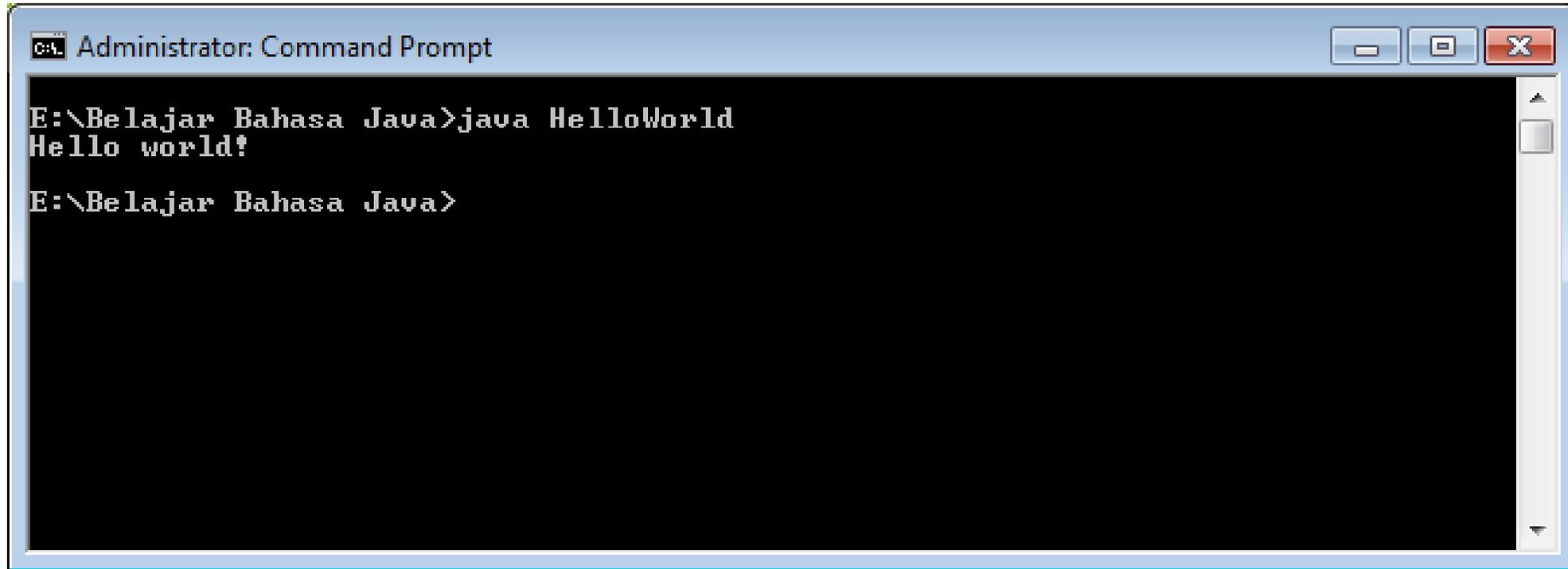
- Kompilasi program *HelloWorld* dari *command prompt*:



```
Administrator: Command Prompt
E:\Belajar Bahasa Java>javac HelloWorld.java
E:\Belajar Bahasa Java>
```

- Hasilnya sebuah arsip bernama `HelloWord.class`

- Jalankan arsip `HelloWorld.class` melalui *command prompt*:

A screenshot of a Windows Command Prompt window titled "Administrator: Command Prompt". The window has a blue title bar with standard minimize, maximize, and close buttons. The main area is black with white text. The text shows the current directory as "E:\Belajar Bahasa Java", the command "java HelloWorld" being entered, and the output "Hello world!". The prompt returns to "E:\Belajar Bahasa Java>".

```
Administrator: Command Prompt
E:\Belajar Bahasa Java>java HelloWorld
Hello world!
E:\Belajar Bahasa Java>
```

- Horeeee..., saya sudah bisa membuat program java!

Class

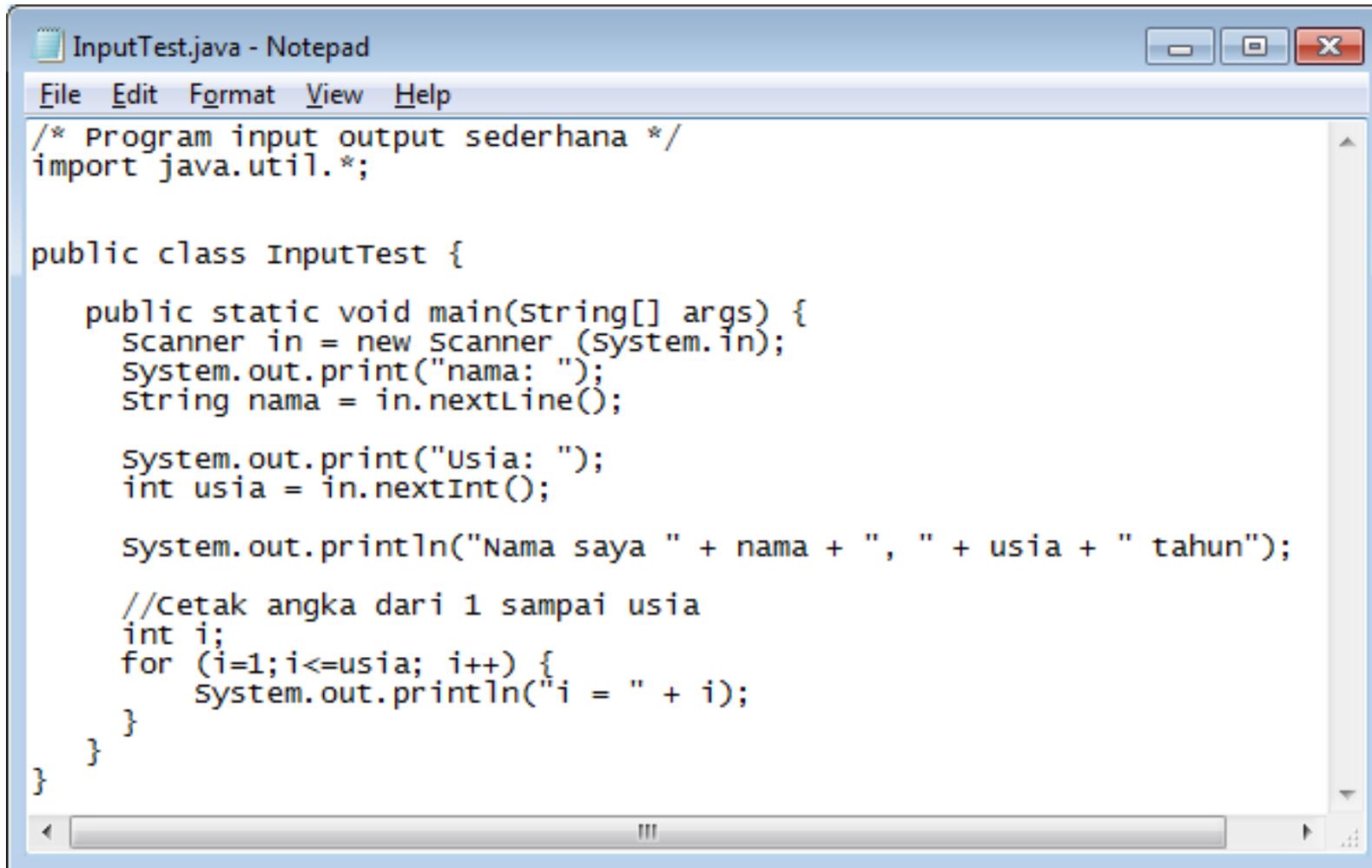
- Bahasa java adalah berorientasi objek. Struktur bahasa java terdiri dari kelas-kelas objek.
- Kelas adalah *blue-print* dari objek, sedangkan objek adalah instans dari kelas pada saat ruuning.
- Setiap kelas di dalam java memiliki *template*:

```
class NamaKelas {  
    // body kelas ditulis di sini  
}
```

- Di dalam kelas terdapat *atribut* dan *method* . Salah satu atau keduanya mungkin tidak terdapat di dalam kelas.
- Atribut di dalam kelas dinyatakan dengan variabel atau objek kelas lain.
- *Method* adalah operasi (prosedur, fungsi, atau konstruktor) yang dimiliki oleh sebuah kelas.
- Kelas `HelloWorld` hanya mempunyai satu *method*, yaitu `main`:

```
public static void main(String[] args) {  
    // menampilkan tulisan "Hello world!"  
    System.out.println("Hello world!");  
}
```

Program Input/Output Sederhana

A screenshot of a Notepad window titled "InputTest.java - Notepad". The window contains a Java program that uses the Scanner class to read user input for a name and age, and then prints the name and age, followed by a loop that prints numbers from 1 to the entered age.

```
InputTest.java - Notepad
File Edit Format View Help
/* Program input output sederhana */
import java.util.*;

public class InputTest {

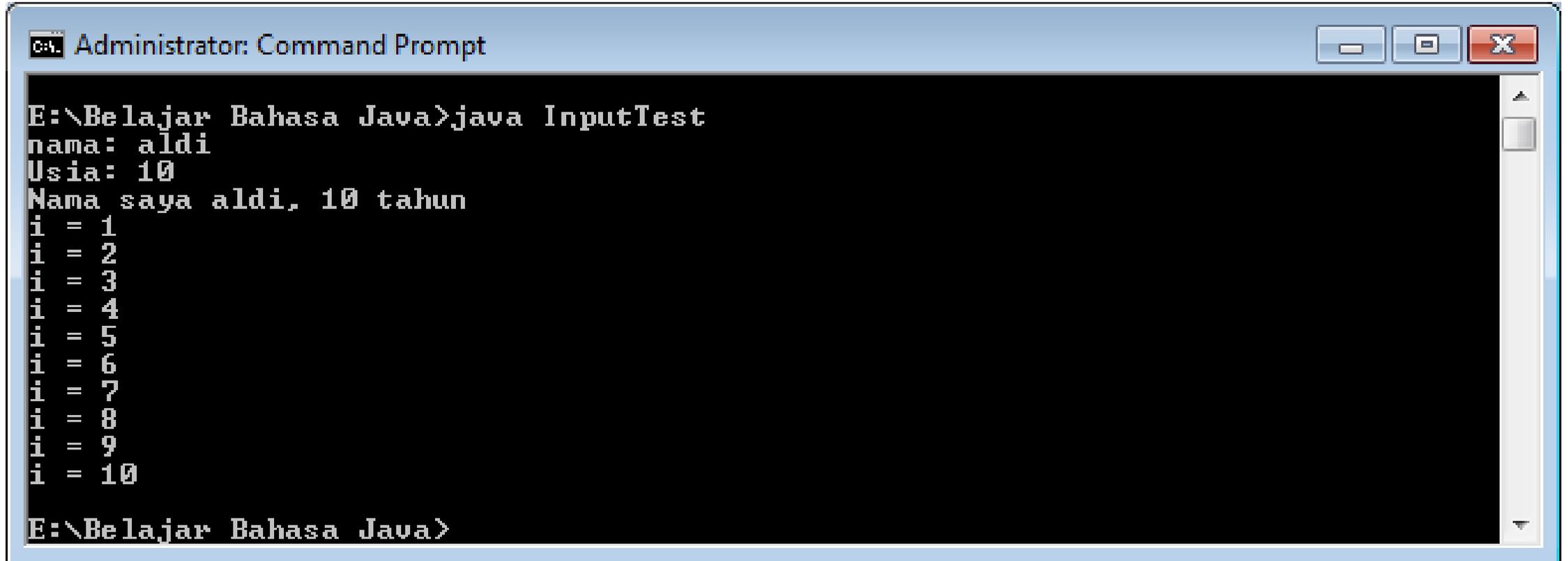
    public static void main(String[] args) {
        Scanner in = new Scanner (System.in);
        System.out.print("nama: ");
        String nama = in.nextLine();

        System.out.print("Usia: ");
        int usia = in.nextInt();

        System.out.println("Nama saya " + nama + ", " + usia + " tahun");

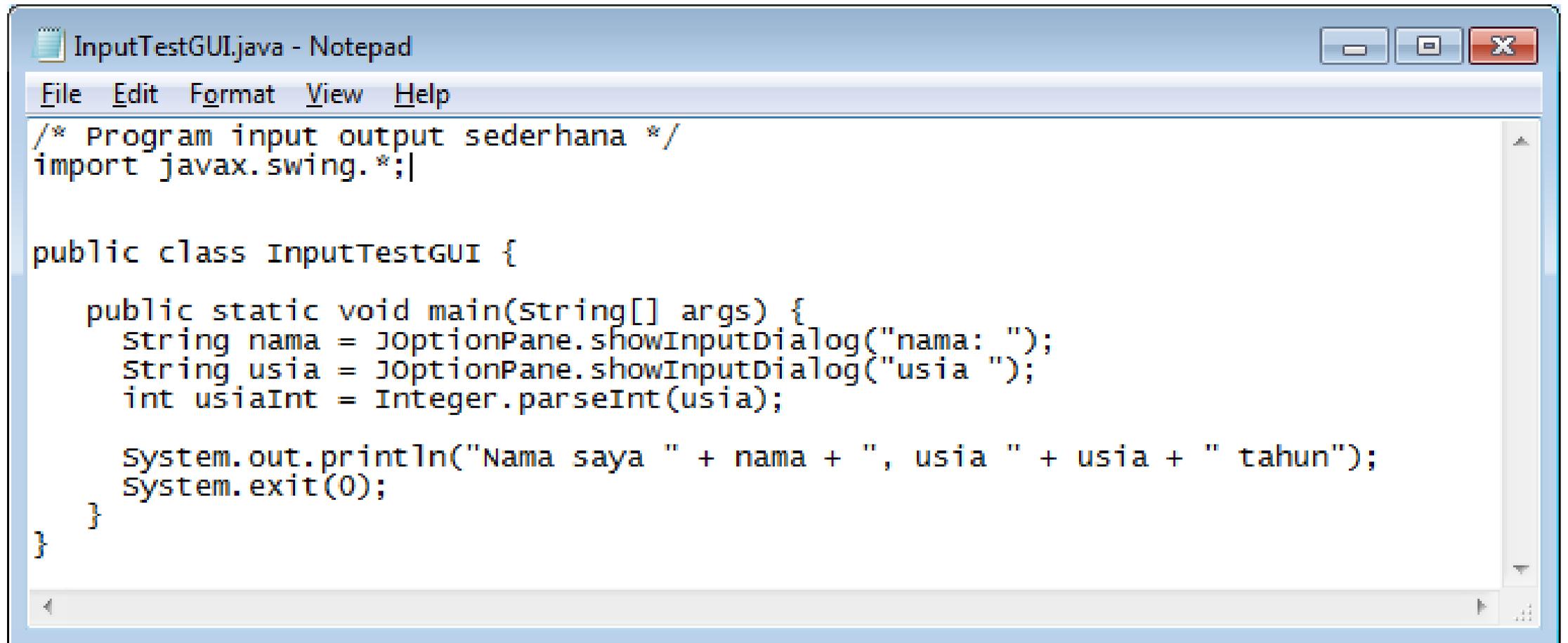
        //Cetak angka dari 1 sampai usia
        int i;
        for (i=1;i<=usia; i++) {
            System.out.println("i = " + i);
        }
    }
}
```

- Kompilasi `InputTest.java` dan jika sudah benar jalankan programnya:



```
Administrator: Command Prompt
E:\Belajar Bahasa Java>java InputTest
nama: aldi
Usia: 10
Nama saya aldi, 10 tahun
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
i = 10
E:\Belajar Bahasa Java>
```

Program Input dengan GUI



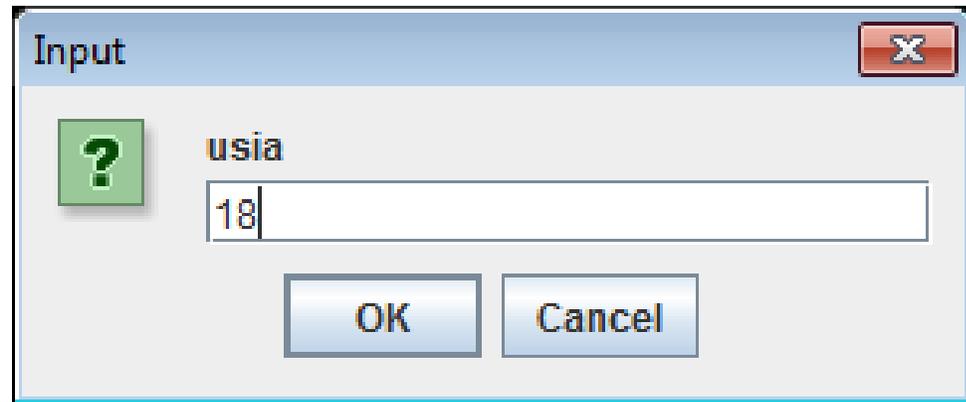
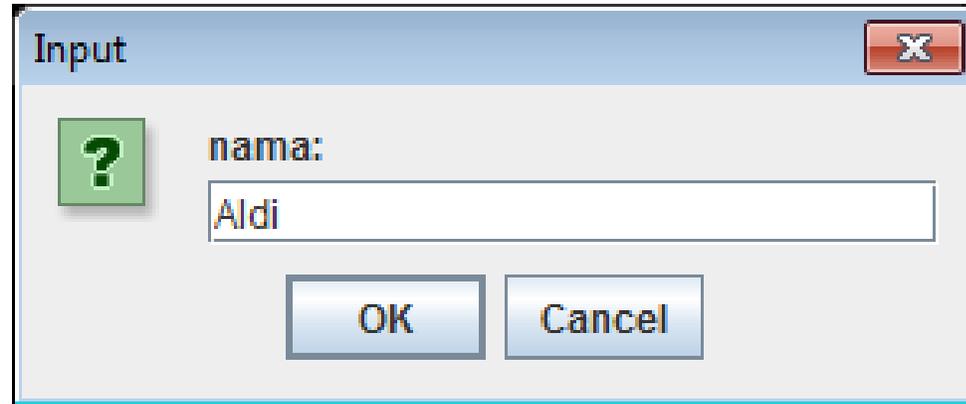
```
InputTestGUI.java - Notepad
File Edit Format View Help
/* Program input output sederhana */
import javax.swing.*;

public class InputTestGUI {

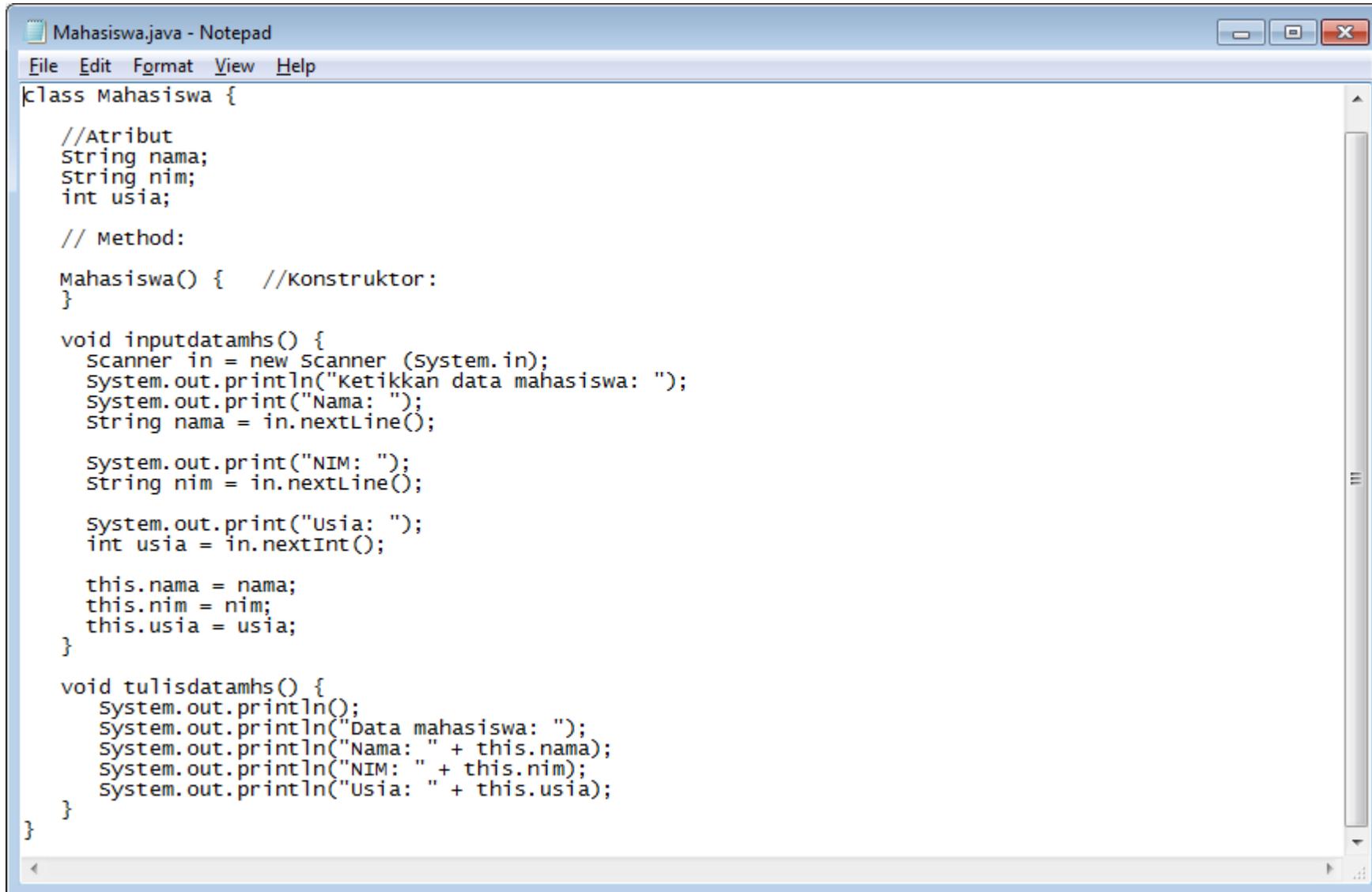
    public static void main(String[] args) {
        String nama = JOptionPane.showInputDialog("nama: ");
        String usia = JOptionPane.showInputDialog("usia ");
        int usiaInt = Integer.parseInt(usia);

        System.out.println("Nama saya " + nama + ", usia " + usia + " tahun");
        System.exit(0);
    }
}
```

- Kompilasi `InputTestGUI.java` dan jika sudah benar jalankan programnya:



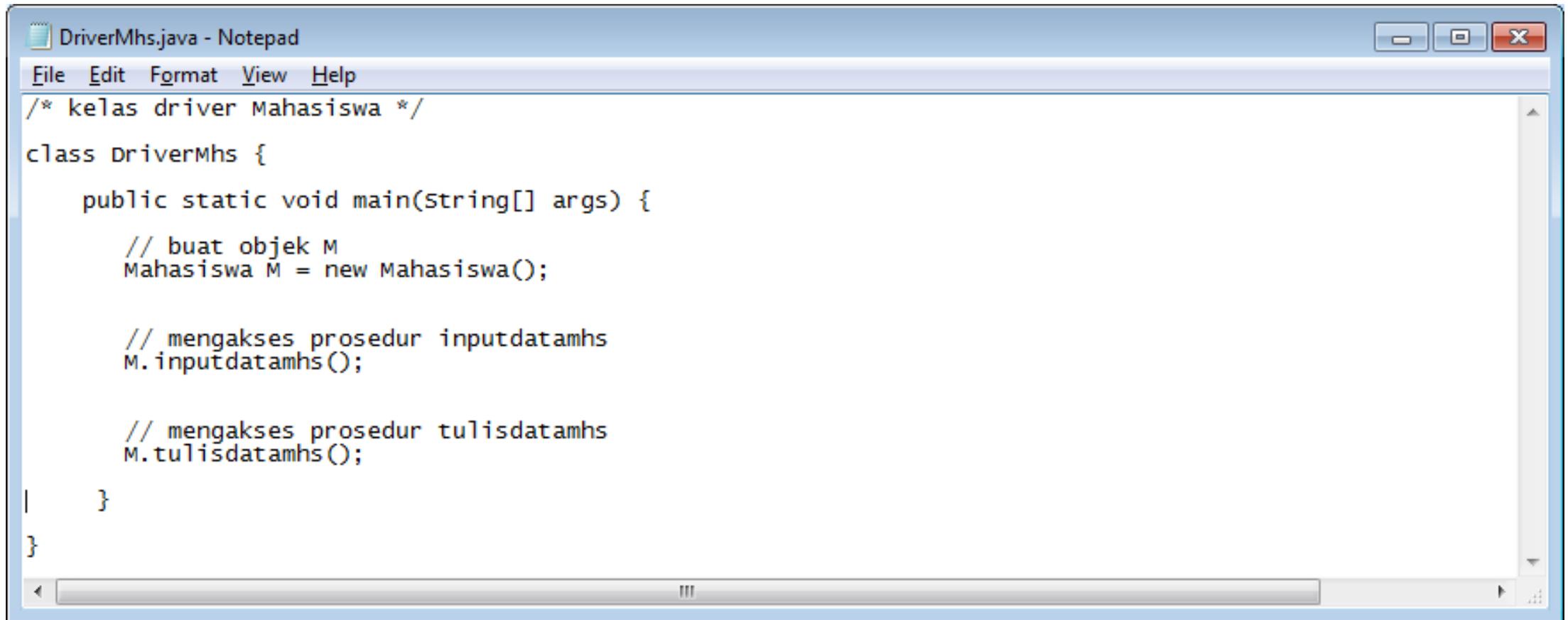
Kelas Mahasiswa



```
class Mahasiswa {  
  
    //Atribut  
    String nama;  
    String nim;  
    int usia;  
  
    // Method:  
  
    Mahasiswa() { //konstruktor:  
    }  
  
    void inputdatamhs() {  
        Scanner in = new Scanner (System.in);  
        System.out.println("Ketikkan data mahasiswa: ");  
        System.out.print("Nama: ");  
        String nama = in.nextLine();  
  
        System.out.print("NIM: ");  
        String nim = in.nextLine();  
  
        System.out.print("Usia: ");  
        int usia = in.nextInt();  
  
        this.nama = nama;  
        this.nim = nim;  
        this.usia = usia;  
    }  
  
    void tulisdatamhs() {  
        System.out.println();  
        System.out.println("Data mahasiswa: ");  
        System.out.println("Nama: " + this.nama);  
        System.out.println("NIM: " + this.nim);  
        System.out.println("Usia: " + this.usia);  
    }  
}
```

Kelas DriverMhs

(yang menggunakan kelas Mahasiswa)

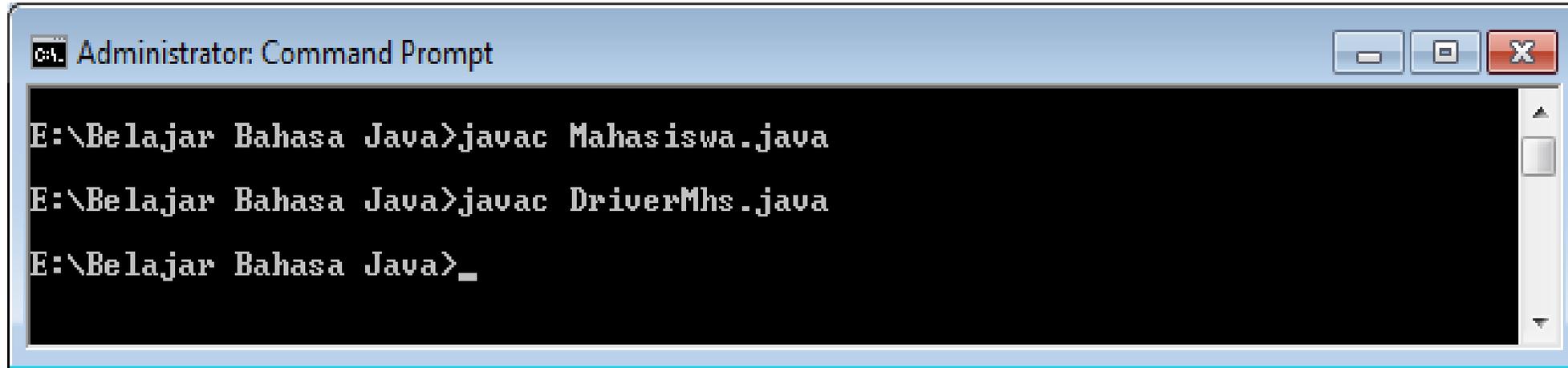


```
DriverMhs.java - Notepad
File Edit Format View Help
/* kelas driver mahasiswa */
class DriverMhs {
    public static void main(String[] args) {
        // buat objek M
        Mahasiswa M = new Mahasiswa();

        // mengakses prosedur inputdatamhs
        M.inputdatamhs();

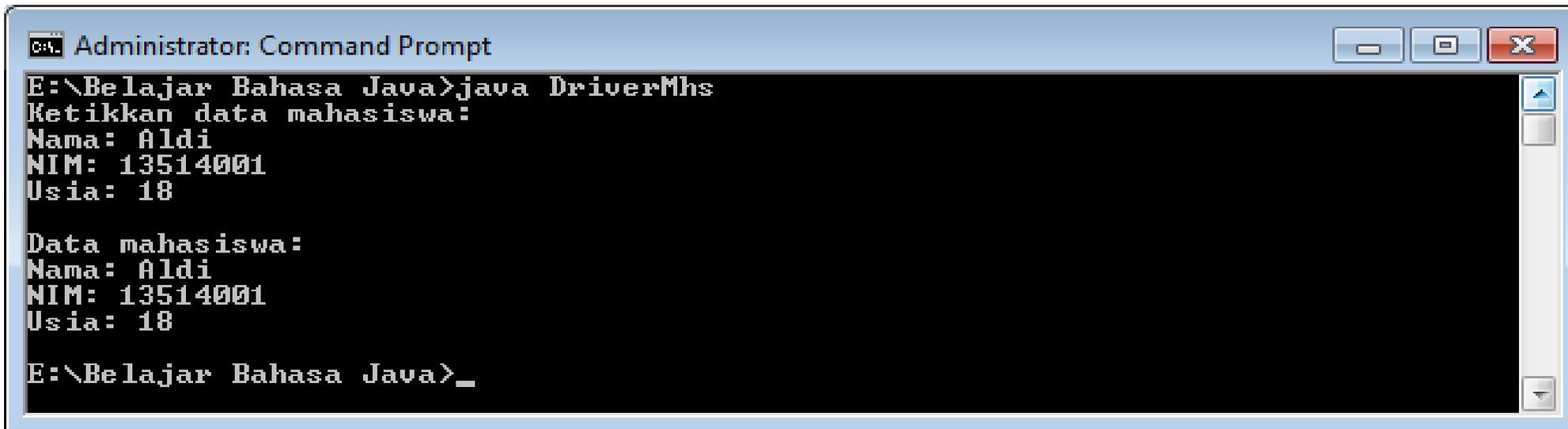
        // mengakses prosedur tulisdatamhs
        M.tulisdatamhs();
    }
}
```

- Kompilasi masing-masing Mahasiswa.java dan DriverMhs.java



```
Administrator: Command Prompt
E:\Belajar Bahasa Java>javac Mahasiswa.java
E:\Belajar Bahasa Java>javac DriverMhs.java
E:\Belajar Bahasa Java>_
```

- Jalankan kelas DriverMhs.class

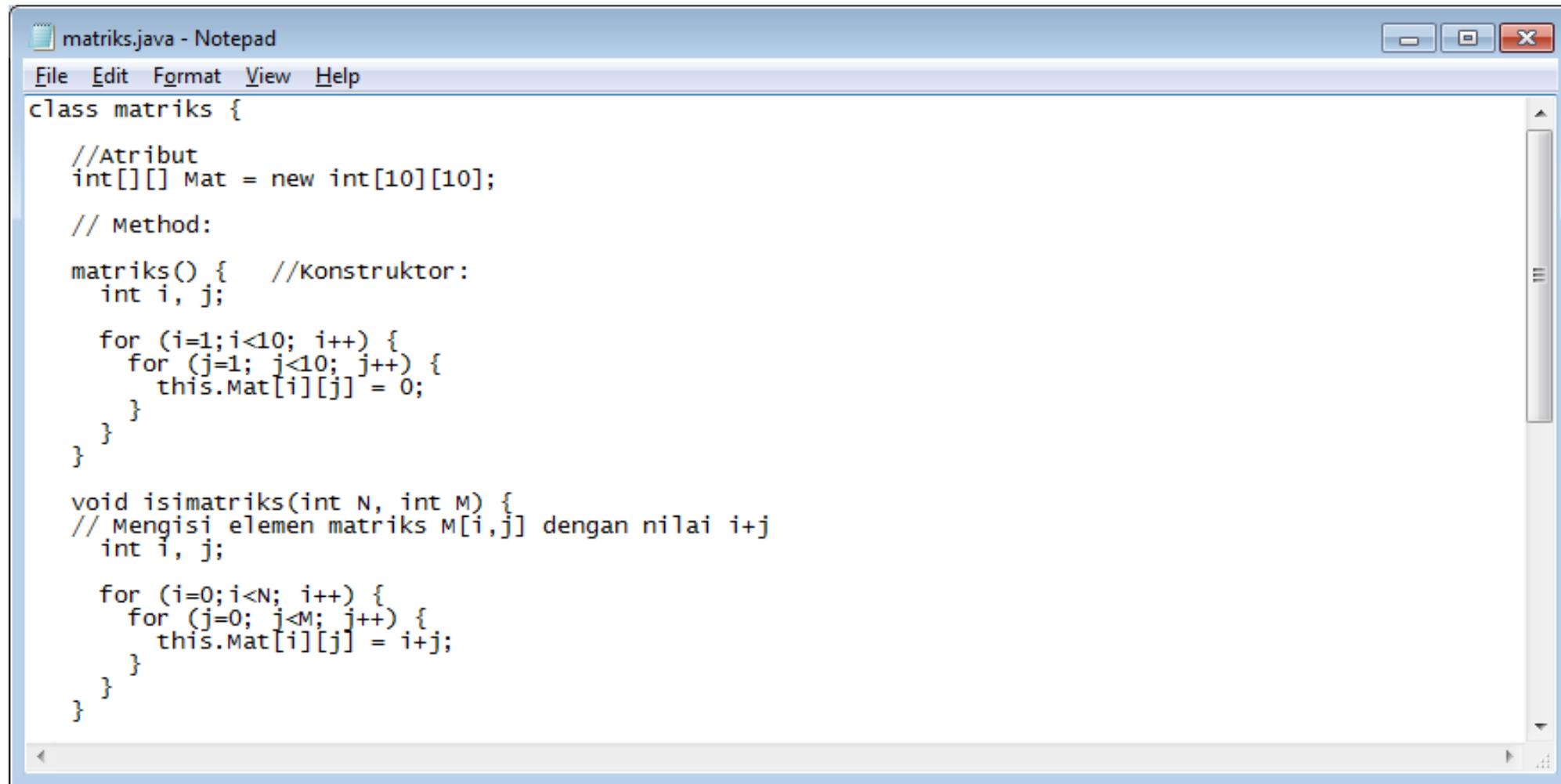


```
Administrator: Command Prompt
E:\Belajar Bahasa Java>java DriverMhs
Ketikkan data mahasiswa:
Nama: Aldi
NIM: 13514001
Usia: 18

Data mahasiswa:
Nama: Aldi
NIM: 13514001
Usia: 18

E:\Belajar Bahasa Java>_
```

Kelas Matriks



```
matriks.java - Notepad
File Edit Format View Help
class matriks {
    //Atribut
    int[][] Mat = new int[10][10];

    // Method:

    matriks() { //Konstruktor:
        int i, j;

        for (i=1;i<10; i++) {
            for (j=1; j<10; j++) {
                this.Mat[i][j] = 0;
            }
        }
    }

    void isimatriks(int N, int M) {
        // Mengisi elemen matriks M[i,j] dengan nilai i+j
        int i, j;

        for (i=0;i<N; i++) {
            for (j=0; j<M; j++) {
                this.Mat[i][j] = i+j;
            }
        }
    }
}
```

Kelas DriverMatriks

(yang menggunakan kelas Mahasiswa)

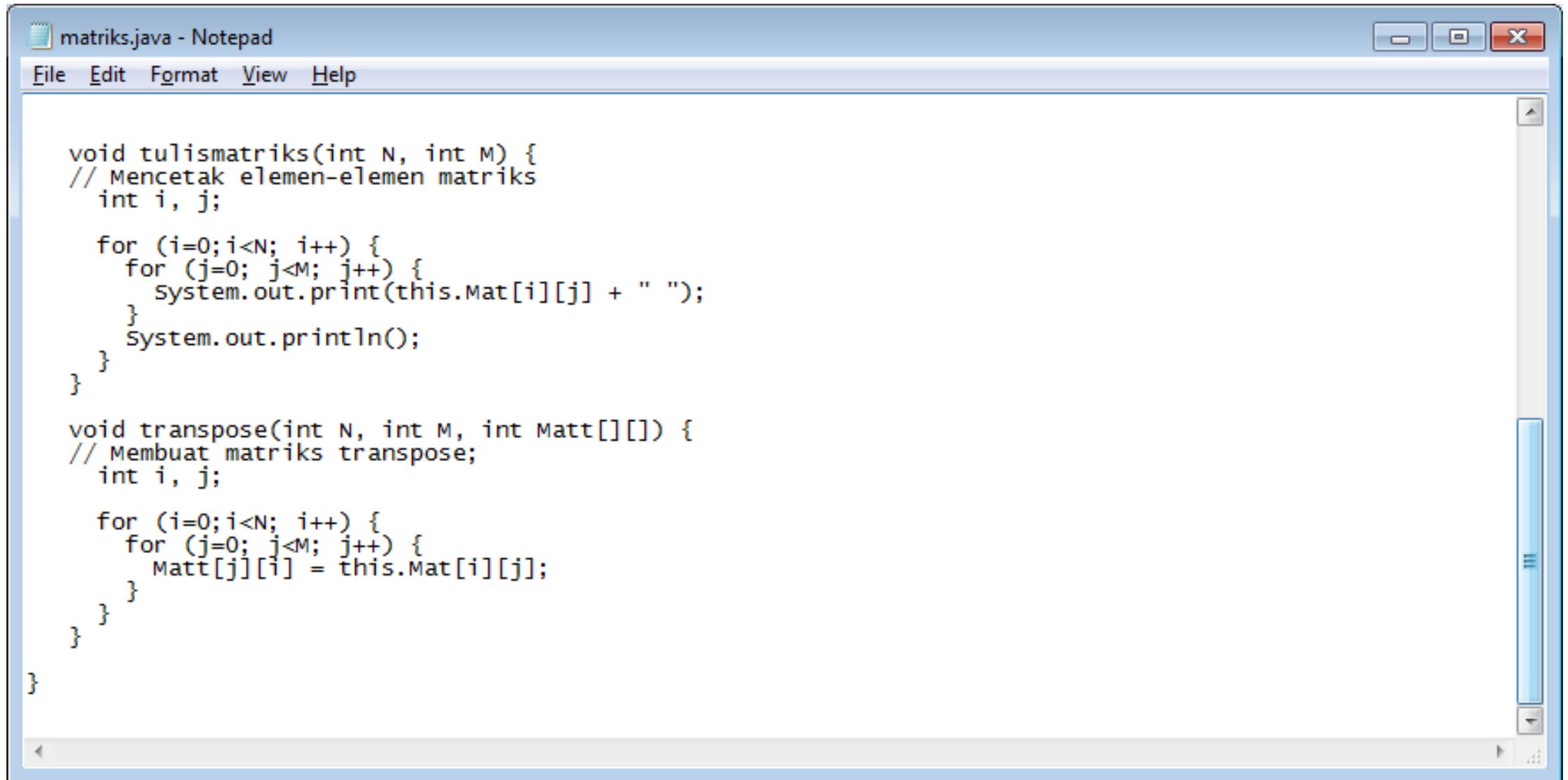
```
DriverMatriks.java - Notepad
File Edit Format View Help
/* kelas driver Matriks */
class DriverMatriks {
    public static void main(String[] args) {
        // buat objek P1
        matriks M = new matriks();

        // mengisi matriks
        M.isimatriks(3,4);

        // menulis matriks
        System.out.println("Isi matriks: ");
        M.tulismatriks(3, 4);

        // membuat matriks transpose
        int[][] M2;
        M2 = new int[4][3];
        M.transpose(3,4,M2);

        // Tulis matriks transpose
        System.out.println("Isi matriks transpose: ");
        int i, j;
        for (i=0; i<4; i++) {
            for (j=0; j<3; j++) {
                System.out.print(M2[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```



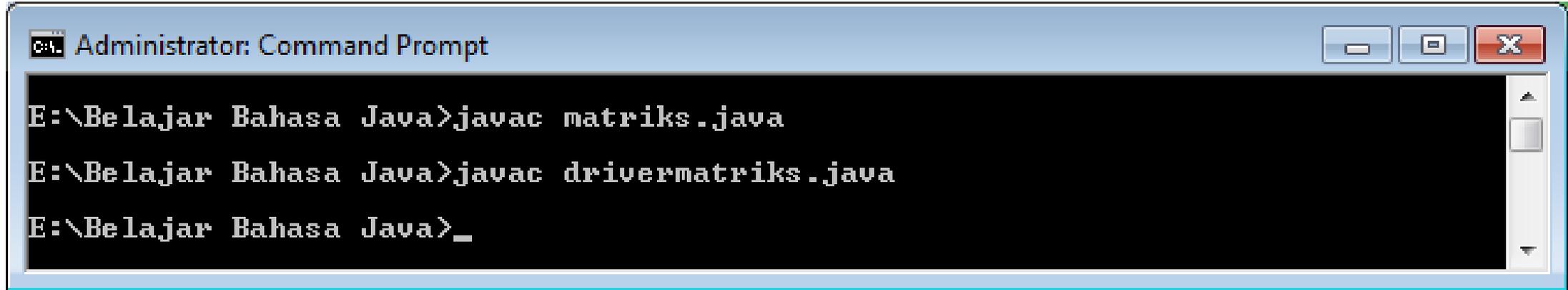
```
void tulismatriks(int N, int M) {
// Mencetak elemen-elemen matriks
int i, j;

for (i=0; i<N; i++) {
for (j=0; j<M; j++) {
System.out.print(this.Mat[i][j] + " ");
}
System.out.println();
}
}

void transpose(int N, int M, int Matt[][]) {
// Membuat matriks transpose;
int i, j;

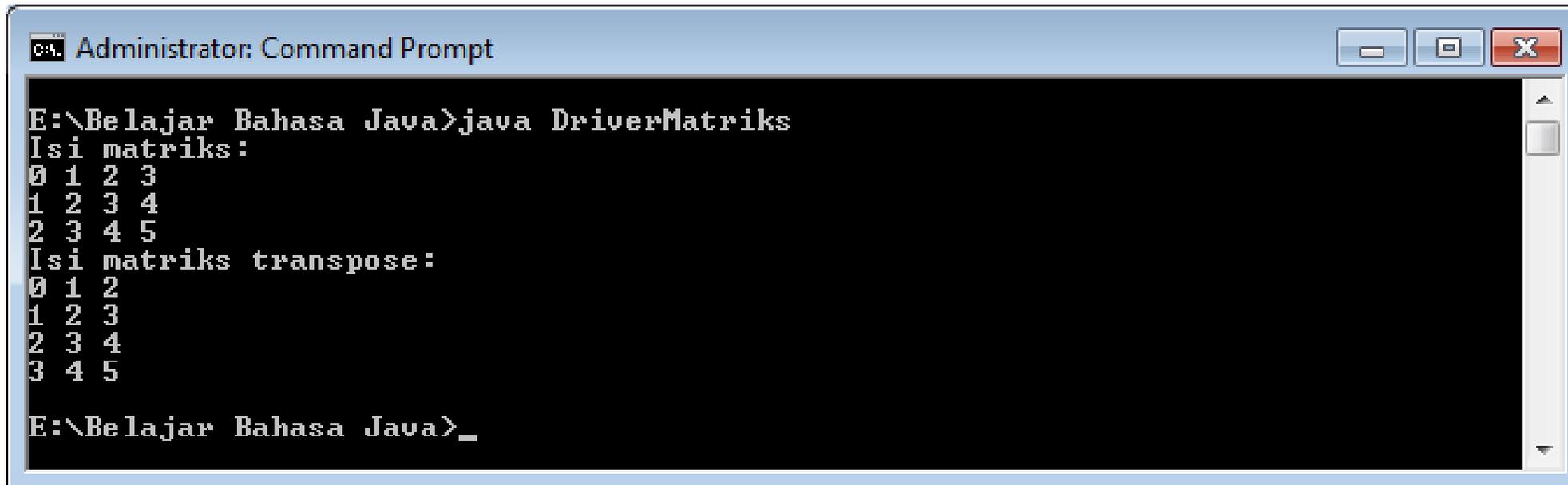
for (i=0; i<N; i++) {
for (j=0; j<M; j++) {
Matt[j][i] = this.Mat[i][j];
}
}
}
}
```

- Kompilasi masing-masing `matriks.java` dan `DriverMatriks.java`



```
Administrator: Command Prompt
E:\Belajar Bahasa Java>javac matriks.java
E:\Belajar Bahasa Java>javac drivermatriks.java
E:\Belajar Bahasa Java>_
```

- Jalankan kelas `DriverMatriks.class`



```
Administrator: Command Prompt
E:\Belajar Bahasa Java>java DriverMatriks
Isi matriks:
0 1 2 3
1 2 3 4
2 3 4 5
Isi matriks transpose:
0 1 2
1 2 3
2 3 4
3 4 5
E:\Belajar Bahasa Java>_
```

Referensi

Materi “Pengantar Pemrograman Bahasa Java” diambil dari berbagai sumber, antara lain:

1. Arief Bahtiar S.T, M.T, Ivan Kurniawan, *Fundamental Java 2 Platform Application Developer*, ComLabs IT Course ITB.
2. Adi Nuralim, *Java Virtual Machine*, <http://belajarjava-19.blogspot.co.id/2011/05/java-virtual-machine-jvm.html>, tanggal akses 3 Septemebr 2015
3. Wikipedia