

BAB I

PENGENALAN JAVA

1.1 Sejarah Java

(Menurut Hakim , Rachmad dan Ir. Surarto Msi. 2009, halaman : 1) Java bermula dari proyek penelitian perubahan Sun Microsystems dengan nama santi Green pada tahun 1991. Terdapat prediksi bahwa mikroprosesor akan digunakan luas pada peralatan-peralatan elektronik. Karena adanya bermacam tipe mikroprosesor, maka dibutuhkan sebuah bahasa pemrograman yang dapat berjalan di semua mikroprosesor.

Terciplah sebuah bahasa pemrograman baru. Oleh James Gosling, yaitu salah satu orang yang berperan besar dalam proyek tersebut, program ini diberi nama Oak. Sesuai dengan pohon Oak yang tumbuh dan bisa dilihat melalui jendela kerjanya di Sun Microsystems.

Selah beberapa waktu kemudian, ditemukan bahwa sudah ada bahasa pemrograman dengan nama Oak. Akhirnya setelah beberapa pegawai Sun mengunjungi sebuah kedai kopi. Nama bahasa pemrograman ini diganti dengan Java. Java merupakan salah satu jenis biji kopi yang ada di kedai tersebut yaitu biji kopi Jawa.

Sun Microsystems mengumumkan kehadiran bahasa Java secara formal ditahun 1995. Bahasa ini mulai disambut hangat masyarakat luas seiring dengan meledaknya era internet.

Dan yang dimana Java ini menggunakan bahasa pemrograman OOP (Object Oeinted Programming) yaitu bentuk pemodeln programnya diaplikasikan dalam bentuk objek-objek dan sedangkan program dalam bentuk tekstual merupakan pemodelan programnya diuraikan secara detail.

1. 2 Keunggulan Java

(Menurut Hakim , Rachmad dan Ir. Surarto Msi. 2009, halaman : 6)Dibanding dengan bahasa pemrograman lain, Java mempunyai beberapa kelebihan yang membuatnya dipakai luas terutama untuk aplikasi enterprise. Java bahkan diprediksi menjadi bahasa masa depan. Java telah banyak diajarkan di lingkungan pendidikan dan dipakai secara luas oleh perusahaan dan lembaga berskala internasional, seperti IBM dan NASA.

Beberapa keunggulan Java, antara lain:

1. Relative Mudah Dipelajari

Java merupakan bahasa pemrograman berorientasi objek (*Object Oriented Programming*) yang mudah dipelajari. Terutama bagi yang sudah familiar dengan Cc, C++, atau C#.

Java meningkatkan pemrograman objek pada C++ dengan menghilangkan pointer pada tipe data biasa dan *multiple inheritance*. Java juga menggunakan pengalokasian memori secara otomatis termasuk menerapkan *garbage collection*.

2. Mudah dikembangkan

Dengan pemrograman berorientasi objek, komponen program Java terpecah menjadi satu atau beberapa class. Pengembangan bahasa Java sangat mudah karena tinggal mengubah atau menambah class yang pernah dibuat. Selin

itu, kita juga dapat menggunakan class-class yang sudah ada dalam Java API ataupun dibuat oleh pihak ketiga.

Nah maksud dari mudah dikembangkan disini adalah mudah untuk dihubungkan dengan aplikasi yang lain, misalkan dalam JDBC bisa dihubungkan dengan Oracle, MySQL dll.

1.3 Instalasi Java

Agar dapat mempelajari Java pada computer Anda, maka perlu dilakukan langkah-langkah berikut:

1. Instalasi JDK ke dalam computer Anda.
2. Install IDE Java yang akan dipakai bisa menggunakan NetBeans, Eclipse, dan lain-lain.

1.4 Instalasi Java Development Kit (JDK)

(Menurut Hakim , Rachmad dan Ir. Surarto Msi. 2009, halaman : 7) Agar dapat bekerja dan mempelajari Java, dalam computer Anda harus terinstal JDK. Java Development Kit (JDK) atau pernah disebut dengan Java Software Development Kit (Java SDK), merupakan seperangkat aplikasi yang digunakan untuk mengembangkan program dengan bahasa Java. Didalamnya terdapat tiga komponen utama, yaitu Java API, Java Virtual Machine, dan Java Compiler. Dapat dilihat bahwa JDK ini adalah JRE ditambah dengan Java Compiler.

JDK merupakan perangkat minimal yang harus dimiliki oleh pemrograman Java. Aplikasi ini bisa di-download melalui situs.

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Lakukan langkah-langkah berikut untuk menginstal JDK :

1. Klik ganda atau klik kanan → Run As Administrator pada file **JDK** untuk melakukan instalasi.
2. Klik tombol **Accept** pada **License Agreement**.

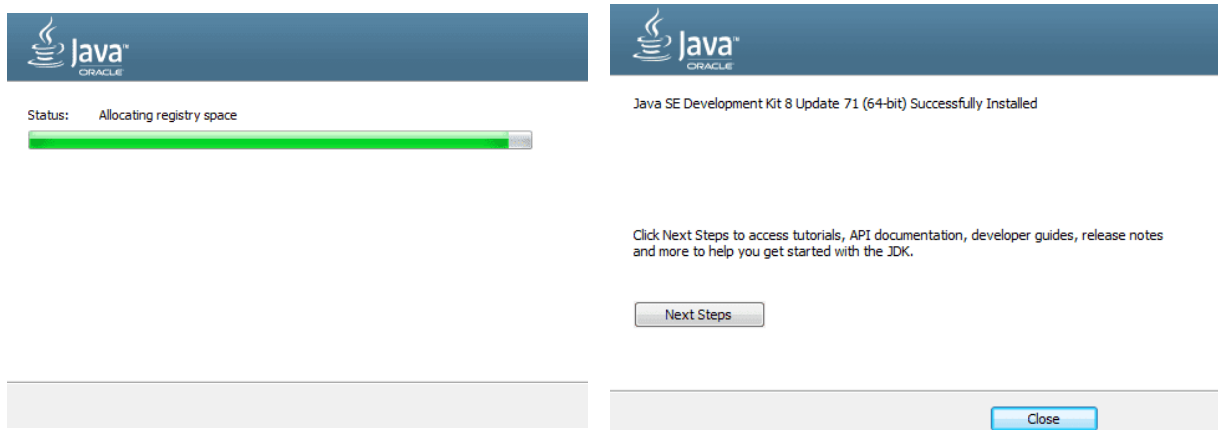


Gambar 1.1 File JDK.

3. Klik tombol **Next**.



Gambar 1.2 Proses Instalasi.



Gambar 1.3 Instalasi selesai.

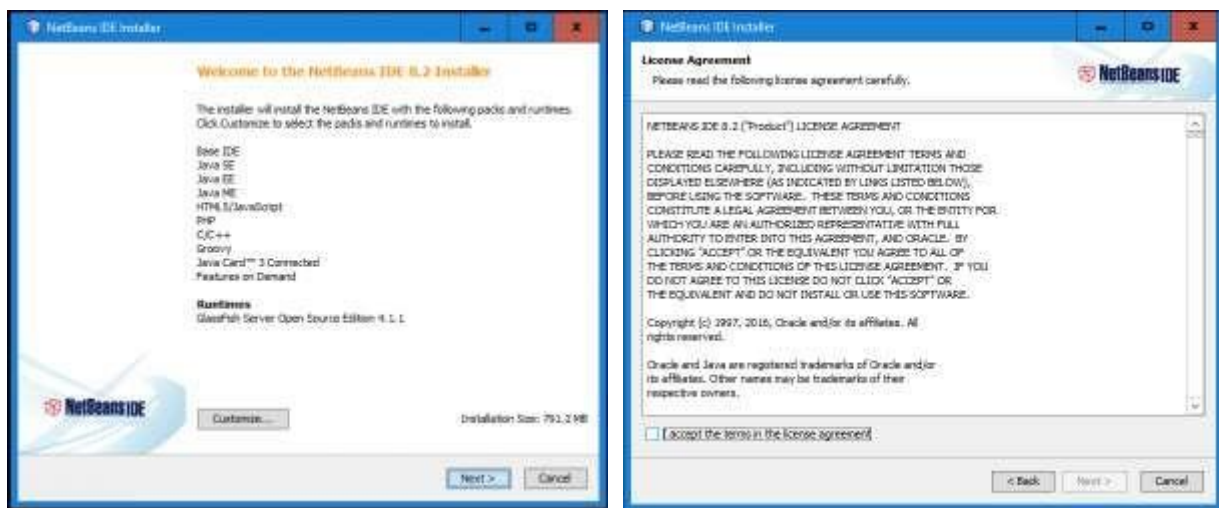
Setelah proses instalasi JDK berhasil , komputer Anda siap digunakan untuk menjalankan program Java.

1.5 Instalasi NetBeans

NetBeans merupakan merupakan integrated Development Environment (IDE) yang telah banyak dipakai untuk mengembangkan program dalam bahasa Java. Aplikasi ini dapat di download pada <https://netbeans.org/downloads/index.html>

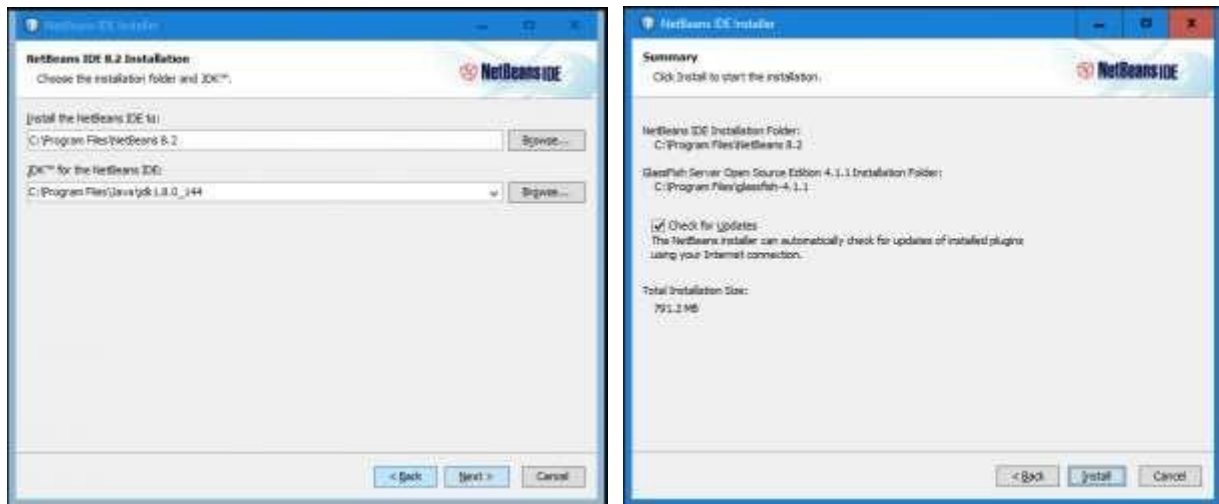
Lakukan langkah-langkah berikut untuk menginstal NetBeans:

1. Jalankan aplikasi NetBeans yang ada di dalam CD atau Komputer.
2. Klik tombol **Next** untuk melanjutkan instalasi, centang pada **I accept the terms in the license agreement** lalu klik **Next**.



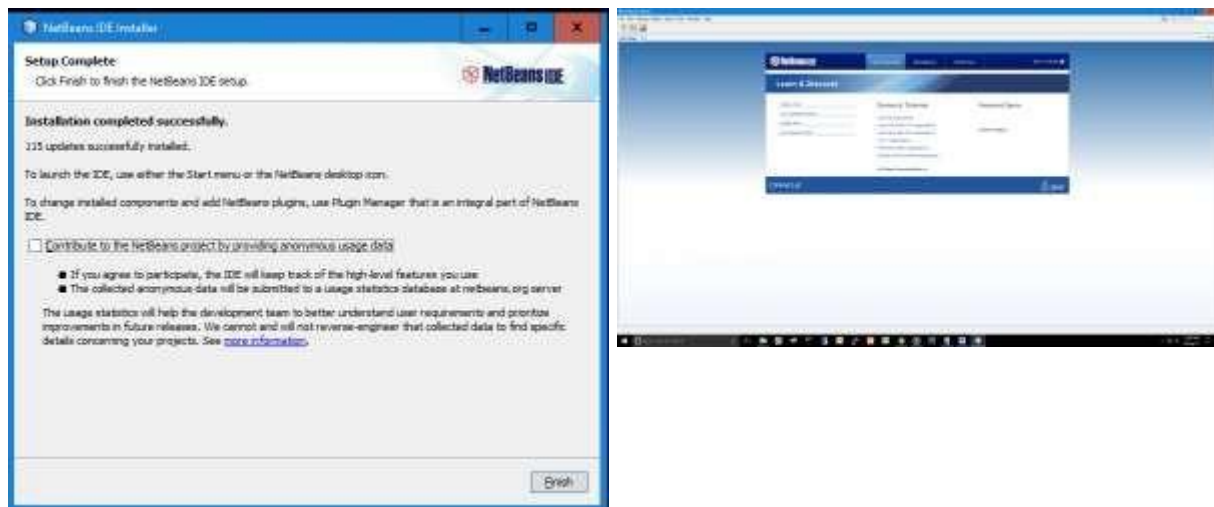
Gambar 1.4 Memulai instalasi Netbeans.

3. Klik tombol **Next** untuk melakukan instalasi pada folder yang telah ditentukan secara otomatis.
4. Klik tombol **install** untuk mulai melakukan instalasi.



Gambar 1.5 Proses Instalasi Netbeans.

5. Tunggu hingga install selesai.
6. Klik tombol **Finish**. Maka NetBeans selesai di install.



Gambar 1.6 Proses Instalasi Netbeans Selesai.

Jadi kesimpulannya perbedaan antara JDK dan IDE itu adalah kalau JDK itu perangkat lunak pendukungnya untuk menjalankan sebuah program. Kalau IDE itu aplikasi-aplikasi utamanya dalam menjalankan sebuah program Java.

BAB II

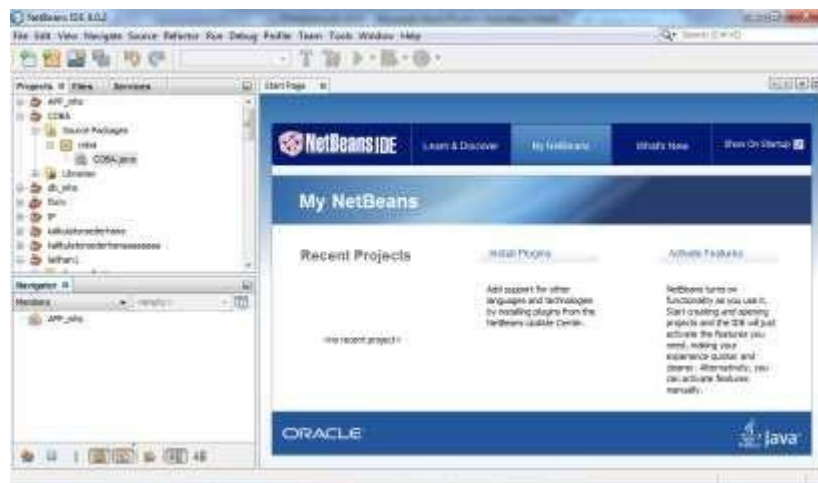
Input dan Output Dari Keyboard

2.1 Input/Output dari keyboard pada NetBeans

I/O merupakan sebuah singkatan dari Input Output, nah kali ini kita akan menggunakan `java.util.scanner`, `java.util.scanner` merupakan library dari Java yang menyediakan fasilitas untuk menscan apa yang diinput dari keyboard.

Berikut adalah tutorial bagaimana membuat input output di Java :

1. Buka aplikasi NetBeans → **File** → **New Project**.



Gambar 2.1 Tampilan Netbeans 8.o.2

2. Setelah itu kita pilih **Java Application** → **Next** → beri nama pada **Project Name** kemudian klik **Finish**.



Gambar 2.2 Membuat Projek Java.

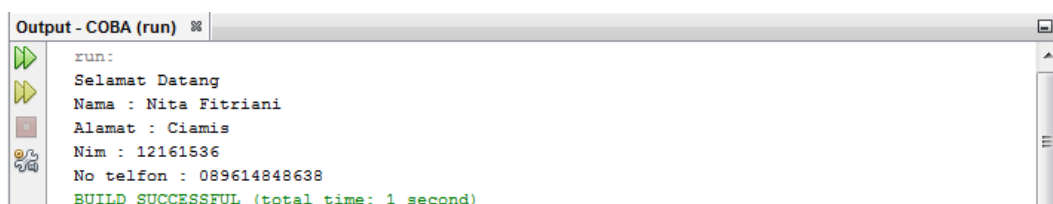
3. Tulis source code dibawah ini :

```
package coba;

/**
 *
 * @author ACER
 */
public class COBA {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        System.out.println("Selamat Datang") ;
        // println artinya untuk enter atau garis baru
        System.out.println("Nama : Nita Fitriani");
        System.out.println("Alamat : Ciamis");
        System.out.println("Nim : 12161536");
        System.out.println("No telfon : 089614848638");
    }
}
```

4. Untuk menguji koding diatas klik kanan → **Run File** maka akan muncul tampilan seperti gambar dibawah ini.



Gambar 2.3 Hasil pengujian *Source code*.

2.2 Scanner

Kali ini kita akan menggunakan Scanner di Java. Berikut adalah tutorial bagaimana membuat input output dengan scanner di Java:

1. Tuliskan source code dibawah ini.

```
package latihanscanner;

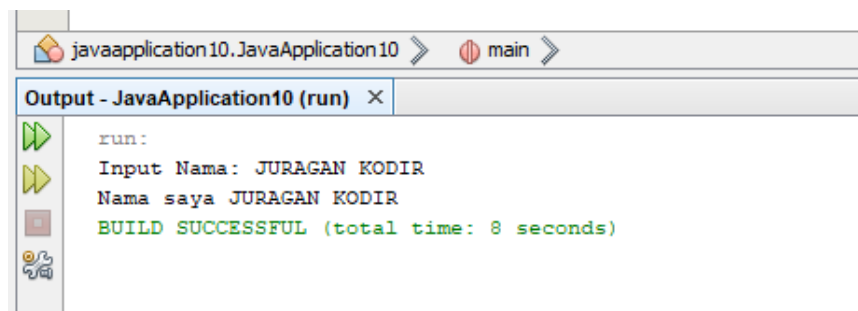
import java.util.Scanner;

/**
 *
 * @author ACER
 */
public class LatihanScanner {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner input = new Scanner (System.in);
        System.out.print("Input Nama: ");
        String nama = input.nextLine();
        System.out.println("Nama saya "+nama);
    }
}
```

2. Untuk menguji koding diatas klik kanan → **Run File** maka akan muncul

tampilan seperti gambar dibawah ini.



Gambar 2.4 Hasil pengujian *Source code scanner*.