

Praktikum 1

Pengenalan Lingkungan JAVA dan Konsep Pemrograman Berorientasi Objek

Tujuan

Mengetahui cara instalasi JDK (Java Development Kit) dan IDE NetBeans.

Memberikan latar belakang serta pemahaman tentang konsep pemrograman berorientasi obyek dan perbandingannya dengan pemrograman prosedural.

Dasar Teori

Ada beberapa software pendukung yang diperlukan diantaranya Java SDK dan Java IDE. Buat pemula bahasa pemrograman Java, proses mendownload paket bahasa pemrograman Java dirasa cukup membingungkan. Mau mendownload Java SDK (Software Development Kit) yang mana ?

- Java SE (Standard Edition)
- Java ME (Micro Edition, kadang disebut Mobile Edition)
- Java EE (Enterprise Edition)

Kalau saya ingin membuat aplikasi untuk kantor, apakah tepat jika saya mendownload Java EE untuk belajar membuat aplikasi tersebut ? Apakah standard edition “hanya” memuat aplikasi terbatas (namanya juga standar). Apakah nanti tidak bermasalah jika awalnya saya menggunakan Java SE dan baru nanti menggunakan Java EE ? Apakah JRE (Java Runtime Environment) perlu diinstall terpisah ?

Jika anda mulai belajar Java, disarankan menggunakan Java SE. Java SE bukan berarti Java dengan kemampuan terbatas. Standard disini lebih kepada paket fundamental Java, bukan batasan dalam arti fasilitas yang serba kurang.

“Java Platform, Standard Edition (also known as Java 2 Platform) lets you develop and deploy Java applications on desktops and servers, as well as today’s demanding Embedded and Real-Time environments. Java SE includes classes that support the development of Java Web Services and provides the foundation for Java Platform, Enterprise Edition (Java EE).” (www.cnet.com)

Java EE biasanya sudah dibundel pada paket development tools maupun application server yang kita gunakan. Netbeans, Sun Java Creator atau Eclipse bisa secara paket menyertakannya.

Java ME sesuai dengan namanya ditujukan untuk aplikasi embedded pada mobile / micro devices, seperti PDA, HP, Barcode Scanner / PDT (Portable Data Terminal) dll.

JRE (Java Runtime Environment) adalah paket lingkungan yang dibutuhkan jika ingin menjalankan aplikasi Java. Jika kita hanya ingin menjalankan aplikasi Java tanpa membuat atau mengubahnya, kita tidak perlu melakukan instalasi Java SDK. Yang diperlukan hanyalah JRE ini. Jika kita melakukan instalasi Java SDK, kita tidak perlu melakukan instalasi JRE, karena JRE ini sudah dibundel dalam Java SDK.

Dengan pemahaman yang jelas mengenai perbedaan masing-masing Java SDK, kita bisa belajar Java dengan lebih tenang dan nyaman. Sekarang marilah kita mulai menginstal Java SDK beserta IDE-nya.

A. Pengenalan Lingkungan Java

Percobaan 1 : Instalasi JDK

Untuk memulai instalasi JDK, Anda bisa mendownloadnya dari website resmi Sun ataupun Oracle. Instalasi JDK versi terbaru dapat didownload di website :
<http://java.sun.com/javase/downloads/index.jsp> atau
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

“Jika Anda menggunakan link yang pertama, tetap saja Anda akan menuju link yang kedua. Karena Java pertama kali dimiliki oleh Sun Microsystem, kemudian dibeli oleh Oracle. Namun sayangnya James Gosling, sang arsitek bahasa pemrograman Java akhirnya turut hengkang dari Oracle pada Januari 2010 lalu. Di Oracle, Gosling bekerja sebagai Chief Technology Officer. Posisi yang sama ketika ia bekerja pada Sun Microsystem. Pada tahun 1991, ia tergabung dalam project Oak, yang merupakan cikal bakal bahasa pemrograman Java.” (detik.com - Senin, 12/04/2010 12:48 WIB)



Gambar: Tampilan Download JDK

Kemudian pilih file yang Anda inginkan. Ada 4 file yang dapat Anda pilih



Gambar: Pilihan JDK

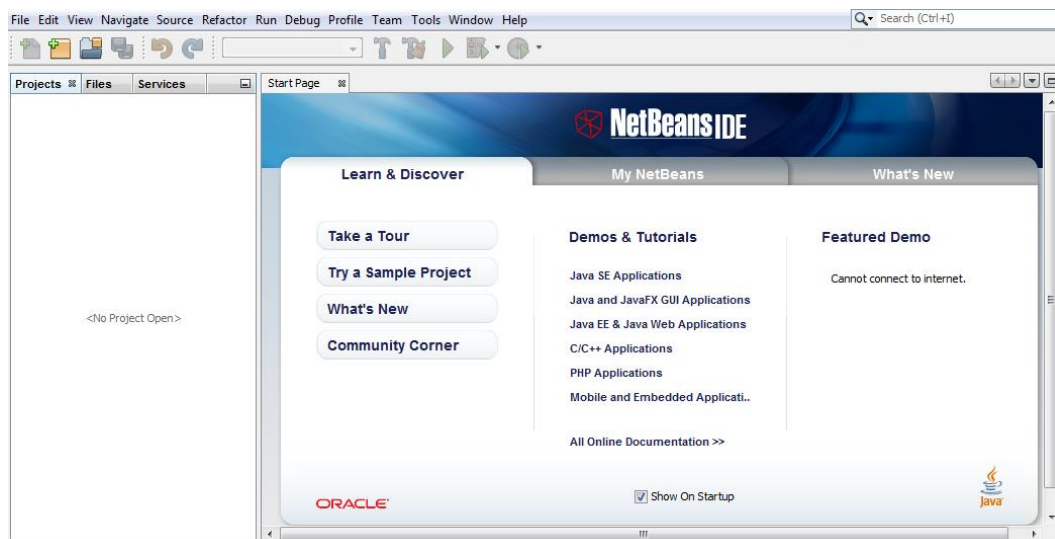
Jika demikian, apakah bisa jika kita langsung memilih paket Netbeans + Java SE misalnya? Jawabnya bisa saja, meski instalasi secara terpisah-pun tidak ada masalah. Tentu saja Java SE harus diinstall lebih dahulu, baru Netbeans menyusul. Jika anda mulai belajar Java, disarankan menggunakan Java SE. Jadi anda bisa mendownload paket Java Platform (JDK) 7u6 ataupun yang sudah *bundled* dengan NetBeans. Kalau sudah didownload, kemudian jalankan file yang sudah didownload dan ikuti petunjuk instalasi. Saya untuk instalasi software tidak perlu saya jelaskan detail disini, karena saya yakin Anda sudah “pakar” dalam instal-menginstal 😊.

Setelah itu coba periksa dekstop Anda, akan tampil logo NetBeans (di sini saya menggunakan NetBeans IDE versi 7.2, saya sarankan Anda menggunakan versi yang lebih update). Kemudian silahkan double click untuk menuju ke aplikasi NetBeans IDE.



Gambar: Logo NetBeans IDE versi 7.2 di dekstop.

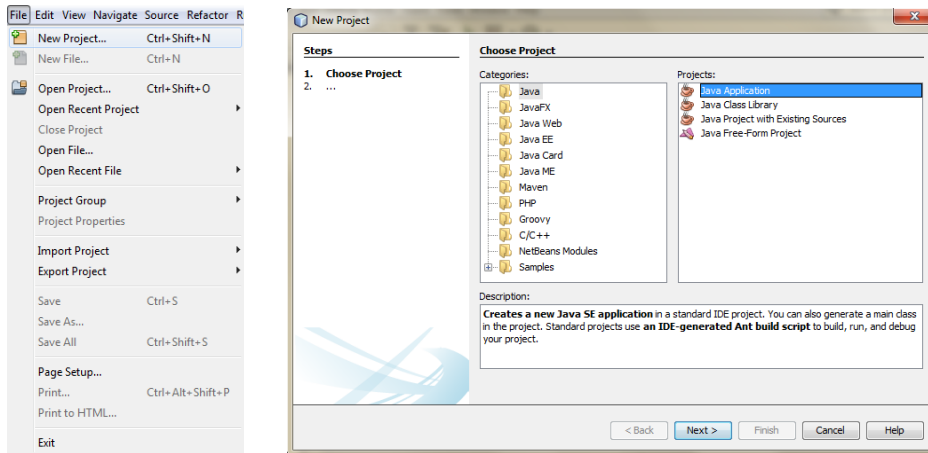
Pertama kali akan tampil jendela seperti di bawah ini. Saya sarankan Anda untuk mencoba-coba NetBeans IDE dengan membaca tutorial dan mencoba demo program yang ada di NetBeans IDE.



Gambar: Tampilan Jendela Kerja NetBeans IDE

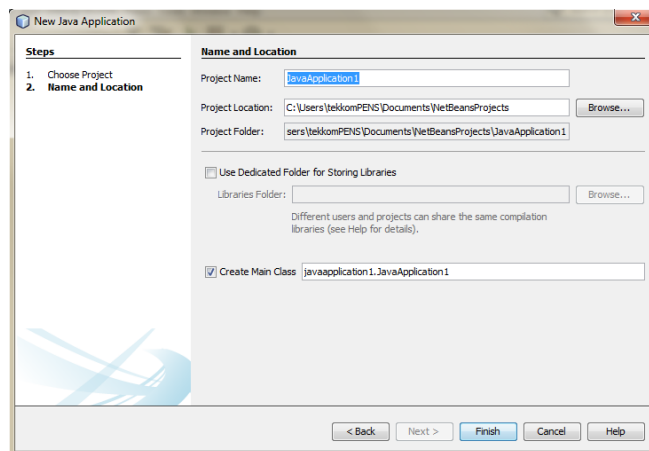
Percobaan 2 : Memulai membuat project baru

Percobaan ini kita akan membuat project pertama kita. Setiap kali memulai praktikum, hal inilah yang Anda harus lakukan. Dimulai dengan memilih menu : **File > New Project**. Kemudian pilih Categories: **Java** dan Projects: **Java Application**. Kemudian tekan **Next**.



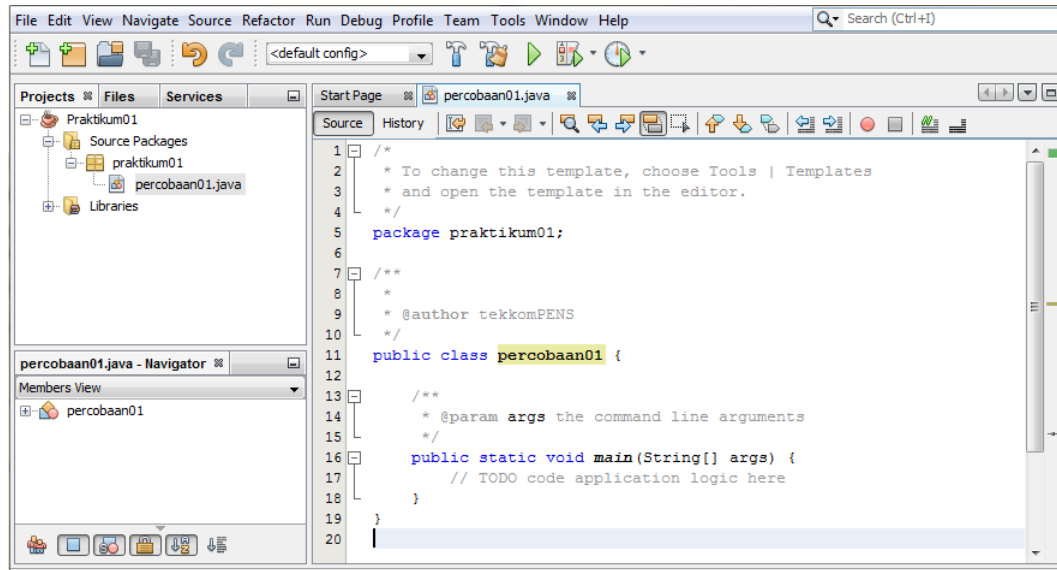
Gambar: Tampilan jendela New Project

Kemudian mengisi nama dan lokasi file pada kolom Name and Location. **Project Name** diisi dengan praktikum “Praktikum01”. **Project Location** diisi dengan lokasi penyimpanan file. **Create Main Class** diisi dengan nama class, misalnya “helloworld”.



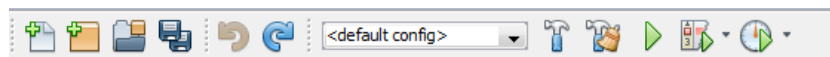
Gambar: Tampilan jendela New Project

Cek kembali informasi yang Anda masukkan, setelah selesai tekan tombol **Finish**. Jika langkah yang Anda lakukan benar, maka akan muncul jendela seperti sebagai berikut.



Gambar: Tampilan jendela New Project

Terdiri dari menu, shortcut, project navigator, editing page. Berikut ini adalah gambar shortcut beberapa perintah yang sering digunakan. Dekatkan kursor Anda untuk mendapatkan informasi lebih lanjut.



Gambar: Shortcut perintah di NetBeans

Masukkan program dibawah ini, jalankan, amati hasilnya dan berikan analisa Anda.

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package praktikum01;

/**
 *
 * @author tekompENS
 */
public class percobaan01 {

    /**
     * @param args the command line arguments
     */

    public static void main(String[] args) {
        // TODO code application logic here

        System.out.println("Hello World!"); //menampilkan string
    }
}

```

Untuk menyederhanakan program, Anda dapat menghilangkan *comment*. Dengan demikian program menjadi lebih ramping. Seperti pada program dibawah ini, jalankan, amati hasilnya dan berikan analisa Anda.

```
public class percobaan02 {  
  
    public static void main(String[] args){  
  
        String name,addr;  
        name = "Rudiawan";  
        addr = "Bengawan Solo";  
  
        System.out.println("Name:"+ name); //menampilkan string name  
        System.out.println("Addres:"+ addr); //menampilkan string addr  
  
    }  
  
}
```

B. Konsep Pemrograman Berorientasi Objek

Bahasa pemrograman prosedural berfokus pada fungsi atau prosedural dan bagaimana komputer menangani pekerjaan (task). Sedangkan pemrograman berorientasi object fokus pada satu kesatuan baik fungsi, data yang terlibat dan pada pekerjaan apa yang akan digunakan pada komputer.

Percobaan 1:

Berikut ini contoh program penjumlahan vector yang diselesaikan secara prosedural. Jalankan di Visual C++, amati hasilnya dan berikan analisa Anda.

```
#include <stdio.h>  
#include <conio.h>  
  
typedef struct vector  
{  
    double x;  
    double y;  
    double z;  
} Vec3;  
  
void incnn(void)  
{  
    Vec3 v1,v2,v3;  
    v1.x = 1.0;  
    v1.y = 2.0;  
    v1.z = 3.0;  
  
    v2.x = 1.0;
```

```

        v2.y = 2.0;
        v2.z = 3.0;

        v3.x = v1.x + v2.x;
        v3.y = v1.y + v2.y;
        v3.z = v1.z + v2.z;

        printf ("v3--> %2.1f %2.1f %2.1f \n", v3.x,v3.y,v3.z);
    }

main ()
{
    incnn ();
    getch ();
}

```

Percobaan 2:

Berikut ini contoh program penjumlahan vector yang diselesaikan dengan cara prosedural dengan menggunakan tipe data struct. Jalankan di Visual C++, amati hasilnya dan berikan analisa Anda.

```

#include <stdio.h>
#include <conio.h>

typedef struct vector
{
    double x;
    double y;
    double z;
} Vec3;

Vec3 VecSet(double a, double b, double c);
Vec3 VecAdd (Vec3 va, Vec3 vb);
void VecPrint( Vec3 v);

main(){
    Vec3 v1, v2,v3;
    v1 = VecSet(1.0,2.0,3.0);
    v2 = VecSet(1.0,2.0,3.0);
    v3 = VecAdd(v1,v2);

    VecPrint(v3);
    getch();
}

Vec3 VecSet(double a,double b,double c)
{
    Vec3 tmp;
    tmp.x =a;
    tmp.y =b;
    tmp.z =c;

    return tmp;
}

Vec3 VecAdd(Vec3 va, Vec3 vb)
{
    Vec3 tmp;
    tmp.x = va.x +vb.x;
    tmp.y = va.y +vb.y;
}

```



```

        tmp.z = va.z +vb.z;

        return tmp;
    }

void VecPrint(Vec3 v)
{
    printf("v3-->%2.1f %2.1f %2.1f \n",v.x,v.y,v.z);
}

```

Percobaan 3:

Contoh program penjumlahan vector yang diselesaikan dengan cara pemrograman berorientasi obyek. Jalankan di Visual C++, amati hasilnya dan berikan analisa Anda.

```

#include<iostream.h>

class Vector{
    long x;
    long y;
    long z;

public:
    void setValue(long,long,long);
    Vector addValue(Vector);
    void printValue(Vector);
};

void Vector::setValue(long a, long b, long c){
    x=a;
    y=b;
    z=c;
}

Vector Vector::addValue(Vector temp){
    temp.x = x + temp.x;
    temp.y = y + temp.y;
    temp.z = z + temp.z;
    return temp;
}

void Vector::printValue(Vector v) {
    cout<<"V3 -> "<<v.x<<" "<<v.y<<" "<<v.z;
}

void main(){
    Vector v1,v2,v3;

    v1.setValue(1,2,3);
    v2.setValue(1,2,3);

    v3 = v1.addValue(v2);
    v3.printValue(v3);
}

```

Percobaan 4:

Berikut ini contoh kelas VolcanoRobot yaitu obyek robot eksplorasi yang memiliki kecepatan, status dan temperatur. Jalankan di IDE Java, amati hasilnya dan berikan analisa Anda.

```
public class VolcanoRobot {  
  
    String status;  
    int speed;  
    float temperature;  
  
    void checkTemperature() {  
        if(temperature>660) {  
            status="returning home";  
            speed=5;  
        }  
    }  
  
    void showAttribute() {  
        System.out.println("Status:"+ status);  
        System.out.println("Speed:"+ speed);  
        System.out.println("Temperature:"+ temperature);  
    }  
  
    public static void main(String[] args) {  
  
        VolcanoRobot dante= new VolcanoRobot();  
  
        VolcanoRobot dante2= new VolcanoRobot();  
  
        dante.status = "exploring";  
        dante.speed = 2;  
        dante.temperature = 50;  
  
        dante.showAttribute();  
  
        System.out.println("increasing speed to 3");  
        dante.speed = 3;  
        dante.showAttribute();  
  
        System.out.println("changing temperature to 67");  
        dante.temperature = 670;  
        dante.showAttribute();  
  
        System.out.println("checking the temperature");  
        dante.checkTemperature();  
        dante.showAttribute();  
    }  
}
```

Tugas

1. Menghitung luas dan keliling lingkaran

Buatlah program untuk menghitung luas dan keliling lingkaran.

Rumus:

Luas lingkaran = $\text{PI} \times \text{jari-jari}^2$

Keliling lingkaran = $2 \times \text{PI} \times \text{jari-jari}$

2. Mengkonversi suatu nilai dari Celcius ke Fahrenheit atau sebaliknya

Buatlah suatu program untuk mengkonversi suatu nilai dari Celcius ke Fahrenheit atau sebaliknya.