

**Modul Pelatihan :  
Pemrograman Berorientasi Obyek dengan Java  
(Bagian 1)**

---



Disusun oleh:

**Herika Hayurani, M.Kom  
Nova Eka Diana, S.Kom, M.Eng  
Nurmaya, S.Kom, M.Eng**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS YARSI  
APRIL 2015**

## DAFTAR ISI

DAFTAR ISI .....	i
DAFTAR TABEL .....	iii
DAFTAR GAMBAR .....	iv
PRAKTIKUM 1    PENGENALAN LINGKUNGAN JAVA.....	1
1.1 Tujuan .....	1
1.2 Deskripsi .....	1
1.3 Praktikum.....	1
1.3.1 Persiapan Perangkat Lunak yang Dibutuhkan.....	1
1.3.2 Konfigurasi Path Java .....	2
1.3.3 Program Java Menggunakan Editor Sederhana.....	3
1.3.4 <i>Runtime Error</i> .....	4
1.3.5 <i>Compile Error</i> .....	5
PRAKTIKUM 2    VARIABEL DAN TIPE DATA .....	6
1.1 Tujuan .....	6
1.2 Deskripsi .....	6
1.3 Praktikum.....	7
1.3.1 Membuat Project di Eclipse .....	7
1.3.2 Mencari Nilai Rata-rata.....	10
1.3.3 Mencari Nilai Maksimum dari Tiga Buah Bilangan .....	12
PRAKTIKUM 3    INPUT DARI KEYBOARD .....	14
3.1 Tujuan .....	14
3.2 Deskripsi .....	14
3.2.1 Class Scanner.....	14
3.2.2 Package dan Import Package .....	15
3.2.3 API (Application Programming Interface).....	15
3.2.4 Class String.....	16

3.2.5 Class StringTokenizer .....	16
3.3 Praktikum .....	17
3.3.1 Program HelloScanner .....	17
3.3.2 Program CobaString .....	17
3.3.3 Program CobaStringTokenizer.....	18
3.3.4 Latihan 1 – Sisipkan Dong! .....	18
3.3.5 Latihan 2 – Cari Warna.....	19
3.3.6 Latihan 3 – Satukan! .....	20
PRAKTIKUM 4 OBJECT, CLASS, METHOD, DAN CONSTRUCTOR.....	21
4.1 Tujuan .....	21
4.2 Deskripsi .....	21
4.3 Praktikum.....	22
4.3.1 Class BujurSangkar.....	22
4.3.2 Class TesBujurSangkar.....	25
4.3.3 Latihan – Persegi Panjang.....	28
PRAKTIKUM 5 INHERITANCE (BAGIAN PERTAMA) .....	29
5.1 Tujuan .....	29
5.2 Deskripsi .....	29
5.3 Praktikum.....	29
5.3.1 Class Person.....	29
5.3.2 Class Student .....	30
5.3.3 Class InternationalStudent .....	31
5.3.4 Class Teacher.....	32
5.3.5 Class MainTester .....	32
5.3.6 Memanggil Rantai Constructor.....	35
5.3.7 Menambahkan super() .....	37

## DAFTAR TABEL

Tabel 1 Package dan Class.....	15
Tabel 2 Method-method di Class String.....	16
Tabel 3 Method-method di Class StringTokenizer.....	16
Tabel 4 Class BujurSangkar.....	24
Tabel 5 Class Person.....	30
Tabel 6 Class Student .....	30
Tabel 7 Class InternationalStudent .....	31
Tabel 8 Class Teacher.....	32
Tabel 9 Class MainTester .....	33
Tabel 10 Tambahan kode di akhir method main dari class MainTester .....	34
Tabel 11 Constructor Person() .....	35
Tabel 12 Constructor Student() .....	35
Tabel 13 Constructor InternationalStudent .....	35
Tabel 14 Constructor Teacher() .....	36
Tabel 15 Modifikasi Class MainTester .....	36
Tabel 16 Constructor Person(String name).....	37
Tabel 17 Constructor Student(String name, String school, double grade).....	37
Tabel 18 Constructor InternationalStudent(String name, String school, double grade, String country) .....	37
Tabel 19 Modifikasi Class MainTester .....	38

## DAFTAR GAMBAR

Gambar 1 <i>Runtime error</i> .....	4
Gambar 2 <i>Compile error</i> .....	5

# **PRAKTIKUM 1**

## **PENGENALAN LINGKUNGAN JAVA**

*Waktu : 30 menit*

### **1.1 Tujuan**

Tujuan dari Praktikum 1 adalah sebagai berikut:

1. Mampu menginstal dan mengkonfigurasi komputer sebelum mengembangkan program Java.
2. Mampu membuat program Java sederhana menggunakan editor *text*.
3. Mampu membuat program Java yang dapat membaca masukan *user* saat *run time*.

### **1.2 Deskripsi**

Untuk dapat membuat program Java, Anda harus memiliki perangkat lunak JDK (*Java Development Kit*). Sedangkan untuk dapat menjalankan program Java yang telah dibuat, Anda harus memiliki perangkat lunak JRE (*Java Runtime Environment*). Pada praktikum pertama ini, Anda akan dituntun untuk mempersiapkan JDK dan JRE, yang meliputi mengunduh *installer* kedua perangkat lunak tersebut, menginstal di komputer Anda, dan mengkonfigurasi “Path” Java pada *Environment Variable* komputer Anda.

### **1.3 Praktikum**

#### **1.3.1 Persiapan Perangkat Lunak yang Dibutuhkan**

Langkah-langkah persiapan:

1. Unduh dan instal JDK (*Java Development Kit*) dan JRE (*Java Runtime Environment*)
  - a. Dengan menggunakan salah satu *web browser*, buka link <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.
  - b. Klik link Download JDK dan JRE dari Java SE 7 atau Java SE 8.
  - c. Pilih “Accept License Agreement”

- d. Klik Download installer sesuai dengan platform / sistem operasi komputer Anda.
  - e. Setelah installer selesai diunduh, klik dua kali installer untuk menginstalnya.
2. Unduh dan instal Eclipse for Java Developers
    - a. Dengan menggunakan salah satu *web browser*, buka link <http://www.eclipse.org/downloads/>.
    - b. Klik Download installer sesuai dengan platform / sistem operasi komputer Anda.
    - c. Setelah installer selesai diunduh, klik dua kali installer untuk menginstalnya.

### 1.3.2 Konfigurasi Path Java

Path Java harus dikonfigurasi sebelum menjalankan program Java. Berikut adalah langkah konfigurasi Java pada komputer dengan sistem operasi Windows 7.

1. Copy path folder “bin” Java JDK terbaru di direktori tempat Java JDK terinstal. Contoh: “C:\Program Files\Java\jdk1.7.0\_07\bin\”. Ingat: path di setiap komputer mungkin tidak sama, tergantung versi update JDK yang diinstal.
2. Klik “Start”, klik kanan pada “Computer”, lalu klik “Properties”.
3. Klik “Advance System Settings”, kemudian klik tab “Advance”, lalu klik tombol “Environment Variables”.
4. Cari System Variable “Path”. Jika di komputer Anda belum ada System Variable “Path”, lanjutkan ke langkah e. Jika di komputer Anda sudah ada System Variable “Path”, lanjutkan ke langkah f.
5. Klik tombol “New”, lalu ketikkan “Path”, dan klik “OK”.
6. Klik “Path” pada kotak “System Variables”, lalu klik tombol “Edit”.
7. Jika “Path” sudah ada isinya, jangan hapus isinya. Namun, tambahkan titik koma “,” di akhir isi “Path” tersebut, lalu paste path folder “bin” Java JDK yang telah diperoleh pada langkah a.
8. Klik “OK”.

### 1.3.3 Program Java Menggunakan Editor Sederhana

Selanjutnya, Anda dapat mulai menulis dan menjalankan program Java. *Compiler* adalah aplikasi yang mengkompilasi kode sumber Java menjadi *bytecode* yang dapat dijalankan pada platform-platform yang mendukung Java. *Bytecode* sama dengan file java class yang direpresentasikan dengan ekstensi file \*.class. *Compiler* direpresentasikan oleh aplikasi `javac` yang terdapat pada folder bin/ tempat hasil instalasi Java. Selanjutnya, *bytecode* akan dijalankan oleh *Java Virtual Machine* (JVM) di platform manapun (Windows/Mac/Unix). JVM direpresentasikan oleh aplikasi `java` yang terdapat pada folder bin/ tempat hasil instalasi Java.

Kali ini, Anda akan mencoba membuat program Java menggunakan editor sederhana, mengkompilasinya, dan menjalankan program Java dengan mengikuti langkah-langkah sebagai berikut:

1. Salin kode program Java berikut ke dalam editor Notepad/Notepad++/Editplus.

```
public class SelamatDatang {  
    public static void main(String[] args) {  
        System.out.println("=====");  
        System.out.println("SELAMAT DATANG DI FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS YARSI");  
        System.out.print("Nama Anda : ");  
        System.out.println(args[0]);  
        System.out.print("Nomor KTP : ");  
        System.out.println(args[1]);  
        System.out.println("=====");  
    }  
}
```

2. Simpan kode program di atas dengan nama file "SelamatDatang.java".
3. Pada sistem operasi Windows 7, buka cmd (*command prompt*) dan arahkan *prompt* ke direktori tempat Anda menyimpan kode program di atas. Pada sistem operasi Mac/Unix/Linux, buka aplikasi terminal dan arahkan *prompt* ke direktori tempat Anda menyimpan kode program di atas.



4. *Compile* kode program di atas pada cmd/terminal dengan mengetikkan perintah:

```
javac SelamatDatang.java
```

Bila terjadi *compile error*, perbaiki *error* yang muncul sesuai dengan pesan *error*.

5. Setelah *compile* berhasil dan tidak ada *error*, selanjutnya eksekusi program di atas pada cmd/terminal dengan mengetikkan perintah:

(catatan: Nama diganti dengan Nama Anda, dan No.KTP diganti dengan nomor KTP Anda)

```
java SelamatDatang [Nama] [No.KTP]
```

6. Lihat output program di cmd/terminal.

#### 1.3.4 Runtime Error

Eksekusi kembali program di atas pada cmd/terminal dengan mengetikkan perintah:

```
java SelamatDatang
```

Amati apa yang terjadi!

Pada program SelamatDatang terdapat *logic error* (kesalahan logika program) dimana akan menyebabkan kesalahan saat program dieksekusi (*runtime error*) sehingga program tidak dapat dijalankan/dieksekusi (lihat Gambar 1).

```
Herikas-MacBook-Pro:Source code RiRi$ javac SelamatDatang.java
Herikas-MacBook-Pro:Source code RiRi$ java SelamatDatang
=====
SELAMAT DATANG DI FAKULTAS TEKNOLOGI INFORMASI UNIVESITAS YARSI
Nama Anda : Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException:
0
    at SelamatDatang.main(SelamatDatang.java:6)
```

Gambar 1 Runtime error

### 1.3.5 Compile Error

Lakukan perubahan pada SelamatDatang.java seperti pada kode di bawah ini, yaitu ubah kata **static** menjadi **statict**.

```
public class SelamatDatang {  
    public statict void main(String[] args) {  
        System.out.println("=====");  
        System.out.println("SELAMAT DATANG DI FAKULTAS TEKNOLOGI INFORMASI UNIVESITAS YARSI");  
        System.out.print("Nama Anda : ");  
        System.out.println(args[0]);  
        System.out.print("Nomor KTP : ");  
        System.out.println(args[1]);  
        System.out.println("=====");  
    }  
}
```

Kompilasi dan jalankan kembali program SelamatDatang di atas.

Amati pesan kesalahan yang muncul!

Pada program SelamatDatang terdapat *syntax error* (kesalahan penulisan sintaks program) dimana akan menyebabkan kesalahan saat program dikompilasi (*compile error*) sehingga program tidak berhasil dikompilasi (lihat Gambar 2).

```
Herikas-MacBook-Pro:Source code RiRi$ javac SelamatDatang.java  
SelamatDatang.java:2: error: <identifier> expected  
    public statict void main(String[] args) {  
        ^  
SelamatDatang.java:2: error: invalid method declaration; return type required  
    public statict void main(String[] args) {  
        ^  
2 errors
```

Gambar 2 Compile error

## PRAKTIKUM 2

### VARIABEL DAN TIPE DATA

*Waktu : 45 menit*

#### 1.1 Tujuan

Tujuan dari Praktikum 2 adalah sebagai berikut:

1. Mampu membuat project Java menggunakan Eclipse.
2. Mampu mendeklarasikan variable dengan tipe data yang sesuai.
3. Mampu mengimplementasikan struktur keputusan/percabangan IF-ELSE
4. Mampu mengimplementasikan operator *conditional* sebagai bentuk lain dari IF-ELSE.

#### 1.2 Deskripsi

Eclipse adalah sebuah editor yang sudah terintegrasi dengan *compiler* dan JVM di dalamnya. Sehingga, kompilasi dan eksekusi program Java yang ditulis menggunakan Eclipse tidak perlu menggunakan *command prompt* atau terminal.

Variabel adalah wadah untuk menampung dan menyimpan sebuah nilai yang akan dibutuhkan dalam proses selanjutnya. Variabel di Java harus dideklarasikan dengan tipe data yang sesuai/*compatible* dengan nilai yang disimpannya.

Tipe data di Java dibagi 2 kategori, yaitu tipe data primitif dan tipe data reference. Tipe data primitif di Java ada 8, yaitu: byte, short, int, long, char, boolean, float, double. Tipe data reference adalah tipe data yang berisi reference/penunjuk ke alamat memori tempat object dibuat dan disimpan. Yang termasuk tipe data reference adalah String dan semua class yang dideklarasikan.

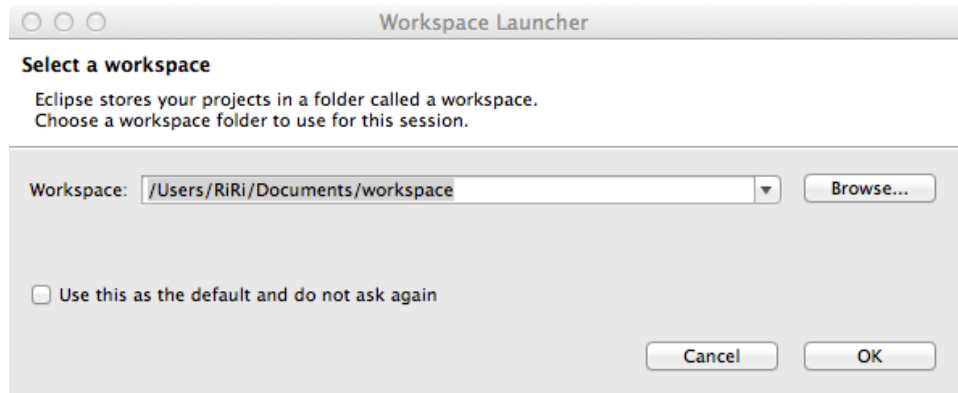
Struktur keputusan adalah struktur dasar pemrograman dimana instruksi dilakukan jika kondisi bernilai true. Sebaliknya, instruksi tidak dilakukan jika kondisi bernilai false. Di java, struktur keputusan dapat dibuat dengan menggunakan IF statement dan conditional operator.

## 1.3 Praktikum

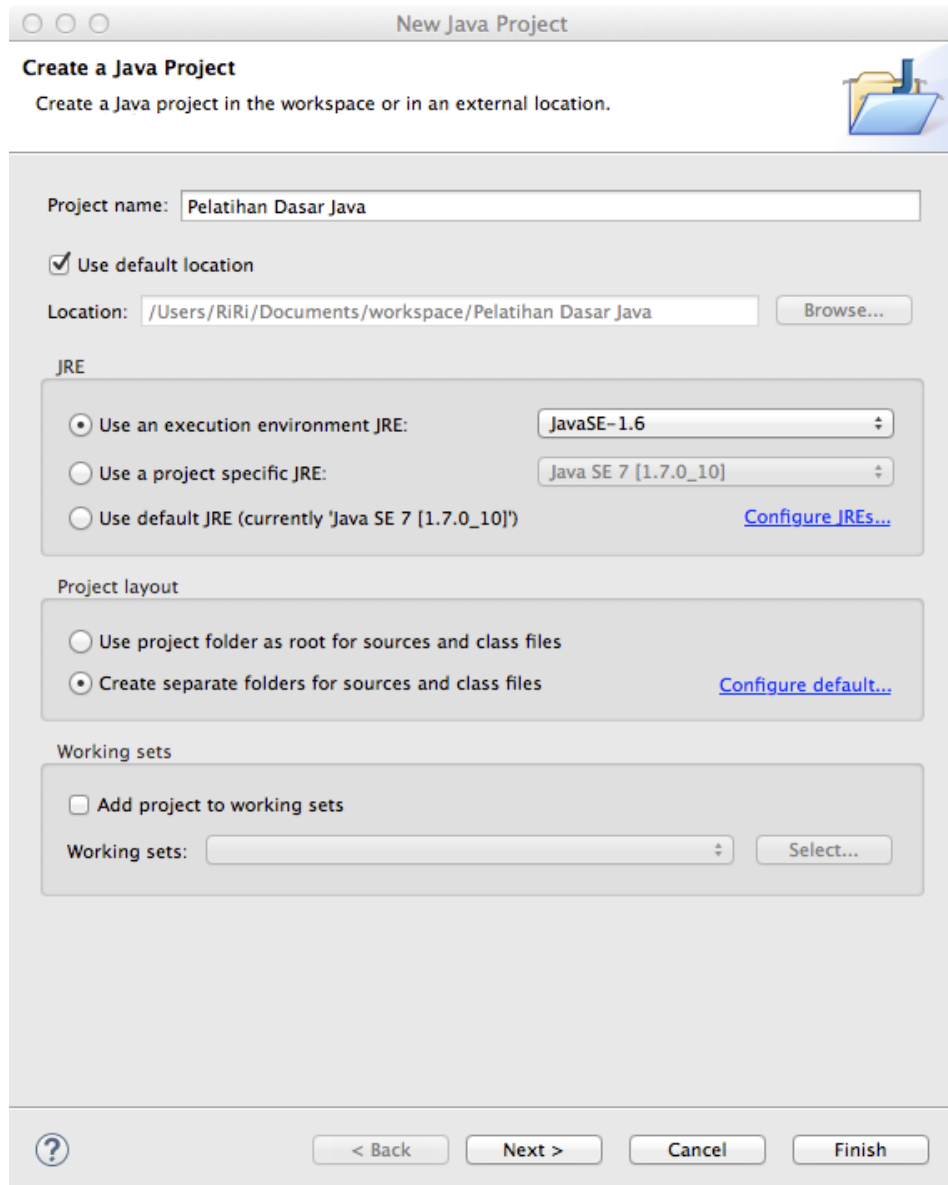
### 1.3.1 Membuat Project di Eclipse

Lakulah langkah-langkah berikut:

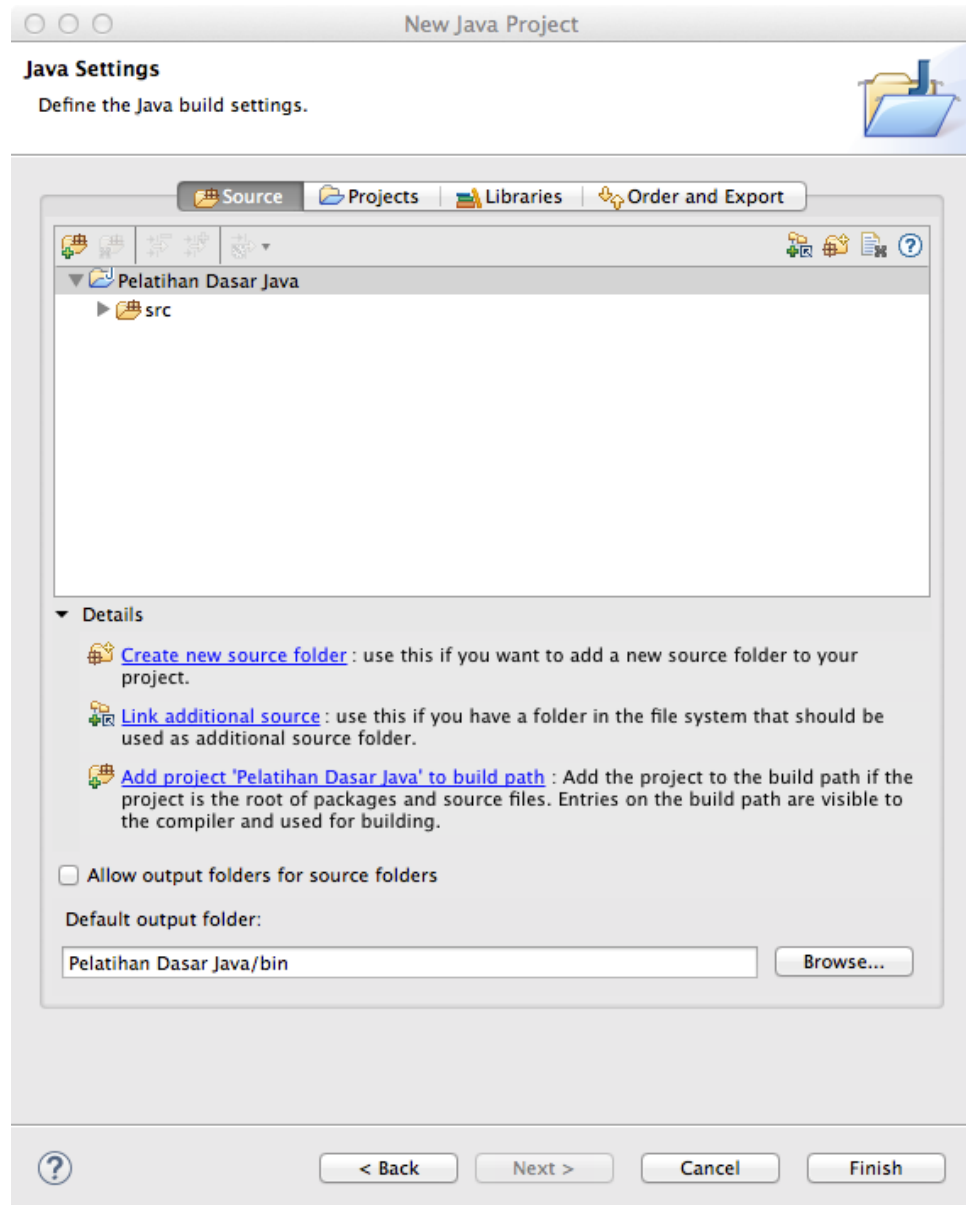
1. Buka Eclipse sampai muncul Workspace Launcher



2. Klik tombol Browse jika ingin mengubah workspace. Workspace adalah folder tempat dimana project Java yang dibuat dengan Eclipse akan disimpan. Lalu klik tombol OK.
3. Untuk membuat project baru di Eclipse, klik File > New > Java Project sampai muncul :



4. Ketikkan nama project. Nama project boleh mengandung spasi. Project akan menjadi folder di dalam workspace Anda.
5. Pilih JRE (Java Runtime Environment) sesuai kebutuhan Anda.
6. Klik Next



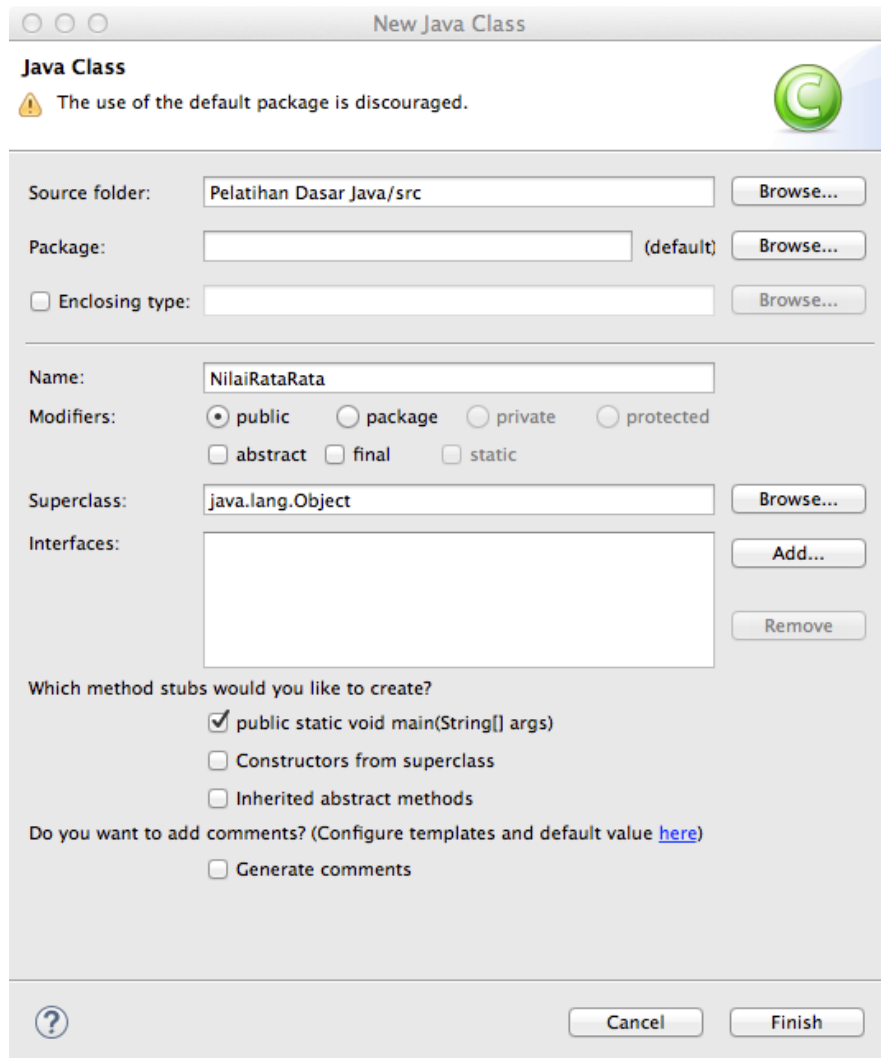
**Keterangan:** src adalah folder tempat source code java disimpan, dan bin adalah folder tempat *bytecode* disimpan.

7. Klik Finish

### 1.3.2 Mencari Nilai Rata-rata

Lakulah langkah-langkah berikut:

1. Pada project yang telah dibuat di Eclipse, klik kanan > New > Class sampai muncul:



2. Masukkan nama class, misalnya: NilaiRataRata
3. Beri tanda centang di check box “public static void main(String[] args)” sehingga Eclipse akan langsung membuatkan method main.
4. Klik Finish.
5. Tuliskan kode di dalam method main seperti di bawah ini:

```

public class NilaiRataRata {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int angka1 = 10;
        int angka2 = 15;
        int angka3 = 10;

        int ratarata = (angka1 + angka2 + angka3) / 3;

        System.out.println("Bilangan pertama : " + angka1);
        System.out.println("Bilangan kedua : " + angka2);
        System.out.println("Bilangan ketiga : " + angka3);
        System.out.println("Rata-rata : " + ratarata);
    }
}

```

6. Lalu, tambahkan komentar seperti kode di bawah ini:

```

/**
 * Class untuk menghitung nilai rata-rata
 * @author RiRi
 * @version 1 Mei 2015
 */
public class NilaiRataRata {

    /**
     * Method yang dijalankan pertama kali di program Java
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        // Deklarasi dan inisialisasi variabel
        int angka1 = 10;
        int angka2 = 15;
        int angka3 = 10;


        /* Deklarasi variabel ratarata
        sekaligus mengisinya dengan ekspresi */
        int ratarata = (angka1 + angka2 + angka3) / 3;


        // Mencetak 3 bilangan dan rata-ratanya
        System.out.println("Bilangan pertama : " + angka1);
        System.out.println("Bilangan kedua : " + angka2);
        System.out.println("Bilangan ketiga : " + angka3);
        System.out.println("Rata-rata : " + ratarata);
    }
}

```

7. Simpan program dengan shortcut ctrl + S atau klik File > Save.  
Menyimpan program di Eclipse sudah termasuk mengkompilasi program.



Bila ada error pada program, maka akan muncul tanda  di sebelah kiri baris kode program.

8. Jalankan program dengan cara : klik kanan pada class > Run As > Java Application atau klik icon  stau klik menu Run > Run.
9. Perhatikan keluaran program.
10. Apakah hasilnya tepat? Jika menurut Anda hasilnya kurang tepat, perbaiki program.

### 1.3.3 Mencari Nilai Maksimum dari Tiga Buah Bilangan

Lakukan langkah-langkah sebagai berikut:

1. Dengan cara yang sama seperti subbab 1.3.3, buatlah class bernama NilaiMaksimum di Eclipse. Lalu, ketikkan kode program berikut:

```
/**
 * Class untuk menentukan nilai maksimum dari 3 bilangan
 * @author RiRi
 * @version 1 Mei 2015
 */
public class NilaiMaksimum {

    /**
     * Method yang dijalankan pertama kali di program Java
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        // Deklarasi dan inisialisasi variabel
        int angka1 = 10;
        int angka2 = 15;
        int angka3 = 30;

        int max = 0;

        /* Menentukan nilai maksimum dengan IF-ELSE */
        if (angka1 > angka2)
            max = angka1;
        else max = angka2;

        if (angka3 > max)
            max = angka3;
    }
}
```

```

        // Mencetak 3 bilangan dan nilai maksimumnya
        System.out.println("Bilangan pertama : " + angka1);
        System.out.println("Bilangan kedua : " + angka2);
        System.out.println("Bilangan ketiga : " + angka3);
        System.out.println("Nilai maksimum : " + max);
    }
}

```

2. Jalankan program. Apa keluaran dari program?
3. Apa maksud dari potongan program di bawah ini?

```

/* Menentukan nilai maksimum dengan IF-ELSE */
if (angka1 > angka2)
    max = angka1;
else max = angka2;

if (angka3 > max)
    max = angka3;

```

4. Ganti potongan kode program IF-ELSE dengan potongan kode berikut:

```

/* Menentukan nilai maksimum dengan conditional operator */
max = angka1 > angka2 ? angka1 : angka2;
max = angka3 > max ? angka3 : max;

```

5. Jalankan program. Apa keluaran dari program?
6. Apa maksud dari potongan program di bawah ini?

```

/* Menentukan nilai maksimum dengan conditional operator */
max = angka1 > angka2 ? angka1 : angka2;
max = angka3 > max ? angka3 : max;

```

## **PRAKTIKUM 3**

### **INPUT DARI KEYBOARD**

*Waktu : 60 menit*

#### **3.1 Tujuan**

Tujuan dari Praktikum 3 adalah sebagai berikut:

1. Mampu membuat program Java yang meminta input dari pengguna
2. Mampu memahami API
3. Mampu menggunakan dan menjelaskan method-method yang ada di class String
4. Mampu menggunakan method-method yang ada di class StringTokenizer
5. Mampu menerapkan struktur keputusan
6. Mampu menerapkan struktur pengulangan

#### **3.2 Deskripsi**

##### **3.2.1 Class Scanner**

Scanner merupakan kelas dalam Java API yang dapat digunakan untuk menangkap input/masukan yang diberikan oleh user secara interaktif dalam *console-mode (command prompt)*. Untuk menggunakan class ini, anda harus melakukan proses import kelas Scanner pada bagian paling atas program file. Dalam Java, kelas Scanner berada dalam package `java.util`.

Berikut ini adalah contoh penggunaan kelas Scanner:

```
// membuat object bertipe Scanner
// kelas Scanner memiliki input berupa InputStream
Scanner s = new Scanner(System.in);

// menyimpan input yang diberikan user
String input = s.next();
```

### 3.2.2 Package dan Import Package

Package merupakan koleksi dari class-class dengan tugas spesifik. Beberapa contoh package dan class yang terdapat di dalamnya dapat dilihat pada Tabel 1.

**Tabel 1 Package dan Class**

Package	Class
java.util	<ul style="list-style-type: none"><li>• Scanner</li><li>• StringTokenizer</li></ul>
java.lang	<ul style="list-style-type: none"><li>• Object</li><li>• Math</li><li>• String</li></ul>
javax.swing	<ul style="list-style-type: none"><li>• JFrame</li><li>• JMenu</li><li>• JButton</li></ul>

Untuk dapat menggunakan class-class yang ada di library Java, program Java harus meng-import dengan menspesifikasikan nama package dan classnya di bagian teratas kode program (sebelum definisi class), kecuali class-class yang terdapat pada package java.lang, karena java.lang merupakan package default untuk seluruh program Java.

Contoh:

```
import javax.swing.JButton;

import java.lang.String; // tidak perlu dilakukan, karena java.lang
adalah default package.
```

### 3.2.3 API (Application Programming Interface)

Semua definisi *class* dan *method* dari *library* Java yang sudah dibuat oleh programmer terdahulu dan dapat di-reuse (digunakan kembali), didaftarkan di dokumentasi API (*Application Programming Interface*). Anda dapat mengakses API secara *online* di url: <http://docs.oracle.com/javase/7/docs/api/>. Anda dapat memiliki API secara *offline*, dengan mengunduh API ke *hard disk* computer Anda dari url: <http://www.oracle.com/technetwork/java/javase/documentation/java-se-7-doc-download-435117.html>.

### 3.2.4 Class String

Class String adalah salah satu class yang berada pada package java.lang. Tidak seperti package-package lainnya, package java.lang adalah package default dari seluruh program Java. Sehingga, untuk menggunakan class-class yang terdapat pada package java.lang, Anda tidak perlu meng-import-nya. Lengkapi Tabel 2 dengan memahami API class String.

**Tabel 2 Method-method di Class String**

No	Nama Method	Tugas Method
1	length()	
2	substring(int beginIndex)	
3	substring(int beginIndex, int endIndex)	
4	startsWith(String prefix)	
5	toUpperCase()	
6	toLowerCase()	
7	charAt(int index)	

### 3.2.5 Class StringTokenizer

Class StringTokenizer adalah class yang dapat digunakan untuk memecah sebuah string menjadi token berdasarkan *whitespace*. Class Tokenizer berada pada package java.util. Sehingga, untuk menggunakannya, Anda harus meng-import-nya yang dituliskan pada bagian atas program sebelum definisi class.

**Tabel 3 Method-method di Class StringTokenizer**

No	Nama Method	Tugas Method
1	hasMoreTokens()	memeriksa apakah masih ada token berikutnya
2	nextToken()	mengambil token selanjutnya

### 3.3 Praktikum

#### 3.3.1 Program HelloScanner

Buatlah class HelloScanner seperti di bawah ini untuk mencetak “Selamat Datang || [nama] || di Hotel Cempaka” dimana [nama] merupakan input dari pengguna.

```
import java.util.Scanner;

public class HelloScanner
{
    public static void main(String[] args)
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Masukkan nama anda\t: ");
        String input = s.next();
        System.out.println(haiSelamatDatang(input));
    }

    public static String haiSelamatDatang(String nama)
    {
        return "Selamat Datang ||\t" + nama + "\t|| di Hotel Cempaka";
    }
}
```

#### CONTOH KELUARAN

```
-----
Masukkan nama anda : Andi
Selamat Datang ||   Andi   || di Hotel Cempaka
```

#### 3.3.2 Program CobaString

Buatlah class CobaString seperti di bawah ini.

```
public class CobaString {
    public static void main(String[] args) {

        // Mendeklarasikan variabel kalimat dan menginisialisasi
        String kalimat = "Hai, apa kabar?";
        // mencetak isi variabel kalimat
        System.out.println("Kalimat: " + kalimat);

        // Mengambil substring "Hai" dari string kalimat
        // dan menyimpannya ke dalam variabel kataSapa.
        String kataSapa = kalimat.substring(0, 3);
        // Mencetak kata sapa
        System.out.println("Kata sapa dari kalimat di atas: " + kataSapa);

        // Mengambil substring berupa karakter terakhir dari string kalimat
        // dan menyimpannya ke dalam variabel tandaBaca.
        String tandaBaca = kalimat.substring(kalimat.length() - 1,
            kalimat.length());
        // Mencetak tanda baca
        System.out.println("Tanda baca dari kalimat di atas: " + tandaBaca);
    }
}
```

```

// Mengecek apakah kalimat diawali dengan kata "Hai".
// Jika iya, maka mencetak "Kalimat diawali dengan 'Hai'".
if (kalimat.startsWith("Hai")) {
    System.out.println("Kalimat diawali dengan 'Hai'.");
} else { // Jika tidak, maka mencetak
    // "Kalimat tidak diawali dengan 'Hai'"
    System.out.println("Kalimat tidak diawali dengan 'Hai'.");
}

// Mencetak karakter pada index 0 dari string kalimat
System.out.println("Karakter pertama dari kalimat di atas: "
    + kalimat.charAt(0));

// Mencetak karakter pada index terakhir dari string kalimat
System.out.println("Karakter terakhir dari kalimat di atas: "
    + kalimat.charAt(kalimat.length() - 1));

System.out.println(kalimat.toUpperCase()); // Mencetak "HAI, APA KABAR?"
System.out.println(kalimat.toLowerCase()); // Mencetak "hai, apa kabar?"
System.out.println(kalimat); // Mencetak isi dari variabel kalimat
}
}

```

### 3.3.3 Program CobaStringTokenizer

Buatlah class CobaStringTokenizer seperti di bawah ini.

```

import java.util.StringTokenizer;

class CobaStringTokenizer {
    public static void main(String[] args) {

        // Membuat object StringTokenizer dengan input
        // berupa string "Ini sebuah kalimat"
        StringTokenizer st = new StringTokenizer("Ini sebuah kalimat");

        // Memeriksa apakah masih ada token.
        while (st.hasMoreTokens()) { // Jika masih ada token,
            // Mencetak token per token dipisahkan dengan newline
            System.out.println(st.nextToken());
        }
    }
}

```

### 3.3.4 Latihan 1 – Sisipkan Dong!

Buatlah sebuah program Java yang dapat menerima 2 buah input dari keyboard:

1. Sebuah string dengan panjang genap, dan
2. Sebuah kata atau kalimat.

**Petunjuk:** Gunakan class Scanner dari package java.util.

Selanjutnya, program Anda akan mencetak string baru dimana kata atau kalimat yang dimasukkan sebagai input kedua, akan disisipkan ke string yang dimasukkan sebagai input pertama. **Petunjuk:** Gunakan method-method dari class String.

Untuk lebih jelasnya, perhatikan contoh jalannya program berikut ini.

*Contoh Keluaran 1:*

```
Masukkan string dengan panjang genap (diakhiri dengan enter): nasi
Masukkan kata (diakhiri dengan enter): makan
Hasil penyisipan kata: namakansi
```

**Keterangan:** kata yang dicetak tebal merupakan input dari pengguna.

### 3.3.5 Latihan 2 – Cari Warna

Buatlah sebuah program Java yang dapat menerima 2 buah input dari keyboard, yaitu:

1. Sebuah kalimat (diakhiri dengan enter), dan
2. Warna favorit pengguna (terdiri dari satu kata).

**Petunjuk:** Gunakan class Scanner dari package java.util.

Jika pada kalimat yang pengguna masukkan, terdapat warna favorit (yang juga dimasukkan pengguna), maka selanjutnya program harus menghitung berapa kali warna favorit disebutkan di dalam kalimat tersebut. **Petunjuk:** Gunakan method-method dari class String dan class StringTokenizer.

Perhatikan contoh jalannya program berikut ini.

*Contoh Keluaran 1:*

```
Masukkan kalimat (diakhiri dengan enter):
Merah kuning hijau di langit yang biru cerah, Tiga band besar Indonesia
membuat gebrakan dengan melakukan kolaborasi, mereka terdiri dari Hijau
Daun, The Potters dan Vegetos yang menjadi satu dengan nama 3 In 1.

Masukkan warna favoritmu (1 kata saja): hijau
```



Pada kalimat di atas, warna hijau disebutkan sebanyak 2 kali.

**Keterangan:** kalimat yang dicetak tebal merupakan input dari pengguna.

*Contoh Keluaran 2:*

Masukkan kalimat (diakhiri dengan enter):

**Merah kuning hijau di langit yang biru cerah, Tiga band besar Indonesia membuat gebrakan dengan melakukan kolaborasi, mereka terdiri dari Hijau Daun, The Potters dan Vegetos yang menjadi satu dengan nama 3 In 1.**

Masukkan warna favoritmu (1 kata saja): **merah**

Pada kalimat di atas, warna merah disebutkan sebanyak 1 kali.

**Keterangan:** kalimat yang dicetak tebal merupakan input dari pengguna.

### 3.3.6 Latihan 3 – Satukan!

Buatlah sebuah program Java yang dapat menerima 2 buah input dari keyboard, berupa 2 kata. **Petunjuk:** Gunakan class Scanner dari package java.util.

Program akan mencetak string baru yang merupakan gabungan kedua kata tersebut. Jika karakter terakhir dari kata pertama sama dengan karakter pertama dari kata kedua, maka hilangkan salah satu karakter tersebut. Contoh: program dengan input "bali" dan "ikan" akan mencetak string "balikan". Namun, program dengan input "lombok" dan "cabe" akan mencetak string "lombokcabe".

**Petunjuk:** Gunakan method-method dari class String dan operator konkatenasi untuk string.

Perhatikan contoh jalannya program berikut ini.

*Contoh Keluaran 1:*

Masukkan kata pertama (diakhiri dengan enter): **bali**  
Masukkan kata kedua (diakhiri dengan enter): **ikan**  
Hasil konkatenasi: balikan

**Keterangan:** kata yang dicetak tebal merupakan input dari pengguna.

*Contoh Keluaran 2:*

Masukkan kata pertama (diakhiri dengan enter): **lombok**  
Masukkan kata kedua (diakhiri dengan enter): **cabe**  
Hasil konkatenasi: lombokcabe

**Keterangan:** kata yang dicetak tebal merupakan input dari pengguna.

## PRAKTIKUM 4

### OBJECT, CLASS, METHOD, DAN CONSTRUCTOR

*Waktu : 60 menit*

#### 4.1 Tujuan

Tujuan dari Praktikum 4 adalah sebagai berikut:

1. Mampu membuat sebuah class,
2. Mampu membuat objects dari sebuah class,
3. Mampu mengimplementasikan constructor yang dapat menginisialisasi atribut dari object
4. Mampu mengidentifikasikan dan membuat *instance variable* dari object,
5. Mampu mengidentifikasikan dan membuat method yang dapat dijalankan oleh objects.
6. Mampu memahami konsep abstraksi
7. Mampu memahami konsep enkapsulasi

#### 4.2 Deskripsi

**Class** adalah cetakan, blueprint, template, prototype, tempat dari object, sedangkan **object** adalah isi/member atau realisasi dari kelas. Satu kelas dapat mempunyai object lebih dari satu atau banyak. Contoh sederhananya seperti berikut : satu cetakan bisa menghasilkan banyak kue. Cetakan adalah class, sedangkan kue adalah object.

**Method** adalah tugas spesifik yang dapat dilakukan oleh object. Misalnya object Burung memiliki method terbang, berkicau, tidur.

Object juga dapat memiliki **atribut/properti/instance variable**. Misalnya object Burung memiliki atribut nama, jenis, sayap.

**Constructor** adalah method khusus tanpa return type, yang digunakan untuk menginisialisasi atribut object.

**Abstraksi** merupakan pemeriksaan yang selektif terhadap aspek-aspek tertentu pada suatu persoalan. Abstraksi berarti memfokuskan pada aspek-aspek esensi,

inheren dan mengabaikan properti-properti yang merupakan akibatnya. Pada pendekatan orientasi object, abstraksi yang digunakan berfokus pada apa object itu. Ini dilakukan sebelum memutuskan bagaimana cara object itu diimplementasikan. Penggunaan abstraksi melindungi kebebasan untuk membuat keputusan-keputusan pokok penyelesaian persoalan dengan sejauh mungkin menghindari rincian-rincian yang terlalu dini.

**Enkapsulasi** adalah menyembunyikan implementasi program. Setiap class dan member class dapat diberikan hak akses atau *access modifier*. Di Java, ada 4 jenis *access modifier*, yaitu:

1. Private : class atau member class hanya dapat diakses (digunakan dan diubah) oleh class itu sendiri
2. Default (tidak ditulis di programnya) : class atau member class hanya dapat diakses (digunakan dan diubah) oleh class-class di dalam package yang sama.
3. Protected : class atau member class hanya dapat diakses (digunakan dan diubah) oleh class-class di dalam package yang sama atau class-class yang merupakan *subclass*-nya.
4. Public : class atau member class dapat diakses (digunakan dan diubah) oleh semua class di luar class tersebut.

**Overloading** adalah penamaan yang sama untuk constructor/method di dalam satu class.

## 4.3 Praktikum

### 4.3.1 Class BujurSangkar

Buatlah sebuah class BujurSangkar (**tanpa method main**) yang dapat digunakan untuk membuat object-object BujurSangkar yang memiliki atribut/properti sisi. Class ini menyediakan method-method sebagai berikut :

1. setSisi → menge-set sisi dari bujur sangkar
2. getSisi → mengembalikan sisi dari bujur sangkar

3. `hitungKeliling` → menghitung dan mengembalikan keliling dari bujur sangkar
4. `hitungLuas` → menghitung dan mengembalikan luas dari bujur sangkar
5. `toString` → memformat dan mengembalikan String output dari program berupa sisi, luas, dan keliling bujur sangkar

Selain method, class ini juga menyediakan 2 buah constructor, yaitu:

1. **No-argument constructor** → constructor akan menginisialisasi nilai sisi sama dengan 1.0 pada argumen dari method `setSisi`
2. **Constructor dengan 1 buah argumen sisi** → constructor memiliki 1 argumen, yaitu sisi dan digunakan untuk menginisialisasi sisi dari bujur sangkar melalui method `setSisi`

Class `BujurSangkar` dapat dilihat pada Tabel 4.

Tabel 4 Class BujurSangkar

```
public class BujurSangkar {

    private double sisi;

    public BujurSangkar() {
        this (1.0);
    }

    public BujurSangkar(double sisi) {
        this.sisi = sisi;
    }

    public double getSisi() {
        return sisi;
    }

    public void setSisi(double sisi) {
        this.sisi = sisi < 1 ? 1 : sisi;
    }

    public double hitungLuas() {
        return sisi * sisi;
    }

    public double hitungKeliling() {
        return 4 * sisi;
    }

    public double hitungLuas(double s) {
        return s * s;
    }

    public double hitungKeliling(double s) {
        return 4 * s;
    }

    public String toString() {
        return "Sisi: " + getSisi()
            + "\nLuas: " + hitungLuas()
            + "\nKeliling: " + hitungKeliling();
    }
}
```

Apakah class ini dapat dijalankan?

Class BujurSangkar tidak dapat dijalankan karena tidak berisi method main. Method main adalah method yang harus ada di dalam sebuah program Java. Program Java dapat terdiri dari lebih dari satu class.

#### 4.3.2 Class TesBujurSangkar

Langkah selanjutnya adalah:

1. Untuk menjalankan class BujurSangkar tersebut, Anda dapat membuat class TesBujurSangkar dimana di dalamnya berisi method main.
2. Dalam method main, buat object Scanner yang dapat menerima input dari keyboard dengan melewati object System.in pada argumen constructor Scanner. Jangan lupa untuk menuliskan import statement di bagian atas program agar dapat menggunakan class Scanner.

```
1 import java.util.Scanner;
2
3 public class TesBujurSangkar
4 {
5     public static void main(String[] args)
6     {
7         Scanner input = new Scanner(System.in);
8     }
9 }
```

3. Dalam method main, buat object BujurSangkar dengan menggunakan no-argument constructor.

```
1 import java.util.Scanner;
2
3 public class TesBujurSangkar
4 {
5     public static void main(String[] args)
6     {
7         Scanner input = new Scanner(System.in);
8         BujurSangkar bs = new BujurSangkar();
9     }
10 }
```

4. Tampilkan ke user program, tulisan sebagai berikut:

Selamat datang di Program Bujur Sangkar Masukkan input berupa sisi bujur sangkar (dalam cm) :
--

```
1 import java.util.Scanner;
2
3 public class TesBujurSangkar
4 {
5     public static void main(String[] args)
6     {
7         Scanner masuk = new Scanner(System.in);
8         BujurSangkar bs = new BujurSangkar();
9         System.out.println("Selamat datang di "
10             + "Program Bujur Sangkar");
11         System.out.print("Masukkan input sisi bujur"
12             + " sangkar (dalam cm) : ");
13     }
14 }
```

5. Tangkap input dari keyboard dengan cara object Scanner menjalankan method nextDouble dan simpan input tersebut dalam variabel masukan yang bertipe double.

```
1 import java.util.Scanner;
2
3 public class TesBujurSangkar
4 {
5     public static void main(String[] args)
6     {
7         Scanner masuk = new Scanner(System.in);
8         BujurSangkar bs = new BujurSangkar();
9         System.out.println("Selamat datang di "
10             + "Program Bujur Sangkar");
11         System.out.print("Masukkan input sisi bujur"
12             + " sangkar (dalam cm) : ");
13         double masukan = masuk.nextDouble();
14     }
15 }
```

6. Set sisi bujur sangkar dengan memanggil method setSisi dengan argumen dari method tersebut adalah variabel masukan.

```

1  import java.util.Scanner;
2
3  public class TesBujurSangkar
4  {
5      public static void main(String[] args)
6      {
7          Scanner masuk = new Scanner(System.in);
8          BujurSangkar bs = new BujurSangkar();
9          System.out.println("Selamat datang di "
10             + "Program Bujur Sangkar");
11          System.out.print("Masukkan input sisi bujur"
12             + " sangkar (dalam cm) : ");
13          double masukan = masuk.nextDouble();
14          bs.setSisi(masukan);
15      }
16  }

```

CATATAN: Gunakan object BujurSangkar yaitu variabel object bs yang sudah dibuat di baris 8, untuk dapat memanggil/menjalankan method setSisi. Argumen dari method setSisi diisi dengan variabel masukan (konsep pass-by-value).

7. Tampilkan seluruh String yang sudah dideklarasikan pada method toString dari class BujurSangkar. Dalam hal ini, cukup dengan cara meng-print variabel object bs.

```

1  import java.util.Scanner;
2
3  public class TesBujurSangkar
4  {
5      public static void main(String[] args)
6      {
7          Scanner masuk = new Scanner(System.in);
8          BujurSangkar bs = new BujurSangkar();
9          System.out.println("Selamat datang di "
10             + "Program Bujur Sangkar");
11          System.out.print("Masukkan input sisi bujur"
12             + " sangkar (dalam cm) : ");
13          double masukan = masuk.nextDouble();
14          bs.setSisi(masukan);
15          System.out.println(bs);
16      }
17  }

```

8. Kompilasi dan jalankan program TesBujurSangkar. Apa keluaran programnya?



### 4.3.3 Latihan – Persegi Panjang

Buatlah sebuah class PersegiPanjang (tanpa method main) yang dapat digunakan untuk membuat object-object PersegiPanjang yang memiliki atribut panjang dan lebar.

Dalam class PersegiPanjang dapat dilakukan hal-hal sebagai berikut :

1. Menge-set panjang dari persegi panjang
2. Menge-set lebar dari persegi panjang
3. Mengembalikan / mendapatkan panjang dari persegi panjang
4. Mengembalikan / mendapatkan lebar dari persegi panjang
5. Menghitung dan mengembalikan keliling dari persegi panjang
6. Menghitung dan mengembalikan luas dari persegi panjang
7. Memformat dan mengembalikan String output dari program berupa panjang, lebar, luas, dan keliling persegi panjang

Selain method, class PersegiPanjang juga menyediakan 2 buah constructor:

1. **No-argument constructor** → menginisialisasi nilai panjang dan lebar sama dengan 1.0 dan 1.0.
2. **Constructor dengan 2 buah argumen (panjang & lebar)** → memiliki 2 argumen, yaitu panjang dan lebar dan digunakan untuk menginisialisasi panjang dan lebar dari persegi panjang melalui method-method.

Sediakan class TesPersegiPanjang yang berisi method main dimana terdapat instansiasi object PersegiPanjang dan merupakan antar muka program dengan user.

Contoh output dari Program sebagai berikut (boleh berbeda):

```
Selamat datang di Program Luas dan Keliling Persegi Panjang
Masukkan panjang (dalam cm) : 8
Masukkan lebar (dalam cm): 7
Luas = 56 cm persegi
Keliling = 30 cm
```

## PRAKTIKUM 5

### INHERITANCE (BAGIAN PERTAMA)

*Waktu : 120 menit*

#### 5.1 Tujuan

Tujuan dari Praktikum 5 adalah peserta mampu memahami konsep inheritance dalam Java.

#### 5.2 Deskripsi

**Inheritance (pewarisan)** adalah konsep dalam pemrograman berorientasi object dimana sebuah class dapat mewarisi atribut dan method dari class lain. Class yang mewarisi disebut sebagai subclass, sedangkan class yang diwarisi disebut sebagai superclass. Saat deklarasi subclass, gunakan *keyword* `extends` setelah deklarasi class, lalu diikuti dengan nama superclass-nya.

Untuk memanggil constructor dari superclass-nya, suatu subclass harus memanggilnya dengan keyword `super` yang diletakkan pada statement pertama di constructor subclass.

#### 5.3 Praktikum

Anda akan membuat class `Person`, `Student`, `Teacher`, dan `InternationalStudent` di dalam **package** `mypeopleexample`. Class `Student` dan class `Teacher` merupakan subclass dari class `Person`. Class `InternationalStudent` merupakan subclass dari class `Student`.

##### 5.3.1 Class Person

Langkah-langkah yang harus dilakukan:

1. Buatlah class `Person` (tanpa method `main`)
2. Buatlah atribut/property/instance variable di class `Person` (deklarasikan dengan tipe yang sesuai):
  - a. `name` → untuk menyimpan nama orang
  - b. `address` → untuk menyimpan alamat orang
3. Buatlah method getter dan setter untuk setiap atribut `Person`.

**Tabel 5 Class Person**

```
package mypeopleexample;

public class Person {
    private String name;
    private String address;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }
}
```

### 5.3.2 Class Student

Langkah-langkah yang harus dilakukan:

1. Buatlah class Student (tanpa method main) yang merupakan subclass dari class Person
2. Buatlah atribut/property/instance variable di class Student (deklarasikan dengan tipe yang sesuai):
  - a. school → untuk menyimpan sekolah siswa
  - b. grade → untuk menyimpan nilai siswa
3. Buatlah method getter dan setter untuk setiap atribut Student.

**Tabel 6 Class Student**

```
package mypeopleexample;

public class Student extends Person {
    private String school;
```

```

        private double grade;

        public String getSchool() {
            return school;
        }

        public void setSchool(String school) {
            this.school = school;
        }

        public double getGrade() {
            return grade;
        }

        public void setGrade(double grade) {
            this.grade = grade;
        }
    }

```

### 5.3.3 Class InternationalStudent

Langkah-langkah yang harus dilakukan:

1. Buatlah class InternationalStudent (tanpa method main) yang merupakan subclass dari class Student
2. Buatlah atribut/property/instance variable di class InternationalStudent (deklarasikan dengan tipe yang sesuai):
  - a. country → untuk menyimpan asal negara
3. Buatlah method getter dan setter untuk setiap atribut InternationalStudent.

**Tabel 7 Class InternationalStudent**

```

package mypeopleexample;

public class InternationalStudent extends Student {
    /**
     * Creates a new instance of InternationalStudent
     */
    public InternationalStudent() {
    }

    private String country;

    public String getCountry() {
    }
}

```

```

        return country;
    }

    public void setCountry(String country) {
        this.country = country;
    }
}

```

#### 5.3.4 Class Teacher

Langkah-langkah yang harus dilakukan:

1. Buatlah class Teacher (tanpa method main) yang merupakan subclass dari class Person
2. Buatlah atribut/property/instance variable di class Teacher (deklarasikan dengan tipe yang sesuai):
  - a. subject → untuk menyimpan subject yang diampu
3. Buatlah method getter dan setter untuk setiap atribut Teacher.

**Tabel 8 Class Teacher**

```

package mypeopleexample;

public class Teacher extends Person {
    private String subject;

    public String getSubject() {
        return subject;
    }

    public void setSubject(String subject) {
        this.subject = subject;
    }
}

```

#### 5.3.5 Class MainTester

Langkah-langkah yang harus dilakukan:

1. Buatlah class MainTester (berisi method main)
2. Di dalam method main, ketikkan kode seperti di Tabel 9.

**Tabel 9 Class MainTester**

```
package mypeopleexample;

public class MainTester {
    public static void main(String[] args) {
        // Create object instances and invoke methods.
        // Note that you can use methods defined
        // in a parent class for object
        // instances of the child class.
        Person person1 = new Person();
        person1.setName("Tom Jones");
        Student student1 = new Student();
        student1.setName("CCR");
        student1.setSchool("Lexington High");
        InternationalStudent internationalStudent1
            = new InternationalStudent();
        internationalStudent1.setName("Bill Clinton");
        internationalStudent1.setSchool("Lexington High");
        internationalStudent1.setCountry("Korea");
        Teacher teacher1 = new Teacher();
        teacher1.setName("Beatles");
        teacher1.setSubject("History");

        // Display name of object instances
        // using the getName() method
        // defined in the Person class.
        System.out.println("Displaying names of all object
```

```

instances");

    System.out.println(" person1.getName() = "
        + person1.getName());

    System.out.println(" student1.getName() = "
        + student1.getName());

    System.out.println(" internationalStudent1.getName() = "
        + internationalStudent1.getName());

    System.out.println(" teacher1.getName() = "
        + teacher1.getName());

}

}

```

Jalankan program MainTester dan amati hasilnya!

**Tabel 10 Tambahan kode di akhir method main dari class MainTester**

```

// Display the class hierarchy of the InternationalStudent
class

// through getSuperclass() method of Class class.
Class class1 = internationalStudent1.getClass();

System.out.println("Displaying class hierarchy of
InternationalStudent Class...");

while (class1.getSuperclass() != null) {

    String child = class1.getName();

    String parent = class1.getSuperclass().getName();

    System.out.println(" " + child
        + " class is a child class of "

```

```
        + parent);  
  
        class1 = class1.getSuperclass();  
  
    }
```

Jalankan program MainTester kembali dan amati hasilnya!

### 5.3.6 Memanggil Rantai Constructor

Tambahkan constructor di class Person, Student, Teacher, dan InternationalStudent:

1. Constructor di class Person (Tabel 11)
2. Constructor di class Student (Tabel 12)
3. Constructor di class InternationalStudent (Tabel 13)
4. Constructor di class Teacher (Tabel 14)

**Tabel 11 Constructor Person()**

```
public Person() {  
  
    System.out.println("Person: constructor is called");  
  
}
```

**Tabel 12 Constructor Student()**

```
public Student() {  
  
    System.out.println("Student: constructor is called");  
  
}
```

**Tabel 13 Constructor InternationalStudent**

```
public InternationalStudent() {  
  
    System.out.println("InternationalStudent: constructor is  
called");  
  
}
```



**Tabel 14 Constructor Teacher()**

```
public Teacher() {  
    System.out.println("Teacher: constructor is called");  
}
```

Setelah ditambahkan keempat constructor pada class yang sesuai, modifikasi kembali class MainTester seperti pada Tabek 15.

**Tabel 15 Modifikasi Class MainTester**

```
package mypeopleexample;  
  
public class MainTester {  
    public static void main(String[] args) {  
        // Create an object instance of  
        // InternationalStudent class.  
        System.out.println("---- About to create an object  
instance of InternationalStudent class...");  
        InternationalStudent internationalStudent1 = new  
InternationalStudent();  
        // Create an object instance of // Teacher class.  
        System.out.println("---- About to create an object  
instance of Teacher class...");  
        Teacher teacher1 = new Teacher();  
    }  
}
```

Jalankan program MainTester kembali dan amati hasilnya!

### 5.3.7 Menambahkan super()

Tambahkan constructor di class Person, Student, dan InternationalStudent:

1. Constructor di class Person (Tabel 16)
2. Constructor di class Student (Tabel 17)
3. Constructor di class InternationalStudent (Tabel 18)

**Tabel 16 Constructor Person(String name)**

```
public Person(String name) {  
    this.name = name;  
    System.out.println("Person: constructor 2 is called");  
}
```

**Tabel 17 Constructor Student(String name, String school, double grade)**

```
public Student(String name, String school, double grade) {  
    super(name);  
    this.school = school;  
    this.grade = grade;  
    System.out.println("Student: constructor 2 is called");  
}
```

**Tabel 18 Constructor InternationalStudent(String name, String school,  
double grade, String country)**

```
public InternationalStudent(String name, String school,  
double grade, String country) {  
    super(name, school, grade);  
    this.country = country;  
    System.out.println("InternationalStudent: constructor 2  
is called");  
}
```

Selanjutnya, modifikasi class MainTester seperti pada Tabel 19.

**Tabel 19 Modifikasi Class MainTester**

```
package mypeopleexample;

public class MainTester {
    public static void main(String[] args) {
        // Create an object instance of
        // InternationalStudent class.
        System.out.println("---- About to create an object"
            + " instance of InternationalStudent class...");
        InternationalStudent internationalStudent1
            = new InternationalStudent("Sang Shin", // Name
                "1 Dreamland", // Address
                4.5, // Grade
                "Korea"); // Country
        System.out.println("internationalStudent1.getName() = "
            + internationalStudent1.getName());
        System.out.println("internationalStudent1.getAddress() = "
            + internationalStudent1.getAddress());
        System.out.println("internationalStudent1.getGrade() = "
            + internationalStudent1.getGrade());
        System.out.println("internationalStudent1.getCountry() = "
            + internationalStudent1.getCountry());
    }
}
```

Jalankan program MainTester kembali dan amati hasilnya!

Untuk informasi lebih lanjut, silakan menghubungi:

**Program Studi Teknik Informatika  
Fakultas Teknologi Informasi  
Universitas YARSI**

1. Herika Hayurani, M.Kom  
E-mail : [herika.hayurani@yarsi.ac.id](mailto:herika.hayurani@yarsi.ac.id)
2. Nova Eka Diana, S.Kom, M.Eng  
E-mail : [nova.diana@yarsi.ac.id](mailto:nova.diana@yarsi.ac.id)
3. Nurmaya, S.Kom, M.Eng  
E-mail : [nurmaya@yarsi.ac.id](mailto:nurmaya@yarsi.ac.id)