JOBSHEET

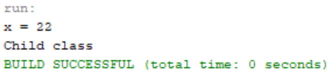
MODUL 7 (POLYMORPHISM)

<table>
<tr><th>No</th><th>Kegiatan</th><th>Contoh</th><th>Latihan</th></tr>
<tr><td>1</td><td>Pengenalan Virtual method Invocation</td><td>

```
package jobsheet1p7;

class parent{
    int x = 22;

    public void info(){
        System.out.println("Parent class");
    }
}

class child extends parent{
    int x = 11;


    public void info(){
        System.out.println("Child class");
    }
}

public class Jobsheet1P7 {
    public static void main(String[] args)
{
        parent obj = new child();

        System.out.println("x = "+obj.x);

        obj.info();
    }

}
```

</td><td>

Buatlah program bebas menggunakan virtual method invocation



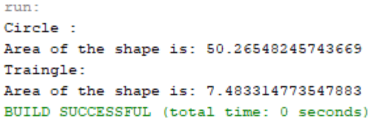
```
run:
x = 22
Child class
BUILD SUCCESSFUL (total time: 0 seconds)
```

</td></tr>
<tr><td>2</td><td>Pengenalan Heterogeneous Collection</td><td>

```
package jobsheet2p7;


import java.util.ArrayList;

import java.util.List;


abstract class shape{

    protected double area;

    public abstract void calculateArea();

    public void displayArea(){
```

</td><td>

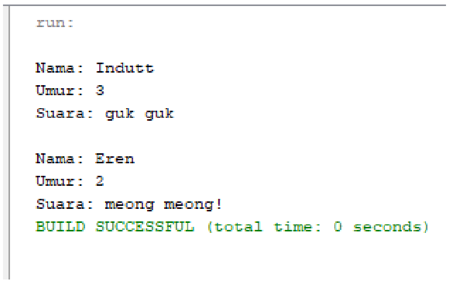
Buatlah program bebas atau modifikasi program disamping menggunakan Heterogeneous Collection



```
run:
Circle :
Area of the shape is: 50.26548245743669
Traingle:
Area of the shape is: 7.483314773547883
BUILD SUCCESSFUL (total time: 0 seconds)
```

</td></tr>
</table>

```
        System.out.println("Area of the shape is: "+area);
    }
}


class circle extends shape{
    private double radius;
    public circle(double newRadius) {
        this.radius = newRadius;
    }


    @Override
    public void calculateArea(){
        area = Math.PI * Math.pow(radius,2);
    }
}
// public class rectangle extends shape {
//     private double width;
//     private double height;


//     public rectangle(double width, double height){
//         this.width = width;
//         this.height = height;
//     }


//     @Override
//     public void calculateArea(){
//         area = width * height;
//     }
// }


class triangle extends shape{
    private double a;
    private double b;
    private double c;

    public triangle(double a, double b, double c){
        this.a = a;
        this.b = b;
        this.c = c;
    }

    @Override
    public void calculateArea(){
        double s = (a+b+c)/2;

        area = Math.sqrt(s*(s-a)*(s-b)*(s-c));
    }
}



public class Jobsheet2P7 {

    public static void main(String[] args) {
        List<shape> shapes = new ArrayList<>();
        shapes.add(new circle(4.0));
        // shapes.add(new rectangle(8, 18.0));
        shapes.add(new triangle(3.0, 5.0, 6.0));

        for(shape shape: shapes){
            if(shape instanceof circle){
```

| | | | |
|---|---|---|---|
| | | System.out.println("Circle : ");<br>circle circle = (circle) shape;<br>circle.calculateArea();<br>circle.displayArea();<br>// } else if (shape instanceof rectangle){<br>// System.out.println("Rectangle : ");<br>// rectangle rectangle = (rectangle) shape;<br>// rectangle.calculateArea();<br>// rectangle.displayArea();<br>} else if (shape instanceof triangle){<br>System.out.println("Traingle: ");<br>triangle triangle = (triangle) shape;<br>triangle.calculateArea();<br>triangle.displayArea();<br>}<br>}<br>}<br>} | |
| 3 | Pengenalan Polymorphic Argument | package jobsheet3p7;<br>public class Jobsheet3P7 {<br>public static void main(String[] args) {<br>Animal[] binatang = new Animal[2];<br>binatang[0] = new Anjing("Indutt", 3);<br>binatang[1] = new Kucing("Eren", 2);<br>for (Animal hewan : binatang) { | Buatlah program bebas atau modifikasi program disamping menggunakan polymorphic argument<br><br>run:<br>Nama: Indutt<br>Umur: 3<br>Suara: guk guk<br>Nama: Eren<br>Umur: 2<br>Suara: meong meong!<br>BUILD SUCCESSFUL (total time: 0 seconds) |

```
            System.out.println("\nNama: " +
hewan.getNama());
            System.out.println("Umur: " +
hewan.getUmur());
            System.out.println("Suara: " +
hewan.makeSound());
        }
    }
}

abstract class Animal {
    private final String nama;
    private int umur;

    Animal(String newName, int newAge) {
        this.nama = newName;
        this.umur = newAge;
    }

    String getNama() {
        return nama;
    }

    int getUmur() {
        return umur;
    }

    abstract String makeSound();
}

class Anjing extends Animal {

    Anjing(String name, int age) {
        super(name, age);
```

<table>
<tr><td></td><td></td><td><pre>
    }

    @Override
    String makeSound() {
        return "guk guk";
    }
}

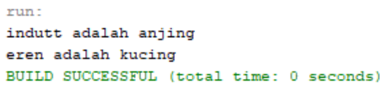
class Kucing extends Animal {

    Kucing(String name, int age) {
        super(name, age);
    }

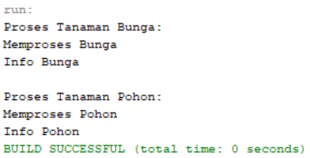
    @Override
    String makeSound() {
        return "meong meong!";
    }
}
</pre></td><td></td></tr>
<tr><td>4</td><td>Pengenalan Operator instanceof</td><td><pre>
Package jobsheet4p7;

public class Jobsheet4P7 {
    public static void main(String[] args) {
        Hewan[] hewan = new Hewan[2];
        hewan[0] = new anjing("indutt", 3);
        hewan[1] = new kucing("eren", 2);

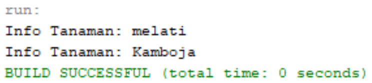
        for(Hewan binatang : hewan){
            if (binatang instanceof anjing){

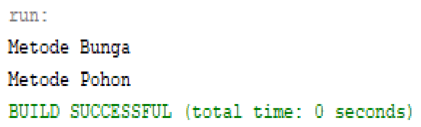
System.out.println(binatang.getName()+"
adalah anjing");
            } else if (binatang instanceof
kucing){
</pre></td><td>Buatlah program bebas atau modifikasi program disamping menggunakan operator instanceof

<pre>
run:
indutt adalah anjing
eren adalah kucing
BUILD SUCCESSFUL (total time: 0 seconds)
</pre></td></tr>
</table>

```
System.out.println(binatang.getName()+"
adalah kucing");
            }
        }
    }
}



abstract class Hewan{
    private final String name;
    private int age;

    Hewan(String nwName, int nwAge){
        this.name = nwName;
        this.age = nwAge;
    }

    String getName(){
        return name;
    }

    int getAge(){
        return age;
    }
    abstract String makeSound();
}

class anjing extends Hewan{

    anjing(String name, int age){
        super(name, age);
    }
```

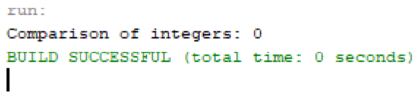
|  |  |  |  |
|---|---|---|---|
|  |  | @Override<br>    String makeSound(){<br>        return "Wank wank!";<br>    }<br>}<br><br>class kucing extends Hewan{<br><br>    kucing(String name, int age){<br>        super(name, age);<br>    }<br><br>    @Override<br>    String makeSound(){<br>        return "Maong maong!";<br>    }<br>} |  |
| 5 | Buatlah program bebas atau modifikasi program disamping menggunakan Object Casting | package bungaa;<br><br>class Tanaman {<br>    void tampilkanInfo()  {<br>        System.out.println("Info Tanaman Umum");<br>    }<br>}<br><br>class Bunga extends Tanaman {<br>    void tampilkanInfo()  {<br>        System.out.println("Info Bunga");<br>    }<br>} | <br>run:<br>Proses Tanaman Bunga:<br>Memproses Bunga<br>Info Bunga<br><br>Proses Tanaman Pohon:<br>Memproses Pohon<br>Info Pohon<br>BUILD SUCCESSFUL (total time: 0 seconds) |

```
class Pohon extends Tanaman {
    void tampilkanInfo() {
        System.out.println("Info Pohon");
    }
}

public class Bungaa {
    public static void prosesTanaman(Object tanaman) {
        if (tanaman instanceof Bunga) {
            Bunga bunga = (Bunga) tanaman;

            System.out.println("Memproses Bunga");
            bunga.tampilkanInfo();
        } else if (tanaman instanceof Pohon) {
            Pohon pohon = (Pohon) tanaman;

            System.out.println("Memproses Pohon");
            pohon.tampilkanInfo();
        } else {

            System.out.println("Memproses Tanaman Lainnya");
        }
    }

    public static void main(String[] args) {
        Tanaman bunga = new Bunga();
        Tanaman pohon = new Pohon();

        System.out.println("Proses Tanaman Bunga:");
```

| | | | |
|---|---|---|---|
| | | prosesTanaman(bunga);<br><br>System.out.println("\nProses Tanaman Pohon:");<br>prosesTanaman(pohon);<br>}<br>} | |
| 6 | Buatlah program bebas dengan Up Casting | package bungaa;<br>class Tanaman {<br>protected String jenis;<br>public Tanaman(String jenis) {<br>this.jenis = jenis;<br>}<br><br>@Override<br>public String toString() {<br>return "Info Tanaman: " + jenis;<br>}<br>}<br><br>class Bunga extends Tanaman {<br>public Bunga(String jenis) {<br>super(jenis);<br>}<br><br>public String methodeBunga() {<br>return "Metode Bunga";<br>}<br>}<br><br>class Pohon extends Tanaman {<br>public Pohon(String jenis) {<br>super(jenis); | <br>run:<br>Info Tanaman: melati<br>Info Tanaman: Kamboja<br>BUILD SUCCESSFUL (total time: 0 seconds) |

| | | | |
|---|---|---|---|
| | |     }<br><br>    public String methodePohon() {<br>        return "Metode Pohon";<br>    }<br>}<br><br>public class Bungaa {<br>    public static void main(String[] args) {<br>        Bunga melati = new Bunga("melati");<br>        Pohon Kamboja = new Pohon("Kamboja");<br><br>        Tanaman tanaman1 = (Tanaman) melati;<br>        Tanaman tanaman2 = (Tanaman) Kamboja;<br><br>System.out.println(tanaman1.toString());<br><br>System.out.println(tanaman2.toString());<br>    }<br>} | |
| 7 | Buatlah program bebas dengan Down Casting | package bungaa;<br><br>class Tanaman {<br>    protected String jenis;<br><br>    public Tanaman(String jenis) {<br>        this.jenis = jenis;<br>    }<br><br><br>    public String toString() {<br>        return "Info Tanaman: " + jenis; | <br>run:<br>Metode Bunga<br>Metode Pohon<br>BUILD SUCCESSFUL (total time: 0 seconds) |

```
    }
}

class Bunga extends Tanaman {
    public Bunga(String jenis) {
        super(jenis);
    }

    public String methodeBunga() {
        return "Metode Bunga";
    }
}

class Pohon extends Tanaman {
    public Pohon(String jenis) {
        super(jenis);
    }

    public String methodePohon() {
        return "Metode Pohon";
    }
}

public class Bungaa {
    public static void main(String[] args) {
        Tanaman tanaman = new
Bunga("melati");
        if (tanaman instanceof Bunga) {
            Bunga bunga = (Bunga) tanaman;

System.out.println(bunga.methodeBunga());
        }
```

| | | | |
|---|---|---|---|
| | | // Membuat objek Tanaman<br>Tanaman tanaman2 = new Pohon("Kamboja");<br>if (tanaman2 instanceof Pohon) {<br>    Pohon pohon = (Pohon) tanaman2;<br>System.out.println(pohon.methodePohon());<br>}<br>}<br>} | |
| 8 | Buatlah program untuk membandingkan kedua nilai menggunakan polimorfis statis | package jobsheetp7;<br>public class JobsheetP7 {<br>    public static int compare(int a, int b) {<br>        if (a > b) {<br>            return 1;<br>        } else if (a < b) {<br>            return 0;<br>        } else {<br>            return 0;<br>        }<br>    }<br>    public static void main(String[] args) {<br>        int intResult = compare(100, 300);<br>        System.out.println("Comparison of integers: " + intResult);<br>    } | <br>run:<br>Comparison of integers: 0<br>BUILD SUCCESSFUL (total time: 0 seconds) |

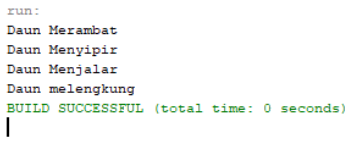
<table>
<tr><td></td><td></td><td>

```
}
```

</td><td></td></tr>
<tr><td>9</td><td>Buatlah program bebas menggunakan polimorfis dinamis dengan jumlah class : Akhiran NIM ganjil : 3 class Akhiran NIM genap : 4 class</td><td>

```
package jobsheet9;

class Tanaman {

    public void tumbuh() {

        System.out.println("Daun Merambat");

    }

}


class Bunga extends Tanaman {


    public void tumbuh() {

        System.out.println("Daun Menyipir");

    }

}


class Pohon extends Tanaman {


    public void tumbuh() {

        System.out.println("Daun Menjalar");

    }

}




class Semak extends Tanaman {


    public void tumbuh() {

        System.out.println("Daun melengkung");

    }

}




public class Jobsheet9 {
```

</td><td>



```
run:
Daun Merambat
Daun Menyipir
Daun Menjalar
Daun melengkung
BUILD SUCCESSFUL (total time: 0 seconds)
```

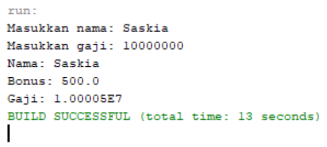
</td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td><pre>
    public static void main(String[] args) {
        Tanaman tanaman91 = new Tanaman();
        tanaman91.tumbuh();

        tanaman91 = new Bunga();
        tanaman91.tumbuh();

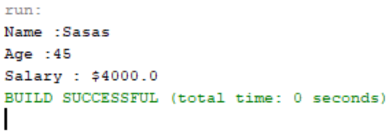
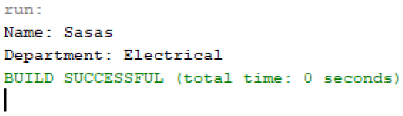
        tanaman91 = new Pohon();
        tanaman91.tumbuh();

        tanaman91 = new Semak();
        tanaman91.tumbuh();
    }
}
</pre></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Modifikasi program disamping menggunakan inputan dinamis</td>
<td><pre>
package jobsheet10;
import java.util.Scanner;
public class Jobsheet10 {
    private String name;
    private double salary;
    private static double
salary_rise_percent = 0.2;

    public Jobsheet10 (String nm, double
sly) {
        this.setName(nm);
        this.setSalary(sly);
    }
    public void setName(String nm) {
        name = nm;
    }
    public void setSalary(double sly) {
        salary = sly;
    }
</pre></td>
<td>
<pre>
run:
Masukkan nama: Saskia
Masukkan gaji: 10000000
Nama: Saskia
Bonus: 500.0
Gaji: 1.00005E7
BUILD SUCCESSFUL (total time: 13 seconds)
</pre></td>
</tr>
</table>

```
    public static void setPresentase(double percent) {
        salary_rise_percent = percent;
    }

    public String getName() {
        return name;
    }

    public double getSalary() {
        return salary;
    }

    public static double getPresentase() {
        return salary_rise_percent;
    }

    public void salaryUp() {
        salary += (salary * salary_rise_percent);
    }
}

class Manager extends Jobsheet10  {
    private static double bonus = 500;

    public Manager(String nm, double sly) {
        super(nm, sly);
    }

    public double getBonus() {
        return bonus;
```

```java
    }

    public void setBonus(double bns) {
        bonus = bns;
    }

    public double getSalary() {
        double salaryBase = super.getSalary();
        return (salaryBase + bonus);
    }
}

class TestManager {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Masukkan nama: ");
        String name = input.next();

        System.out.print("Masukkan gaji: ");
        double salary = input.nextDouble();

        Manager mng = new Manager(name, salary);

        System.out.println("Nama: " + mng.getName());
        System.out.println("Bonus: " + mng.getBonus());
        System.out.println("Gaji: " + mng.getSalary());
    }
}
```

| | | | |
|---|---|---|---|
| 11 | Buatlah program tambahan mengikuti contoh disamping lalu mengganti keyword super menjadi this | package jobsheet11;<br>class Person {<br>    String name = "Dori";<br>    int age = 21;<br>}<br><br>class Lecture extends Person {<br>    float salary = 4000f;<br>    String name = "Sasas";<br>    int age = 45;<br><br>    public void showInfo() {<br>        System.out.println("Name :" + this.name);<br>        System.out.println("Age :" + this.age);<br>        System.out.println("Salary : $" + salary);<br>    }<br>}<br><br><br>public class Jobsheet11 {<br>    public static void main(String[] args) {<br>        Lecture rismon = new Lecture();<br>        rismon.showInfo();<br>    }<br>} | <br>run:<br>Name :Sasas<br>Age :45<br>Salary : $4000.0<br>BUILD SUCCESSFUL (total time: 0 seconds) |
| 12 | Modifikasi program mengikuti contoh disamping (bebas) | package jobsheeet12;<br>import java.util.Date;<br>class Employee {<br>    private static final double BASE_SALARY = 15000.00;<br>    private String name; | <br>run:<br>Name: Sasas<br>Department: Electrical<br>BUILD SUCCESSFUL (total time: 0 seconds) |

```
    private double salary;

    private Date birthDate;



    public Employee(String name, double
salary, Date DoB) {

        this.name = name;

        this.salary = salary;

        this.birthDate = DoB;

    }



    public Employee(String name, double
salary) {

        this(name, salary, null);

    }



    public Employee(String name, Date DoB) {

        this(name, BASE_SALARY, DoB);

    }



    public Employee(String name) {

        this(name, BASE_SALARY);

    }



    public String getName() {

        return name;

    }

}



class Manager extends Employee {

    private String department;



    public Manager(String name, double
salary, String dept) {

        super(name, salary);
        department = dept;
    }

    public Manager(String name, String dept)
{
        super(name);
        department = dept;
    }

    public String getDepartment() {
        return department;
    }
}

public class Jobsheeet12 {
    public static void main(String[] args) {
        Employee man = new Manager("Sasas",
15000.00, "Electrical");

        if (man instanceof Manager) {
            Manager manager = (Manager) man;
            System.out.println("Name: " +
man.getName());
            System.out.println("Department:
" + manager.getDepartment());
        }
    }
}
```