

**LAPORAN PRAKTIKUM 6**  
**INHERITANCE, ABSTRACT CLASS, INTERFACE**

*Laporan Ini Diajukan Untuk Memenuhi Tugas*  
*Mata Kuliah Praktikum Pemrograman Berorientasi Objek*



Disusun oleh :

Ari Maulana Hardan

211511007

**POLITEKNIK NEGERI BANDUNG**  
**PROGRAM STUDI D3 TEKNIK INFORMATIKA**  
**TAHUN 2022**

## A. Source Code

### a. Sortable.java

```
package task3;

public class Sortable {
    public static void shell_sort(Sortable[] array) {
        int i, j, increment;
        Sortable temp;
        increment = 3;
        while (increment > 0) {
            for (i = 0; i < array.length; i++) {
                j = i;
                temp = array[i];
                while ((j >= increment) && (array[j - increment].compareTo(temp) == 1)) {
                    array[j] = array[j - increment];
                    j = j - increment;
                }
                array[j] = temp;
            }
            if (increment / 2 != 0)
                increment = increment / 2;
            else if (increment == 1)
                increment = 0;
            else
                increment = 1;
        }
    }

    public int compareTo(Sortable other) {
        return 0;
    }
}
```

### b. Employee.java

```
package task3;

class Employee extends Sortable{
    public Employee(String n, double s, int day, int month, int year) {
        name = n;
        salary = s;
        hireday = day;
        hiremonth = month;
        hireyear = year;
    }

    public void print() {
```

```

        System.out.println(name + " " + salary + " " + hireYear());
    }

    public void raiseSalary(double byPercent) {
        salary *= 1 + byPercent / 100;
    }

    public int hireYear() {
        return hireyear;
    }

    @Override
    public int compareTo(Sortable other) {
        Employee otherEmployee = (Employee) other;
        if (this.salary < otherEmployee.salary)
            return -1;
        if (this.salary > otherEmployee.salary)
            return 1;
        return 0;
    }

    private String name;
    private double salary;
    private int hireday;
    private int hiremonth;
    private int hireyear;
}

```

### c. EmployeeTest.java

```

package task3;

public class EmployeeTest {
    public static void main(String[] args) {
        Employee[] staff = new Employee[4];
        staff[0] = new Employee("Antonio Rossi", 2000000, 1, 10, 1989);
        staff[1] = new Employee("Maria Bianchi", 2500000, 1, 12, 1991);
        staff[2] = new Employee("Isabel Vidal", 3000000, 1, 11, 1993);
        staff[3] = new Employee("H2SO4", 2700000, 1, 11, 1993);

        Sortable.shell_sort(staff);
        int i;
        for (i = 0; i < 4; i++)
            staff[i].raiseSalary(5);
        for (i = 0; i < 4; i++)
            staff[i].print();
    }
}

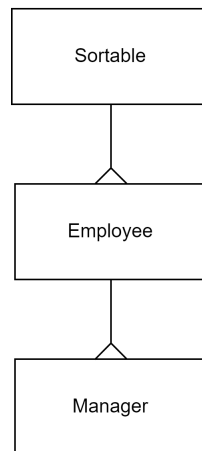
```

#### d. ManagerTest.java

```
package task3;

public class ManagerTest {
    public static void main(String[] args) {
        Employee[] staff = new Employee[3];
        staff[0] = new Employee("Antonio Rossi", 2000000, 1, 10, 1989);
        staff[1] = new Employee("Isabel Vidal", 3000000, 1, 11, 1993);
        staff[2] = new Manager("Maria Bianchi", 2500000, 1, 12, 1991);
        Sortable.shell_sort(staff);
        int i;
        for (i = 0; i < 3; i++)
            staff[i].raiseSalary(5);
        for (i = 0; i < 3; i++)
            staff[i].print();
    }
}
```

## 1. Diagram Hirarki Kelas



## 2. Case 1

Hal yang dilakukan:

- 1) Menambahkan class Sortable yang berfungsi untuk melakukan sorting, class ini merupakan superclass dari class Employee.
- 2) Melakukan overriding pada method compareTo di class Employee. Pada kasus ini overriding method bertujuan untuk melakukan komparasi atribut salary, agar proses sorting pada class Sortable dilakukan berdasarkan atribut salary.

```
@Override
public int compareTo(Sortable other) {
    Employee otherEmployee = (Employee) other;
    if (this.salary < otherEmployee.salary)
        return -1;
    if (this.salary > otherEmployee.salary)
        return 1;
    return 0;
}
```

Method compareTo yang di override ini nantinya akan dipanggil pada method shell\_sort() di class Sortable.

```
package task3;

public class Sortable {
    public static void shell_sort(Sortable[] array) {
        int i, j, increment;
        Sortable temp;
```

```

        increment = 3;
        while (increment > 0) {
            for (i = 0; i < array.length; i++) {
                j = i;
                temp = array[i];
                while ((j >= increment) && (array[j - increment].compareTo(temp)
== 1)) {
                    array[j] = array[j - increment];
                    j = j - increment;
                }
                array[j] = temp;
            }
            if (increment / 2 != 0)
                increment = increment / 2;
            else if (increment == 1)
                increment = 0;
            else
                increment = 1;
        }
    }

    public int compareTo(Sortable other) { //→ Dioveride pada class employee
        return 0;
    }
}

```

- 3) Sehingga array dari object Employee dapat di sort berdasarkan salary seperti pada contoh berikut:

```

package task3;

public class EmployeeTest {
    public static void main(String[] args) {
        Employee[] staff = new Employee[4];
        staff[0] = new Employee("Antonio Rossi", 2000000, 1, 10,
1989);
        staff[1] = new Employee("Maria Bianchi", 2500000, 1, 12,
1991);
        staff[2] = new Employee("Isabel Vidal", 3000000, 1, 11, 1993);
        staff[3] = new Employee("H2SO4", 2700000, 1, 11, 1993);

        Sortable.shell_sort(staff);
        int i;
        for (i = 0; i < 4; i++)
    }
}

```

```
        staff[i].raiseSalary(5);  
    for (i = 0; i < 4; i++)  
        staff[i].print();  
    }  
}
```

Output

```
<terminated> EmployeeTest [Java Application] C:\Program Files\Java\j  
Antonio Rossi 2100000.0 1989  
Maria Bianchi 2625000.0 1991  
H2S04 2835000.0 1993  
Isabel Vidal 3150000.0 1993  
|
```

### 3. Case 2

- 1) Kita dapat membuat object Manager pada variable yang bertipe Employee karena Manager adalah subclass dari Employee (Polymorphism).
- 2) Karena class Manager merupakan subclass dari class Employee, class ini dapat mengakses method shell\_sort() dan method compareTo yang sudah di override tadi. Oleh karena itu, object dari Manager dapat juga di sort berdasarkan salary.

```
package task3;

public class ManagerTest {
    public static void main(String[] args) {
        Employee[] staff = new Employee[3];
        staff[0] = new Employee("Antonio Rossi", 2000000, 1, 10,
1989);
        staff[1] = new Employee("Isabel Vidal", 3000000, 1, 11, 1993);
        staff[2] = new Manager("Maria Bianchi", 2500000, 1, 12,
1991);

        Sortable.shell_sort(staff);
        int i;
        for (i = 0; i < 3; i++)
            staff[i].raiseSalary(5);
        for (i = 0; i < 3; i++)
            staff[i].print();
    }
}
```

#### Output

```
Antonio Rossi 2100000.0 1989
Maria Bianchi 3012500.0 1991
Isabel Vidal 3150000.0 1993
```

#### Link Github :

<https://github.com/arimaulanahardan/LearnJava-Praktikum-6-Inheritance-abstract-class-Interface-.git>