16TKO3043 PEMROGRAMAN BERORIENTASI OBJEK

W6 – Studi Kasus 6

LAPORAN

Oleh:

M. Fatur Maulidan Azzahra 211511020

PROGRAM STUDI D-III

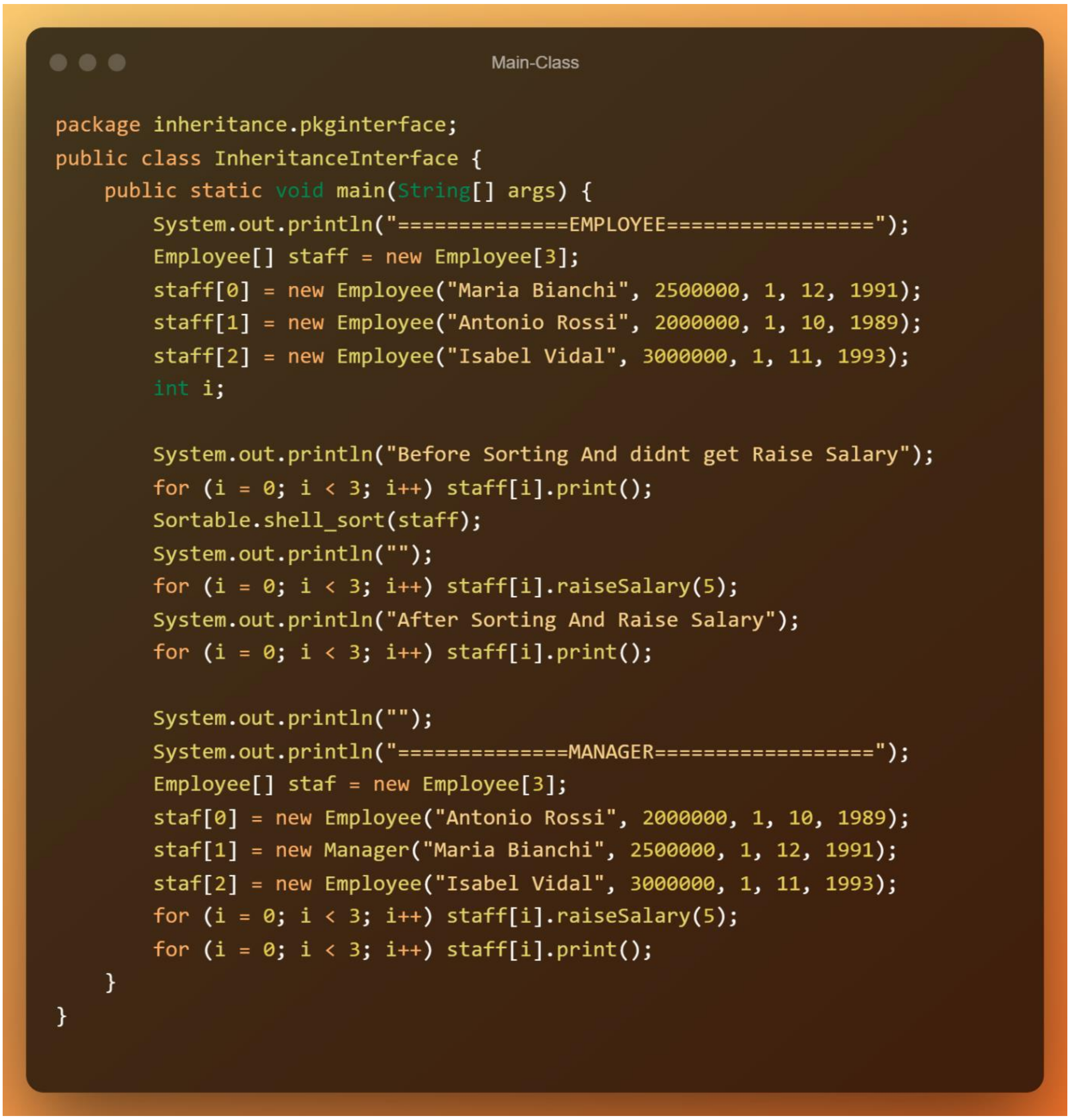
TEKNIK INFORMATIKA JURUSAN TEKNIK INFORMATIKA DAN

KOMPUTER POLITEKNIK NEGERI BANDUNG

2022

Kasus 1 -

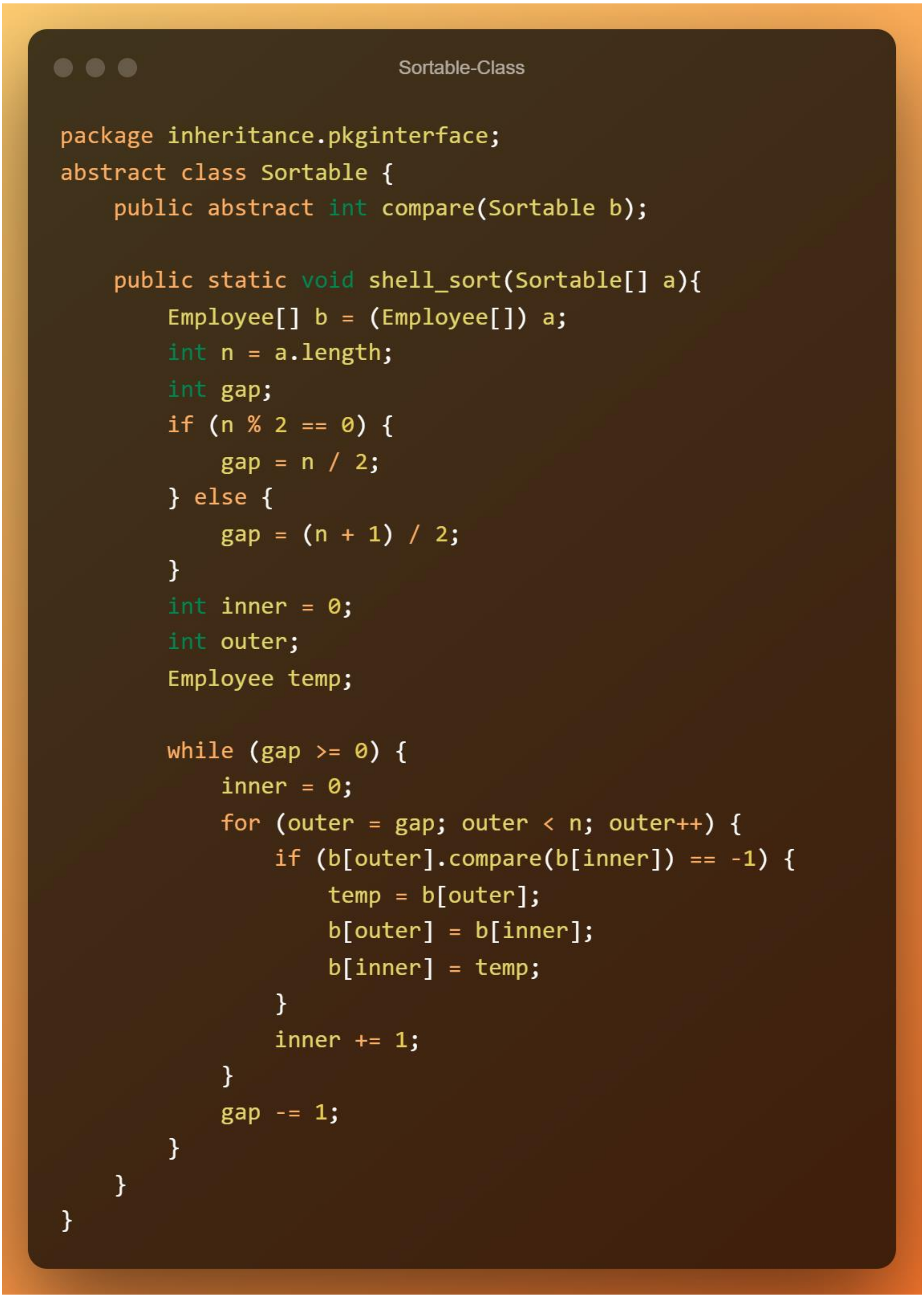
- **Tangkapan Layar Hasil Akhir Program**



```
package inheritance.pkginterface;
public class InheritanceInterface {
    public static void main(String[] args) {
        System.out.println("==============EMPLOYEE=================");
        Employee[] staff = new Employee[3];
        staff[0] = new Employee("Maria Bianchi", 2500000, 1, 12, 1991);
        staff[1] = new Employee("Antonio Rossi", 2000000, 1, 10, 1989);
        staff[2] = new Employee("Isabel Vidal", 3000000, 1, 11, 1993);
        int i;

        System.out.println("Before Sorting And didnt get Raise Salary");
        for (i = 0; i < 3; i++) staff[i].print();
        Sortable.shell_sort(staff);
        System.out.println("");
        for (i = 0; i < 3; i++) staff[i].raiseSalary(5);
        System.out.println("After Sorting And Raise Salary");
        for (i = 0; i < 3; i++) staff[i].print();

        System.out.println("");
        System.out.println("==============MANAGER==================");
        Employee[] staf = new Employee[3];
        staf[0] = new Employee("Antonio Rossi", 2000000, 1, 10, 1989);
        staf[1] = new Manager("Maria Bianchi", 2500000, 1, 12, 1991);
        staf[2] = new Employee("Isabel Vidal", 3000000, 1, 11, 1993);
        for (i = 0; i < 3; i++) staff[i].raiseSalary(5);
        for (i = 0; i < 3; i++) staff[i].print();
    }
}
```

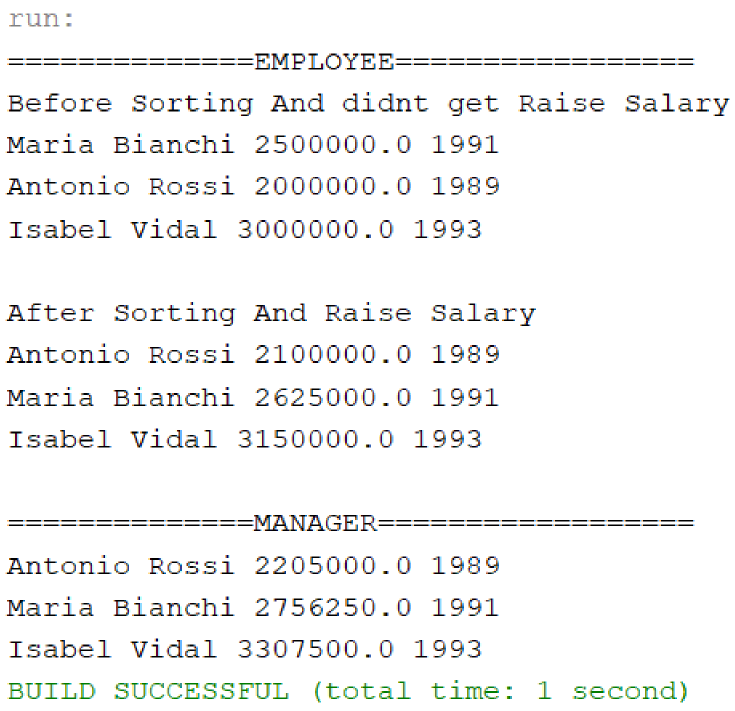


```java
package inheritance.pkginterface;
abstract class Sortable {
    public abstract int compare(Sortable b);

    public static void shell_sort(Sortable[] a){
        Employee[] b = (Employee[]) a;
        int n = a.length;
        int gap;
        if (n % 2 == 0) {
            gap = n / 2;
        } else {
            gap = (n + 1) / 2;
        }
        int inner = 0;
        int outer;
        Employee temp;

        while (gap >= 0) {
            inner = 0;
            for (outer = gap; outer < n; outer++) {
                if (b[outer].compare(b[inner]) == -1) {
                    temp = b[outer];
                    b[outer] = b[inner];
                    b[inner] = temp;
                }
                inner += 1;
            }
            gap -= 1;
        }
    }
}
```

- **Tangkapan Layar Setiap Jawaban Soal yang Dipertanyakan**

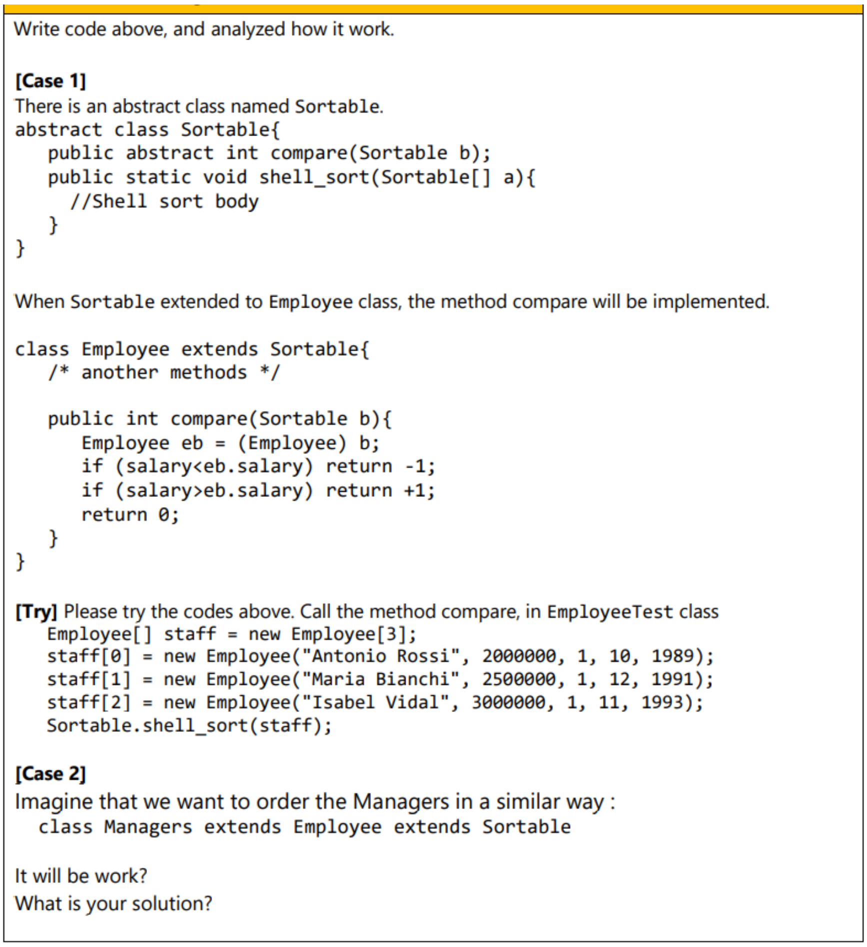


```
run:
==============EMPLOYEE=================
Before Sorting And didnt get Raise Salary
Maria Bianchi 2500000.0 1991
Antonio Rossi 2000000.0 1989
Isabel Vidal 3000000.0 1993

After Sorting And Raise Salary
Antonio Rossi 2100000.0 1989
Maria Bianchi 2625000.0 1991
Isabel Vidal 3150000.0 1993

==============MANAGER==================
Antonio Rossi 2205000.0 1989
Maria Bianchi 2756250.0 1991
Isabel Vidal 3307500.0 1993
BUILD SUCCESSFUL (total time: 1 second)
```

- **Permasalahan yang dihadapi**



Write code above, and analyzed how it work.

[Case 1]

There is an abstract class named Sortable.

```
abstract class Sortable{
   public abstract int compare(Sortable b);
   public static void shell_sort(Sortable[] a){
     //Shell sort body
   }
}
```

When Sortable extended to Employee class, the method compare will be implemented.

```
class Employee extends Sortable{
   /* another methods */

   public int compare(Sortable b){
      Employee eb = (Employee) b;
      if (salary<eb.salary) return -1;
      if (salary>eb.salary) return +1;
      return 0;
   }
}
```

[Try] Please try the codes above. Call the method compare, in EmployeeTest class

```
   Employee[] staff = new Employee[3];
   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);
   Sortable.shell_sort(staff);
```

[Case 2]

Imagine that we want to order the Managers in a similar way :

```
  class Managers extends Employee extends Sortable
```

It will be work?

What is your solution?

- **Solusi dari Permasalahan yang dihadapi**
  1. Case 1 Menambahkan Sortable
  2. Case 2 Tidak bisa double extends dalam kasus penggunaan bahasa Java salah satu perlu menjadi interface. Misalkan sortable menjadi interface nanti
- **Nama teman yang membantu memecahkan permasalahan di persoalan ini (Referensi sumber)**

  Case 1 Aldrin 211511003

  Case 2 PBO Teori