

Nama : Welson Mario Naibaho

Kelas : TI 2C

No Absen : 22

Hal : Laporan Praktikum PBO

Class Employee

```
12 public class Employee {
13     protected String name;
14
15     public String getEmployeeInfo() {
16         return "Name = " + name;
17     }
18 }
```

Class Payable

```
13 public interface Payable {
14     public int getPaymentAmount();
15 }
```

Class InternshipEmployee

```
11 public class InternshipEmployee extends Employee{
12     private int length;
13
14     public InternshipEmployee(String name, int length){
15         this.length = length;
16         super.name = name;
17     }
18
19
20     public int getLength() {
21         return length;
22     }
23
24     public void setLength(int length) {
25         this.length = length;
26     }
27
28     @Override
29     public String getEmployeeInfo(){
30         String info = super.getEmployeeInfo()+"\n";
31         info += "Registered as internship employee for "+length+" month/s\n";
32         return info;
33     }
34 }
```

Class PermanentEmployee

```
11 public class PermanentEmployee extends Employee implements Payable {
12
13     private int salary;
14
15     public PermanentEmployee(String name, int salary) {
16         super.name = name;
17         this.salary = salary;
18     }
19
20     public int getSalary() {
21         return salary;
22     }
23
24     public void setSalary(int salary) {
25         this.salary = salary;
26     }
27     @Override
28     public int getPaymentAmount() {
29         return (int) (salary+0.05*salary);
30     }
31     @Override
32     public String getEmployeeInfo() {
33         String info = super.getEmployeeInfo()+"\n";
34         info += "Registered as permanent employee with salary "+salary+"\n";
35         return info;
36     }
37 }
```

Class ElectricityBill

```
11 public class ElectricityBill extends Employee implements Payable {
12
13     private int kwh;
14     private String category;
15
16     public ElectricityBill(int kwh, String category) {
17         this.kwh = kwh;
18         this.category = category;
19     }
20
21     public int getKwh() {
22         return kwh;
23     }
24
25     public void setKwh(int kwh) {
26         this.kwh = kwh;
27     }
28
29     public String getCategory() {
30         return category;
31     }
32
33     public void setCategory(String category) {
34         this.category = category;
35     }
36 }
```

Lanjutan

```
37  @Override
38  public int getPaymentAmount() {
39      return kwh * getBasePrice();
40  }
41
42  public int getBasePrice() {
43      int bPrice = 0;
44      switch (category) {
45          case "R-1":
46              bPrice = 100;
47              break;
48          case "R-2":
49              bPrice = 200;
50              break;
51      }
52      return bPrice;
53  }
54
55  public String getBillInfo() {
56      return "KWH = " + kwh + "\n"
57             + "category = " + category + "(" + getBasePrice() + " per KWH)\n";
58  }
59 }
```

Class Tester1 (Main)

```
11  public class Tester1 {
12
13      public static void main(String[] args) {
14          PermanentEmployee pEmp = new PermanentEmployee("Dedik", 500);
15          InternshipEmployee iEmp = new InternshipEmployee("Sunarto", 5);
16          ElectricityBill eBill = new ElectricityBill(5, "A-1");
17          Employee e;
18          Payable p;
19          e = pEmp;
20          e = iEmp;
21          p = pEmp;
22          p = eBill;
23      }
24  }
```

Tester 2 (Main)

```
11  public class Tester2 {
12      public static void main(String[] args) {
13          PermanentEmployee pEmp = new PermanentEmployee("dedik", 500);
14          Employee e;
15          e = pEmp;
16          System.out.println(""+e.getEmployeeInfo());
17          System.out.println("-----");
18          System.out.println(""+pEmp.getEmployeeInfo());
19      }
20  }
```

Output

```
Name = dedik
Registered as permanent employee with salary 500
```

```
Name = dedik
Registered as permanent employee with salary 500
```

```
BUILD SUCCESSFUL (total time: 0 seconds)
```

Tester 3 (Main)

```
11 public class Tester3 {
12     public static void main(String[] args) {
13         PermanentEmployee pEmp = new PermanentEmployee("Dedik", 500);
14         InternshipEmployee iEmp = new InternshipEmployee("Sunarto", 5);
15         ElectricityBill eBill = new ElectricityBill(5, "A-1");
16         Employee e[] = {pEmp, iEmp};
17         Payable p[] = {pEmp, eBill};
18         Employee e2[] = {pEmp, iEmp, eBill};
19     }
20 }
21 }
```

Tester 4 (Main)

```
11 public class Tester4 {
12     public static void main(String[] args) {
13         Owner ow = new Owner();
14         ElectricityBill eBill = new ElectricityBill(5, "R-1");
15         ow.pay(eBill); //pay for electricity bill
16         System.out.println("-----");
17
18         PermanentEmployee pEmp = new PermanentEmployee("Dedik", 500);
19         ow.pay(pEmp);
20         System.out.println("-----");
21
22         InternshipEmployee iEmp = new InternshipEmployee("Sunarto", 5);
23         ow.showMyEmployee(pEmp);
24         System.out.println("-----");
25         ow.showMyEmployee(iEmp);
26     }
27 }
```

Output

```
Total payment = 500
KWH = 5
category = R-1(100 per KWH)

-----
Total payment = 525
Name = Dedik
Registered as permanent employee with salary 500

-----
Name = Dedik
Registered as permanent employee with salary 500

You have to pay her/him monthly!!
-----
Name = Sunarto
Registered as internship employee for 5 month/s

No need to pay him/her
BUILD SUCCESSFUL (total time: 0 seconds)
```

Jawaban Pertanyaan Percobaan 1

1. InternshipEmployee dan PermanentEmployee
2. PermanentEmployee dan ElectricityBill
3. Karena class pEmp dan iEmp mengextends class e / Employee
4. Karena class pEmp dan eBill mengimplemen class p / Payable
5. P = iEmp Error karena tdk mengimplemen class Payable, e=eBill error karena tidak mengextends class Employee
6. Polimorfisme adalah konsep dalam pbo dimana beberapa class memiliki method yang sama

Jawaban Pertanyaan Percobaan 2

1. Karena pEmp memiliki parameter yang sama, dan karena class pEmp mengextends class Employee dan terdapat e = pEmp
2. Karena objek pEmp dideklarasikan dari class PermanentEmployee yang mengextends class Employee
3. Karena method sudah dikenali oleh java dan method yang dijalankan oleh JVM berbeda

Jawaban Pertanyaan Percobaan 3

1. Karena pEmp dan iEmp merupakan turunan dari class Employee
2. Karena pEmp dan eBill mengimplements class Payable
3. Terjadi error karena eBill tidak mengextends class Employee

Jawaban Pertanyaan Percobaan 4

1. Karena eBill dan pEmp mengimplements class Payable dan method pay yang dimiliki class, Payable juga dipanggil oleh class Owner
2. Agar owner dapat memanggil method pay yang didalamnya sudah diimplemen oleh class ElectricityBill dan PermanentEmployee
3. Error karena class iEmp tidak mengimplements class Payable, namun jika class InternshipEmployee mengimpelements Payable, maka tidak terjadi error
4. Perlu, untuk mengecek apakah objek pay pada payable p itu merupakan hasil instansiasi dari class ElectricityBill
5. Dikarenakan objek eb mengimplements class Payable