LAPORAN TUGAS PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK

Laporan ini disusun untuk memenuhi Tugas Mata Kuliah Pemrograman Berorientasi Objek

Disusun Oleh :

Benny Yoga Suhardi 211511035

PROGRAM STUDI D3 TEKNIK INFORMATIKA

JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA

POLITEKNIK NEGERI BANDUNG

TAHUN 2022

Studi Kasus 1

Membuat Class Commission.java

```
1
2  public class Commission extends Hourly{
3      double totalSales;
4      double commissionRate;
5
6      public Commission(String eName, String eAddress, String ePhone,
7              String socSecNumber, double rate, double rateEmployee) {
8          super(eName,eAddress,ePhone,socSecNumber,rate);
9          commissionRate = rateEmployee;
10     }
11
12     public void addSales(double TotalSales) {
13         totalSales += TotalSales;
14     }
15
16     public double pay() {
17         double payment = commissionRate + totalSales + super.pay();
18         totalSales = 0;
19         return payment;
20     }
21
22     public String toString() {
23         String result = super.toString();
24         result += "\nCurrent Total Sales: " + totalSales;
25
26         return result;
27     }
28
29 }
30
```

Ketentuan dari commission.java sesuai dengan Studi kasus

- Comission extend dari Hourly.java
- Mempunyain 2 insance variable yaitu totalsales dan commissionRate

```
double totalSales;
double commissionRate;
```

- Mempunyai constructor dengan 6 Parameter sesuai dengan yang ada pada

```
public Commission(String eName, String eAddress, String ePhone,
        String socSecNumber, double rate, double rateEmployee) {
    super(eName,eAddress,ePhone,socSecNumber,rate);
    commissionRate = rateEmployee;
}
```

- Mempunyai procedure addsales dengan merepresentasikan instance variable total sales

```
public void addSales(double TotalSales) {
    totalSales += TotalSales;
}
```

- Method pay dimana memanggil method class induknya kemudian ditambah dengan commission rate dan totalsales kemudian di set Kembali menjadi 0

```
public double pay() {
    double payment = commissionRate + totalSales + super.pay();
    totalSales = 0;
    return payment;
}
```

- Method to string yang memanggil method induknya kemudian ditambah dengan total sales

```
public String toString() {
    String result = super.toString();
    result += "\nCurrent Total Sales: " + totalSales;

    return result;
}
```

Membuat Class Staff Java untuk mengetes class

```
public class Staff {
    private StaffMember[] staffList;

    public Staff() {
        staffList = new StaffMember[8];

        staffList[0] = new Executive("Sam", "123 Main Line", "555-0469",
                "123-45-6789", 2423.07);

        staffList[1] = new Employee ("Carla", "456 Off Line",
                "555-0101", "987-65-4321", 1246.15);

        staffList[2] = new Employee ("Woody", "789 Off Rocker",
                "555-0000", "010-20-3040", 1169.23);

        staffList[3] = new Hourly ("Diane", "678 Fifth Ave.",
                "555-0690", "958-47-3625", 10.55);

        staffList[4] = new Volunteer ("Norm", "987 Suds Blvd.",
                "555-8374");

        staffList[5] = new Volunteer ("Cliff", "321 Duds Lane",
                "555-7282");

        staffList[6] = new Commission("Brownz", "3321 Bandung",
                "214-421", "012214-421", 6.25, 0.2);

        staffList[7] = new Commission("Suuuu", "321 Rancaekek",
                "3321-421", "001-221-332", 9.75, 0.15);
```

```
        ((Executive) staffList[0]).awardBonus(500.00);

        ((Hourly) staffList[3]).addHours(40);

        ((Commission)staffList[6]).addHours(35);
        ((Commission)staffList[6]).addSales(400);;

        ((Commission)staffList[7]).addHours(40);
        ((Commission)staffList[7]).addSales(950);;
    }


    public void payday() {
        double amount;

        for (int k = 0; k < staffList.length; k++) {
            System.out.println(staffList[k]);

            amount = staffList[k].pay(); // polymorphic

            if (amount == 0.0)
                System.out.println("Thanks!");
            else    //why can't we invoke pay()'s here?
                System.out.println("Pain: " + amount);

            System.out.println("-----------------------------------------");
        }
    }
}
```

- Mendeklarasikan array dengan Panjang arraynya 8

```
staffList = new StaffMember[8];
```

- Menambahkan 2 object commission pada array yang telah di deklarasikan dan dimasukkan kedalam array dengan commission pertama $6.25 perjam dan 20% commission dan yang kedua $9.75 perjam dan 15% commission

```
staffList[6] = new Commission("Brownz", "3321 Bandung",
        "214-421", "012214-421", 6.25, 0.2);

staffList[7] = new Commission("Suuuu", "321 Rancaekek",
        "3321-421", "001-221-332", 9.75, 0.15);
```

- Melakukan perhitungan pada kedua commission dengan method yang telah tersedia dengan jam kerja 35 jam dengan total salary 400$ untuk commission pertama dan 40 jam kemudian total salarynya adalah 950$ untuk commission kedua kemudian dilakukan parse pada pemanggilan methodnya sesuai dengan jenis object aslinya Ketika dilakukan polymorphism

```
((Commission)staffList[6]).addHours(35);
((Commission)staffList[6]).addSales(400);;

((Commission)staffList[7]).addHours(40);
((Commission)staffList[7]).addSales(950);;
```

Hasil dari Run Program

```
Name : Sam
Address : 123 Main Line
Phone : 555-0469

Social Security Number: 123-45-6789
Pain: 2923.07
---------------------------------------
Name : Carla
Address : 456 Off Line
Phone : 555-0101

Social Security Number: 987-65-4321
Pain: 1246.15
---------------------------------------
Name : Woody
Address : 789 Off Rocker
Phone : 555-0000

Social Security Number: 010-20-3040
Pain: 1169.23
---------------------------------------
```

```
Name : Diane
Address : 678 Fifth Ave.
Phone : 555-0690

Social Security Number: 958-47-3625
Current hours: 40
Pain: 422.0
---------------------------------------
Name : Norm
Address : 987 Suds Blvd.
Phone : 555-8374

Thanks!
---------------------------------------
Name : Cliff
Address : 321 Duds Lane
Phone : 555-7282
```

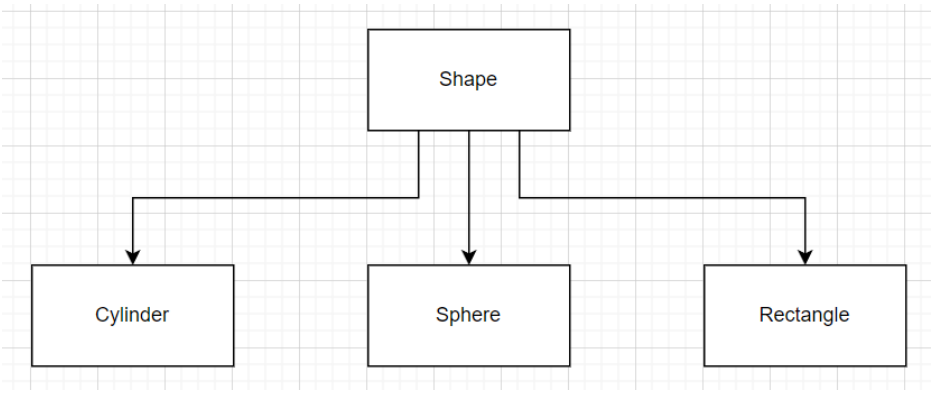
```
---------------------------------------
Name : Brownz
Address : 3321 Bandung
Phone : 214-421

Social Security Number: 012214-421
Current hours: 35
Current Total Sales: 400.0
Pain: 618.95
---------------------------------------
Name : Suuuu
Address : 321 Rancaekek
Phone : 3321-421

Social Security Number: 001-221-332
Current hours: 40
Current Total Sales: 950.0
Pain: 1340.15
---------------------------------------
```

Studi Kasus 2

Pada Program ini dibuat beberapa class yang dimana class tersebut menggunakan inheritance untuk membentuk classnya seperti class Cylinder.java, Paint.java, PaintThings.java, Rectangle.java Shape dan Sphere.java



Soal 1 Membuat Abstract Class shape dengan ketentuan

- Membuatn Instance variable Shapename dengan tipe String
- Membuat absctract method area()
- Membuat toString Method mengembalikan nilai nama dari shape

Untuk hasil dari program sesuai ketentuan tersebut seperti pada source code dibawah ini

```
public abstract class shape {
    private String shapeName;

    public shape(String shapename) {
        this.shapeName = shapename;
    }

    public abstract double area();
    public String toString() {
        String Result = "Shape Name " + this.shapeName;
        return Result;
    }

}
```

Soal 2

- Membuat file sphare.java mengekstend dari induknya shape dengan beberapa ketentuan Mengoverride method area dengan perhitungan 4*PI*radius^2
- Membuat file rectangle.java mengekstend dari induknya shape dengan beberapa ketentuan Mengoverride method area dengan perhitungan Panjang*lebar
- Membuat file Cylinder.java mengekstend dari induknya shape dengan beberapa ketentuan Mengoverride method area dengan perhitungan 4*PI*radius^2*tinggi

Cylinder.java

```java
package kasus2;

public class Cylinder extends shape{
    private double radius;
    private double height;

    public Cylinder(double r, double h) {
        super("Cylinder");
        this.radius = r;
        this.height = h;
    }

    @Override
    public double area() {
        return Math.PI*radius*radius*height;
    }
    public String toString() {
        return super.toString() + " of radius " + radius + " and of height " + height;
    }

}
```

Rectangle.java

```java
package kasus2;

public class Rectangle extends shape{
    private double length;
    private double width;

    public Rectangle(double l, double w) {
        super("Rectangle");
        this.length = l;
        this.width = w;
    }

    @Override
    public double area() {
        return this.width*this.length;
    }
    public String toString() {
        return super.toString() + " of length " + this.length + " and of width " + this.width;
    }
}
```

Sphere.java

```java
package kasus2;

public class Sphere extends shape{
    private double radius;

    public Sphere(double r) {
        super("Sphere");
        this.radius = r;
    }

    @Override
    public double area() {
        return 4*Math.PI*radius*radius;
    }
    public String toString() {
        return super.toString() + " of radius " + radius;
    }

}
```

Soal3 Membuat paint.java yang dimana salah satu methodnya memanggil object lainnya

```java
package kasus2;

public class Paint {
    private double coverage ;
    public Paint(double c) {
        this.coverage = c;
    }

    public double amount(shape s) {
        System.out.println("Computing amount for " + s);
        return s.area();
    }
}
```

Soal 4 Membuat PaintThings.java dengan ketentuan sebagai berikut

- Meninisiasikan 3 object dengan dengan object pertama object dect dengan class rectangle dengan Panjang 20 dan lebar 30 kemudian object kedua bigBall dengan class sphere dangan radius 15 dan yang terakhir membuat object tank dengan class Cylinder dengan radius 10 dan tinggi 30
- Membuat pemanggilan ketiga object tersebut dengan mengassigmentkan kedalam sebuah variable(polimorphysm)
- Mentest programnya dengan merunprogram

```java
package kasus2;

import java.text.DecimalFormat;

public class PaintThings {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        final double coverage = 350;
        Paint paint = new Paint(coverage);

        Rectangle deck = new Rectangle(20,30);
        Sphere bigBall = new Sphere(15);
        Cylinder tank = new Cylinder(10,30);

        double deckamt, ballamt,tankamt;

        deckamt = paint.amount(deck);
        ballamt = paint.amount(bigBall);
        tankamt = paint.amount(tank);

        DecimalFormat fmt = new DecimalFormat("0.#");
        System.out.println("\nNumber of gallons of paint needed...");
        System.out.println("Deck " + fmt.format(deckamt));
        System.out.println("BigBall " + fmt.format(ballamt));
        System.out.println("Tank " + fmt.format(tankamt));
    }

}
```

Hasil Run Program

```
Computing amount for Shape Name Rectangle of length 20.0 and of width 30.0
Computing amount for Shape Name Sphere of radius 15.0
Computing amount for Shape Name Cylinder of radius 10.0 and of height 30.0

Number of gallons of paint needed...
Deck 600
BigBall 2827.4
Tank 9424.8
```

Studi Kasus 3

Case 1

Membuat Numbers.java dimana di dalam class tersebut terdapat algoritma selection sort untuk mensort array dan menampilkannya

```java
package kasus3;
import java.util.Scanner;

public class Number {
    public static void main(String[] args) {
        Integer[] intList;
        int size;

        Scanner scan = new Scanner(System.in);

        System.out.println("\nHow many integers do want to sort?? ");
        size = scan.nextInt();
        intList = new Integer[size];

        System.out.println("\nEnter the number...");
        for(int i =0 ; i<size; i++) {
            intList[i] = scan.nextInt();
        }

        Sorting.selectionSort(intList);
        System.out.println("\nYour number in sorted oerder...");
        for(int i =0;i<size;i++) {
            System.out.println(intList[i] + " ");
        }
        System.out.println();
    }
}
```

Hasil Eksekusi Program

```
How many integers do want to sort??
4

Enter the number...
 4 2 3 1

Your number in sorted oerder...
1
3
2
4
```

Case 2

Membuat String.Java sama seperti number.java membuat algoritma sort tetapi dengan selection sort dan menampilkannya secara ascending

```java
import java.util.Scanner;

public class Strings {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Integer[] intList;
        int size;

        Scanner scan = new Scanner(System.in);

        System.out.println("\nHow many integers do want to sort?? ");
        size = scan.nextInt();
        intList = new Integer[size];

        System.out.println("\nEnter the number...");
        for(int i =0 ; i<size; i++) {
            intList[i] = scan.nextInt();
        }

        Sorting.insertionSort(intList);
        System.out.println("\nYour number in sorted oerder...");
        for(int i =0;i<size;i++) {
            System.out.println(intList[i] + " ");
        }
        System.out.println();
    }

}
```

Hasil Eksekusi Program

```
How many integers do want to sort??
4

Enter the number...
4 2 3 1

Your number in sorted oerder...
4
3
2
1
```

Case 3

Membuat SalesPerson.java

```java
package kasus3;

public class Salesperson implements Comparable{

    private String firstName, lastName;
    private int totalSales;

    public Salesperson(String first, String last, int sales)
    {
        firstName = first;
        lastName = last;
        totalSales = sales;
    }

    public String toString()
    {
        return lastName + ", " + firstName + ":  \t" + totalSales;
    }

    public boolean equals (Object other)
    {
        return (lastName.equals(((Salesperson)other).getLastName()) &&
                firstName.equals(((Salesperson)other).getFirstName()));
    }

```

```java
    @Override
    public int compareTo(Object other) {
        int result = 0;
        if (totalSales > ((Salesperson)other).totalSales) result = 1;
        else if(totalSales < ((Salesperson)other).totalSales) result = -1;
        else {
            if (lastName.compareTo(((Salesperson) other).getLastName()) < 0)
                return 1;
            else
                return -1;
        }
        return result;
    }
    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public int getTotalSales() {
        return totalSales;
    }

}
```

Case 4

Membuat driver Program untuk menampilkan hasil sorting berdasarkan data SalesPerson yang tersedia

```
1  package kasus3;
2
3  public class weeklySales {
4      public static void main(String[] args)
5      {
6          Salesperson [] salesStaff = new Salesperson [4];
7          salesStaff[0] = new Salesperson ("Benny", "Yoga", 3000);
8          salesStaff[1] = new Salesperson ("Rizki", "Halohalo", 4000);
9          salesStaff[2] = new Salesperson ("Nuval", "Ardana", 3000);
10         salesStaff[3] = new Salesperson ("Upap", "Combo", 2800);
11         Sorting.insertionSort(salesStaff);
12         System.out.println("\nRanking of Sales for the week\n");
13         for(Salesperson s : salesStaff)
14         System.out.println(s);
15     }
16 }
17
```

Hasil Eksekusi Program dengan InsertionSort secara descending

```
Ranking of Sales for the week

Halohalo, Rizki:        4000
Ardana, Nuval:          3000
Yoga, Benny:    3000
Combo, Upap:    2800
```

Hasil Eksekusi Program dengan SelectionSort Secara asending

```
Ranking of Sales for the week

Combo, Upap:    2800
Yoga, Benny:    3000
Ardana, Nuval:          3000
Halohalo, Rizki:        4000
```

Link Github : BennyYoga/PBO_Praktikum (github.com)