MODUL III

INHERITANCE DAN POLYMORPHISM

1. Tujuan

a. Mahasiswa mampu menulis program yang *extensible* dan *modifiable*.

b. Mahasiswa mampu mendefinisikan classes yang telah digunakan sebelumnya pada inheritance.

c. Mahasiswa mampu menerapkan polmorfisme untuk message pada beberapa object.

2. Latihan praktikum

Buat file .java di editor masing-masing, dan lakukan latihan pemrograman yang ditunjukkan setiap nomor.

a. Inheritance

```
public class Point2D {

   private int x, y;


   public Point2D() {
      this.x = 0;
      this.y = 0;
   }
   public Point2D(int x, int y) {
      this.x = x;
      this.y = y;
   }


   public int getX() {
      return this.x;
   }
   public void setX(int x) {
      this.x = x;
   }
   public int getY() {
      return this.y;
   }
   public void setY(int y) {
      this.y = y;
   }


   public String toString() {
      return "(" + this.x + "," + this.y + ")";
   }
}
```

```
public class Point3D extends Point#D {

   private int z;


   public Point3D() {
      super();
      this.z = 0;
   }
   public Point3D(int x, int y, int z) {
      super(x, y);
      this.z = z;
   }


   public int getZ() {
      return this.z;
   }
   public void setZ(int z) {
      this.z = z;
   }


   @Override
   public String toString() {
      return "(" + super.getX() + "," + super.getY() + "," + this.z + ")";
   }
}
```

```
public class TestPoint2DPoint3D {
   public static void main(String[] args) {

      Point2D p2a = new Point2D(1, 2);
      System.out.println(p2a);
      Point2D p2b = new Point2D();
      System.out.println(p2b);
      p2a.setX(3);
      p2a.setY(4);
      System.out.println(p2a);
      System.out.println("x: " + p2a.getX());
      System.out.println("x: " + p2a.getY());


      Point3D p3a = new Point3D(11, 12, 13);
      System.out.println(p3a);
      Point2D p3b = new Point3D();
      System.out.println(p3b);

      p3a.setX(21);
      p3a.setY(22);
      p3a.setZ(23);
      System.out.println(p3a);
      System.out.println("x: " + p3a.getX());
      System.out.println("y: " + p3a.getY());
      System.out.println("z: " + p3a.getZ());
   }
}
```

b. Polymorphism

```
    public class Monster {

       private String name;


       public Monster(String name) {
          this.name = name;
       }

       public String attack() {
          return "!^_&^$@+%$* Bingung cara nyerang!! aaarrgh!!!";
        }
    }
```

```
public class FireMonster extends Monster {

   public FireMonster(String name) {
      super(name);
   }

   @Override public String attack() {
      return "Serangan badai API!!!";
   }
}
```

```
public class WaterMonster extends Monster {

   public WaterMonster(String name) {
      super(name);
   }

   @Override public String attack() {
      return "Serangan banjir AIR!!!";
   }
}
```

```
public class StoneMonster extends Monster {

   public StoneMonster(String name) {
      super(name);
   }

   @Override public String attack() {
      return "Serangan hujan BATU!!";
   }
}
```

```
public class TestMonster {
   public static void main(String[] args) {
      Monster m1 = new FireMonster("r2u2");
      Monster m2 = new WaterMonster("u2r2");
```

```
        Monster m3 = new StoneMonster("r2r2");


        System.out.println(m1.attack());
        System.out.println(m2.attack());
        System.out.println(m3.attack());


        m1 = new StoneMonster("a2b2");
        System.out.println(m1.attack());


        Monster m4 = new Monster("u2u2");
        System.out.println(m4.attack());
    }
}
```

3. Tugas praktikum

1. Realisasikan diagram di bawah ini ke dalam bentuk program Java!