LAPORAN PRAKTIKUM

PBO

TUGAS 9

Disusun oleh :

Bintang Pancahaya Prasetyo

244107020115

TI – 2G

Jurusan Teknologi Informasi

Prodi D-IV Teknik Informatika

Politeknik Negeri Malang

POLIMORFISME

Kode Program

- Destroyable

```java
public interface Destroyable {
    public abstract void destroyed();
}
```

- Barrier

```java
public class Barrier implements Destroyable {
    private int strength;

    public Barrier(int strength) {
        this.strength = strength;
    }

    public void setStrength(int strength) {
        this.strength = strength;
    }

    public int getStrength() {
        return this.strength;
    }

    @Override
    public void destroyed() {
        this.strength -= 9;
    }

    public String getBarrierInfo() {
        return "Barrier Strength = " + this.strength;
    }
}
```

- Zombie

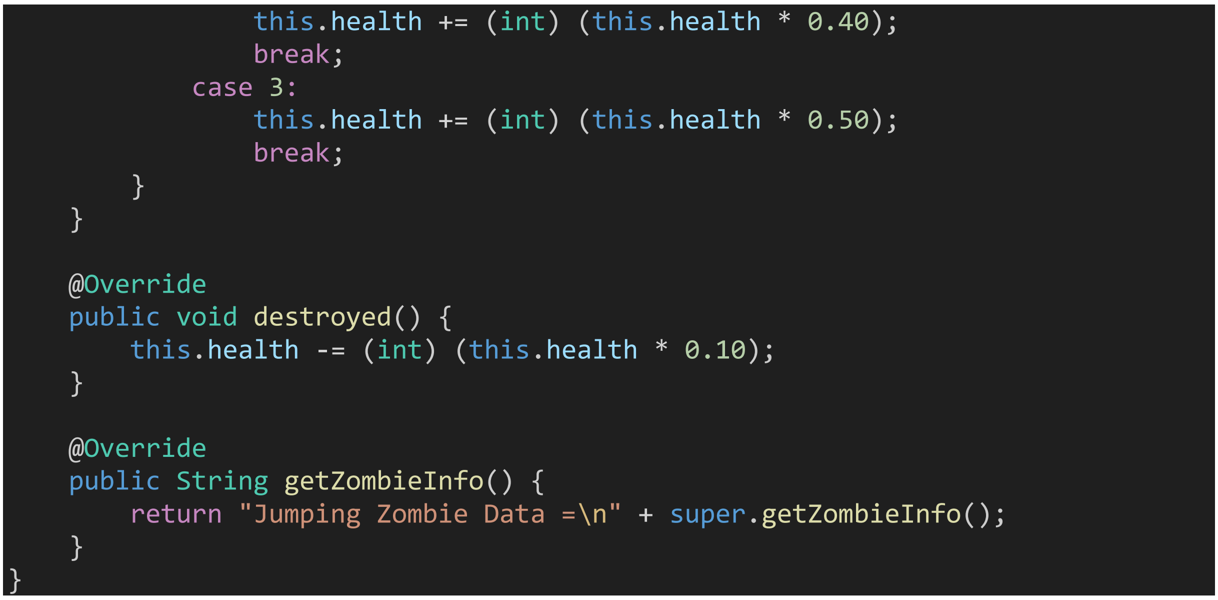
```java
public abstract class Zombie implements Destroyable {
    protected int health;
    protected int level;

    public Zombie(int health, int level) {
        this.health = health;
        this.level = level;
    }

    public abstract void heal();

    @Override
    public abstract void destroyed();

    public String getZombieInfo() {
        return "Health = " + this.health + "\nLevel = " + this.level;
    }
}
```

- WalkingZombie

```
public class WalkingZombie extends Zombie {

    public WalkingZombie(int health, int level) {
        super(health, level);
    }

    @Override
    public void heal() {
        switch (this.level) {
            case 1:
                this.health += (int) (this.health * 0.10);
                break;
            case 2:
                this.health += (int) (this.health * 0.30);
                break;
            case 3:
                this.health += (int) (this.health * 0.40);
                break;
        }
    }

    @Override
    public void destroyed() {
        this.health -= (int) (this.health * 0.20);
    }

    @Override
    public String getZombieInfo() {
        return "Walking Zombie Data =\n" + super.getZombieInfo();
    }
}
```

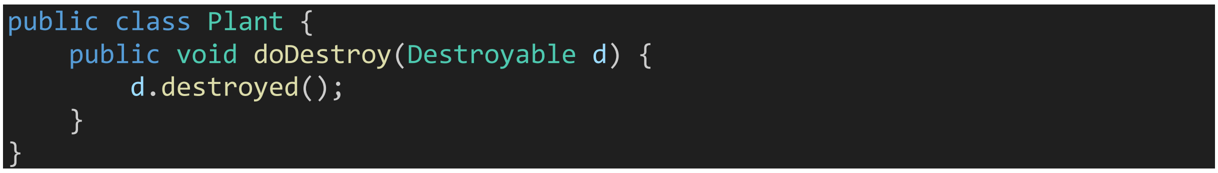
- JumpingZombie

```
public class JumpingZombie extends Zombie {

    public JumpingZombie(int health, int level) {
        super(health, level);
    }

    @Override
    public void heal() {
        switch (this.level) {
            case 1:
                this.health += (int) (this.health * 0.30);
                break;
            case 2:
```



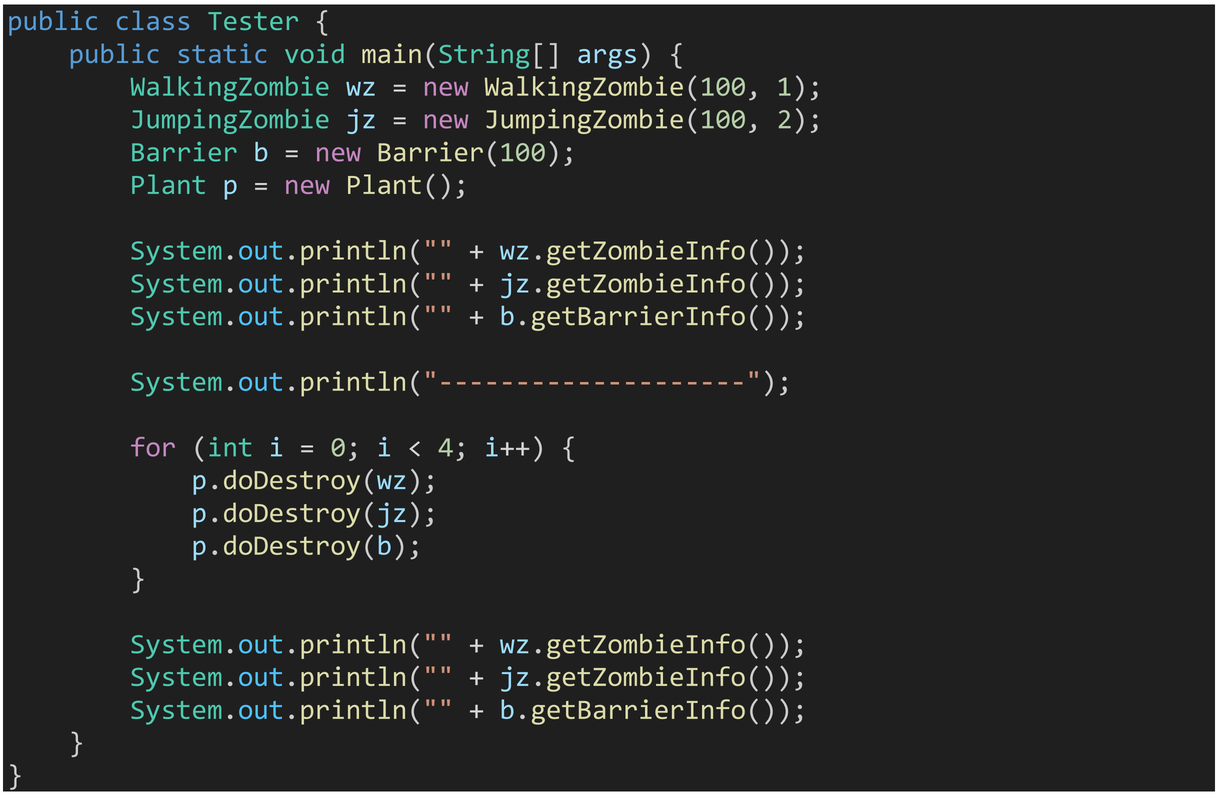
```
                this.health += (int) (this.health * 0.40);
                break;
            case 3:
                this.health += (int) (this.health * 0.50);
                break;
        }
    }

    @Override
    public void destroyed() {
        this.health -= (int) (this.health * 0.10);
    }

    @Override
    public String getZombieInfo() {
        return "Jumping Zombie Data =\n" + super.getZombieInfo();
    }
}
```

- Plant



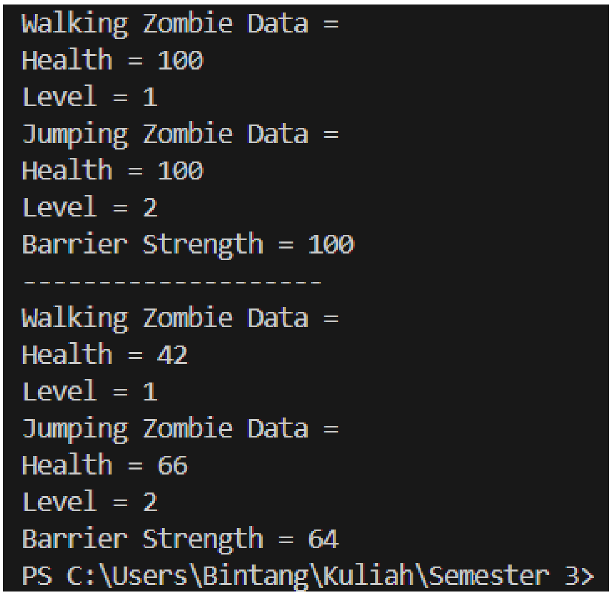
```
public class Plant {
    public void doDestroy(Destroyable d) {
        d.destroyed();
    }
}
```

- Tester



```
public class Tester {
    public static void main(String[] args) {
        WalkingZombie wz = new WalkingZombie(100, 1);
        JumpingZombie jz = new JumpingZombie(100, 2);
        Barrier b = new Barrier(100);
        Plant p = new Plant();

        System.out.println("" + wz.getZombieInfo());
        System.out.println("" + jz.getZombieInfo());
        System.out.println("" + b.getBarrierInfo());

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

        for (int i = 0; i < 4; i++) {
            p.doDestroy(wz);
            p.doDestroy(jz);
            p.doDestroy(b);
        }

        System.out.println("" + wz.getZombieInfo());
        System.out.println("" + jz.getZombieInfo());
        System.out.println("" + b.getBarrierInfo());
    }
}
```

Output



```
Walking Zombie Data =
Health = 100
Level = 1
Jumping Zombie Data =
Health = 100
Level = 2
Barrier Strength = 100
--------------------
Walking Zombie Data =
Health = 42
Level = 1
Jumping Zombie Data =
Health = 66
Level = 2
Barrier Strength = 64
PS C:\Users\Bintang\Kuliah\Semester 3>
```