

## **Teori Tugas 5**

*Disusun untuk memenuhi tugas mata kuliah Pemrograman Berorientasi Objek (Teori)*



Disusun Oleh:

Dwika Ali Ramdhan (231511042)

2B – D3

Jurusan Teknik Komputer dan Informatika

Politeknik Negeri Bandung

2024

## DAFTAR ISI

<b>DAFTAR ISI .....</b>	<b>1</b>
<b>PEMBAHASAN .....</b>	<b>2</b>
1.1. Cari/buat program yang mengimplementasikan konsep abstract class, interface dan implementation dalam satu kasus.....	2
1.1.1. Role.java .....	2
1.1.2. Warrior.java .....	2
1.1.3. Mage.java.....	3
1.1.4. Archer.java.....	3
1.1.5. Weapon.java .....	4
1.1.6. Sword.java .....	4
1.1.7. Staff.java.....	5
1.1.8. Bow.java .....	5
1.2. Buat diagram kelasnya.....	5
1.3. Buat main classnya juga dengan menjalankan aplikasi yang mengimplementasikan konsep data.....	5

## PEMBAHASAN

### 1.1. Cari/buat program yang mengimplementasikan konsep abstract class, interface dan implementation dalam satu kasus.

Disini saya mengambil inspirasi atau contoh kasus pada suatu game RPG dimana terdapat pemilihan role yang paling umum biasanya ada pada game tersebut yaitu Warrior, Mage, dan Archer dimana di tiap role memiliki jenis senjatanya sendiri untuk role Warrior menggunakan Sword, untuk Mage menggunakan Staff, dan untuk Archer menggunakan Bow. Berikut merupakan implementasi programnya :

#### 1.1.1. Role.java

```
public abstract class Role {
    protected String name;
    protected Weapon weapon;

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

    public abstract void attack();
    public abstract void skill();
    public abstract void specialMove();
    public abstract void defend();

    public void roleInfo() {
        System.out.println("Role : " + name);
        System.out.println("Weapon Type: " + weapon.getWeaponUse());
        System.out.println("Weapon Name : " + weapon.getWeaponName());
    }
}
```

Kelas role ini merupakan kelas abstrak menggunakan kata abstract, yang berarti tidak dapat diinstansiasi secara langsung. Kelas ini berfungsi sebagai acuan untuk kelas-kelas turunan. Metode yang dideklarasikan sebagai abstrak yaitu attack(), skill(), specialMove(), dan defend() harus diimplementasikan oleh semua kelas yang mewarisi Role.

Kelas ini adalah superclass untuk kelas-kelas seperti Warrior, Mage, dan Archer, yang mengadopsi semua atribut dan metode yang ada dalam kelas ini.

#### 1.1.2. Warrior.java

```
public class Warrior extends Role {

    public Warrior() {
        super("Warrior");
        this.weapon = new Sword();
    }

    @Override
    public void attack() {
        System.out.println(name + " Basic Attack : Sword Slash");
    }

    @Override
    public void skill() {
        System.out.println(name + " Skill : War Cry");
    }

    @Override
    public void specialMove() {
```

```

        System.out.println(name + " Special Move : Berserker Rage");
    }

    @Override
    public void defend() {
        System.out.println(name + " Defend : Iron Wall");
    }
}

```

Kelas Warrior mengimplementasikan metode-metode abstrak dari kelas Role. Dapat dilihat bagaimana konsep abstraksi digunakan untuk mendefinisikan hal yang dapat dilakukan.

Kelas Warrior adalah subclass dari Role. Ini berarti Warrior mewarisi semua atribut dan metode dari Role, seperti name dan weapon, serta metode roleInfo().

#### 1.1.3. Mage.java

```

public class Mage extends Role {

    public Mage() {
        super("Mage");
        this.weapon = new Staff();
    }

    @Override
    public void attack() {
        System.out.println(name + " Basic Attack : Lightning Bolt");
    }

    @Override
    public void skill() {
        System.out.println(name + " Skill : Lightning Storm");
    }

    @Override
    public void specialMove() {
        System.out.println(name + " Special Move : Summon Lightning Lord");
    }

    @Override
    public void defend() {
        System.out.println(name + " Defend : Magic Barrier");
    }
}

```

Kelas Mage mengimplementasikan metode-metode abstrak dari kelas Role, mendefinisikan tindakan yang berbeda untuk peran ini, seperti serangan sihir dan Gerakan gerakan terkait dengan sihir.

Kelas Mage adalah subclass dari Role, mewarisi atribut dan metode dari kelas induk Role. Kelas ini memberikan implementasi untuk metode yang diwarisi.

#### 1.1.4. Archer.java

```

public class Archer extends Role {

    public Archer() {
        super("Archer");
        this.weapon = new Bow();
    }

    @Override
    public void attack() {
        System.out.println(name + " Basic Attack : Quick Shot");
    }

    @Override

```

```

    public void skill() {
        System.out.println(name + " Skill : Multi-Shot!");
    }

    @Override
    public void specialMove() {
        System.out.println(name + " Special Move : Rain of Arrows");
    }

    @Override
    public void defend() {
        System.out.println(name + " Defend : Evasion Roll");
    }
}

```

Kelas Archer mengimplementasikan metode-metode abstrak dari kelas Role, menyediakan definisi untuk tindakan seorang archer. Kelas ini merupakan subclass dari Role, mewarisi atribut dan metode dari Role, dan memberikan implementasi spesifik untuk metode yang diwarisi.

#### 1.1.5. Weapon.java

```

public abstract class Weapon {
    protected String weaponType;
    private String weaponName;

    public Weapon(String weaponType, String weaponName) {
        this.weaponName = weaponName;
        this.weaponType = weaponType;
    }

    public String getWeaponName() {
        return weaponName;
    }

    public void setWeaponName(String weaponName) {
        this.weaponName = weaponName;
    }

    public String getWeaponUse() {
        return weaponType;
    }
}

```

Kelas ini adalah kelas abstrak yang digunakan untuk mendefinisikan senjata secara umum. Metode getWeaponName() dan getWeaponUse() bukanlah metode abstrak, tetapi mereka memberikan fungsionalitas yang akan digunakan oleh kelas turunan. Kelas ini menjadi superclass bagi kelas-kelas turunan seperti Sword, Staff, dan Bow. Kelas ini akan mewarisi atribut dan metode dari Weapon, dan menyediakan detail mengenai senjata.

#### 1.1.6. Sword.java

```

public class Sword extends Weapon {
    public Sword() {
        super ("Melee", "Blade of Destruction");
    }
}

```

Kelas ini adalah subclass dari Weapon. Sword mewarisi atribut dari Weapon dan menggunakan konstruktor superclass untuk menginisialisasi tipe dan nama senjata. Meskipun tidak ada metode abstrak yang diimplementasikan di sini, kelas ini menyajikan spesifikasi dari kelas abstrak Weapon.

### 1.1.7. Staff.java

```
public class Staff extends Weapon {  
    public Staff() {  
        super("Magic", "Staff of Eternity");  
    }  
}
```

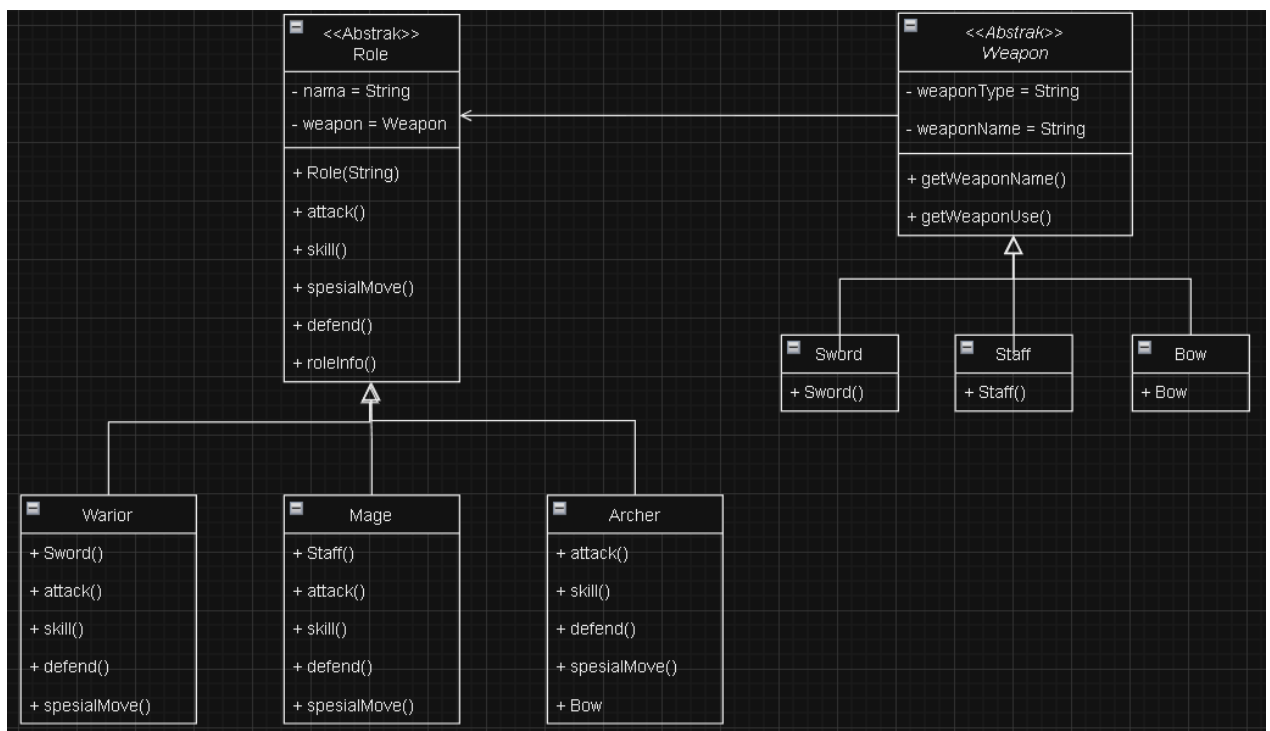
Kelas ini merupakan subclass dari Weapon, mewarisi semua atribut dan metode dari Weapon. Menyediakan implementasi spesifik untuk jenis senjata (staff) dengan menggunakan konstruktor dari Weapon.

### 1.1.8. Bow.java

```
public class Bow extends Weapon {  
    public Bow() {  
        super("Ranged", "Shadowstrike Bow");  
    }  
}
```

Kelas ini adalah subclass dari Weapon, mewarisi semua atribut dan metode dari Weapon. Memberikan spesifikasi konkret dari kelas abstrak Weapon, menggunakan konstruktor untuk mendefinisikan jenis senjata (bow).

## 1.2. Buat diagram kelasnya



## 1.3. Buat main classnya juga dengan menjalankan aplikasi yang mengimplementasikan konsep data.

```
public class Main {  
    public static void main(String[] args) {  
        //Membuat Role  
        Role warrior = new Warrior();  
        Role mage = new Mage();  
        Role archer = new Archer();  
  
        System.out.println();  
    }  
}
```

```

        // Warrior
        warrior.roleInfo();
        warrior.attack();
        warrior.skill();
        warrior.specialMove();
        warrior.defend();

        System.out.println();

        //Mage
        mage.roleInfo();
        mage.attack();
        mage.skill();
        mage.specialMove();
        mage.defend();

        System.out.println();

        //Archer
        archer.roleInfo();
        archer.attack();
        archer.skill();
        archer.specialMove();
        archer.defend();
    }
}

```

Output :

```

Role : Warrior
Weapon Type: Melee
Weapon Name : Blade of Destruction
Warrior Basic Attack : Sword Slash
Warrior Skill : War Cry
Warrior Special Move : Berserker Rage
Warrior Defend : Iron Wall

Role : Mage
Weapon Type: Magic
Weapon Name : Staff of Eternity
Mage Basic Attack : Lightning Bolt
Mage Skill : Lightning Storm
Mage Special Move : Summon Lightning Lord
Mage Defend : Magic Barrier

Role : Archer
Weapon Type: Ranged
Weapon Name : Shadowstrike Bow
Archer Basic Attack : Quick Shot
Archer Skill : Multi-Shot!
Archer Special Move : Rain of Arrows
Archer Defend : Evasion Roll

```