LAPORAN TUGAS TEKNIK PEMRORGAMAN

PERTEMUAN 4

NAMA: FAUZAN JUNIAR MULYANA

NIM: 241524011

POLITEKNIK NEGERI BANDUNG

PROGRAM STUDI D4-TEKNIK INFORMATIKA

2024

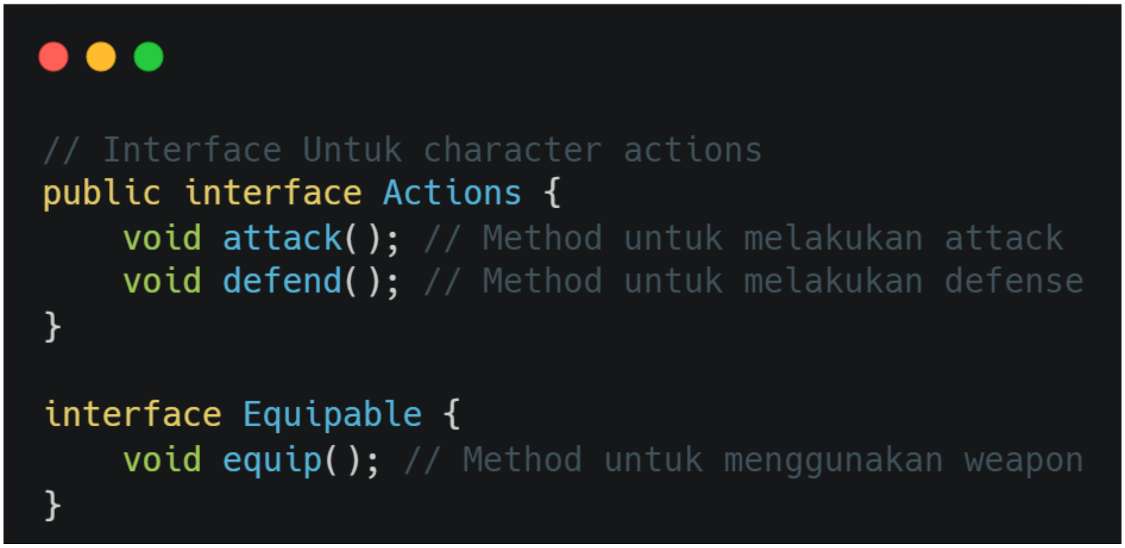
DAFTAR ISI

KASUS RUMIT

Terdapat 10 file yang terdiri dari interface, abstract class, polymorphism, inheritence.

1. INTERFACE

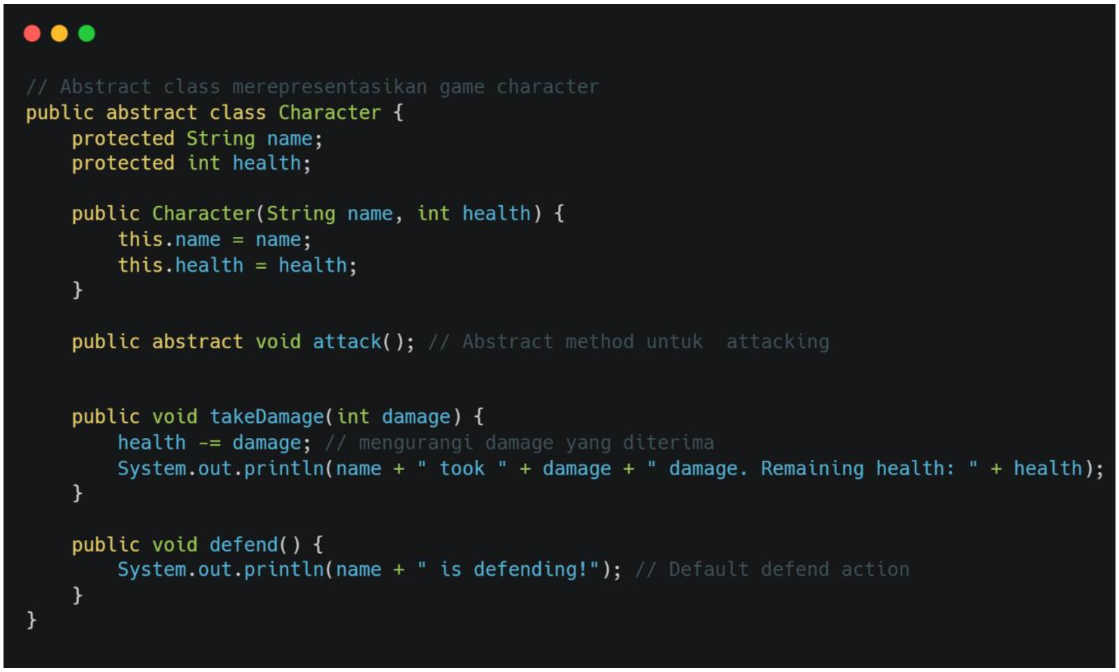
Actions.java



```java
// Interface Untuk character actions
public interface Actions {
    void attack(); // Method untuk melakukan attack
    void defend(); // Method untuk melakukan defense
}

interface Equipable {
    void equip(); // Method untuk menggunakan weapon
}
```

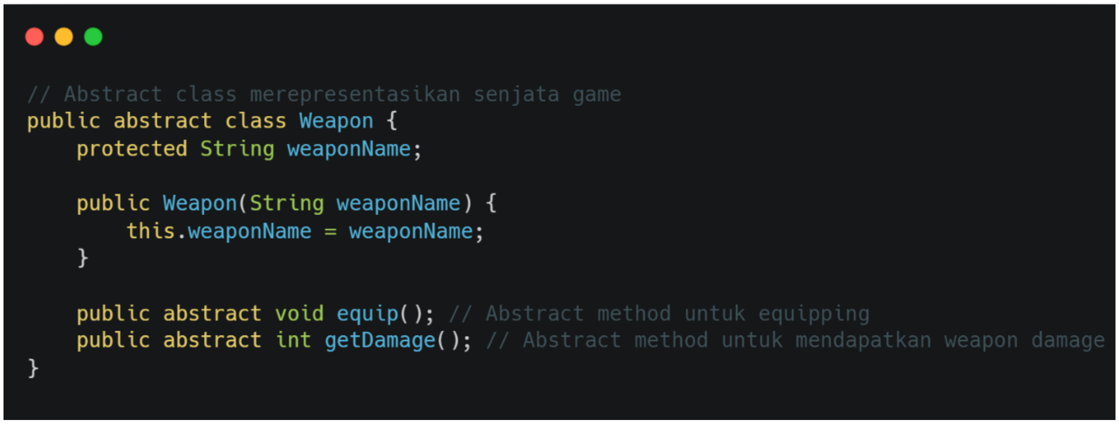
2. ABSTRACT CLASS (SuperClass)

Character.java



```java
// Abstract class merepresentasikan game character
public abstract class Character {
    protected String name;
    protected int health;

    public Character(String name, int health) {
        this.name = name;
        this.health = health;
    }

    public abstract void attack(); // Abstract method untuk  attacking


    public void takeDamage(int damage) {
        health -= damage; // mengurangi damage yang diterima
        System.out.println(name + " took " + damage + " damage. Remaining health: " + health);
    }

    public void defend() {
        System.out.println(name + " is defending!"); // Default defend action
    }
}
```

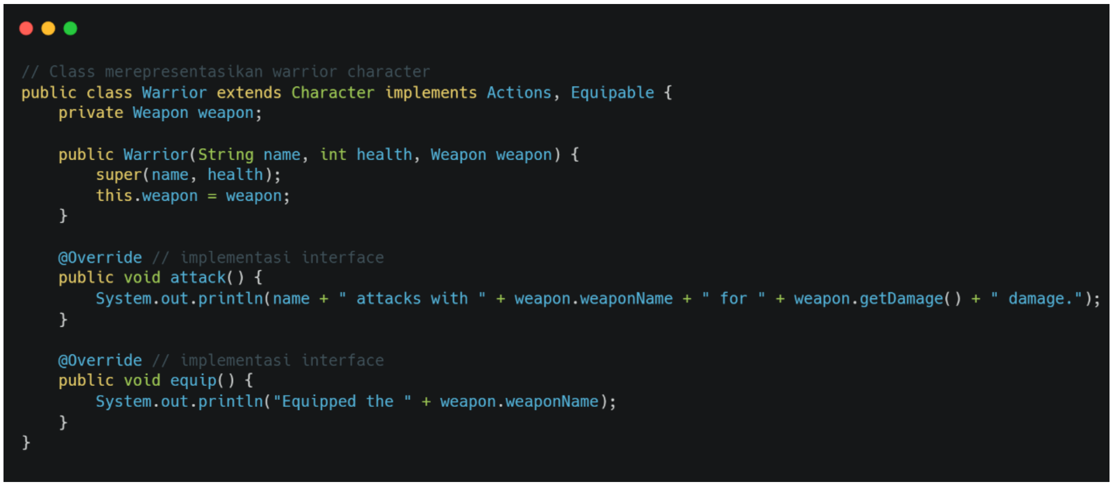
Weapon.java



```java
// Abstract class merepresentasikan senjata game
public abstract class Weapon {
    protected String weaponName;

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

    public abstract void equip(); // Abstract method untuk equipping
    public abstract int getDamage(); // Abstract method untuk mendapatkan weapon damage
}
```

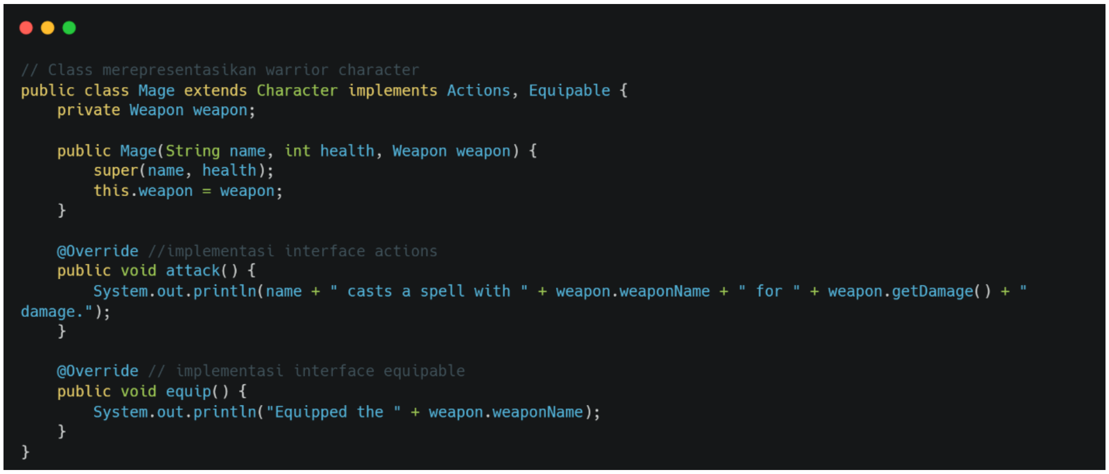
3. CONCRETE CLASS (SubClass)

Warrior.java



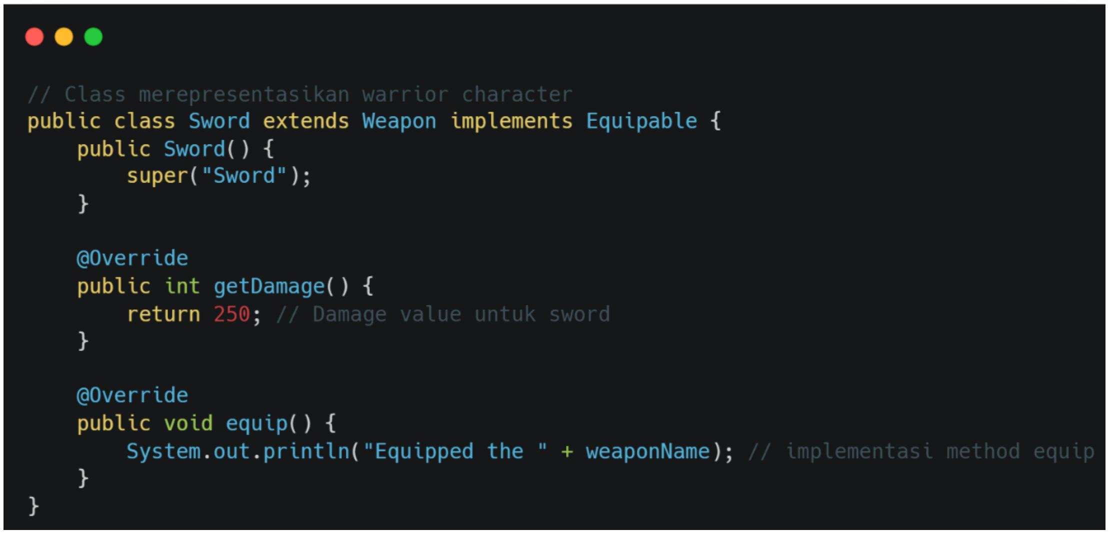
```java
// Class merepresentasikan warrior character
public class Warrior extends Character implements Actions, Equipable {
    private Weapon weapon;

    public Warrior(String name, int health, Weapon weapon) {
        super(name, health);
        this.weapon = weapon;
    }

    @Override // implementasi interface
    public void attack() {
        System.out.println(name + " attacks with " + weapon.weaponName + " for " + weapon.getDamage() + " damage.");
    }

    @Override // implementasi interface
    public void equip() {
        System.out.println("Equipped the " + weapon.weaponName);
    }
}
```

Mage.java



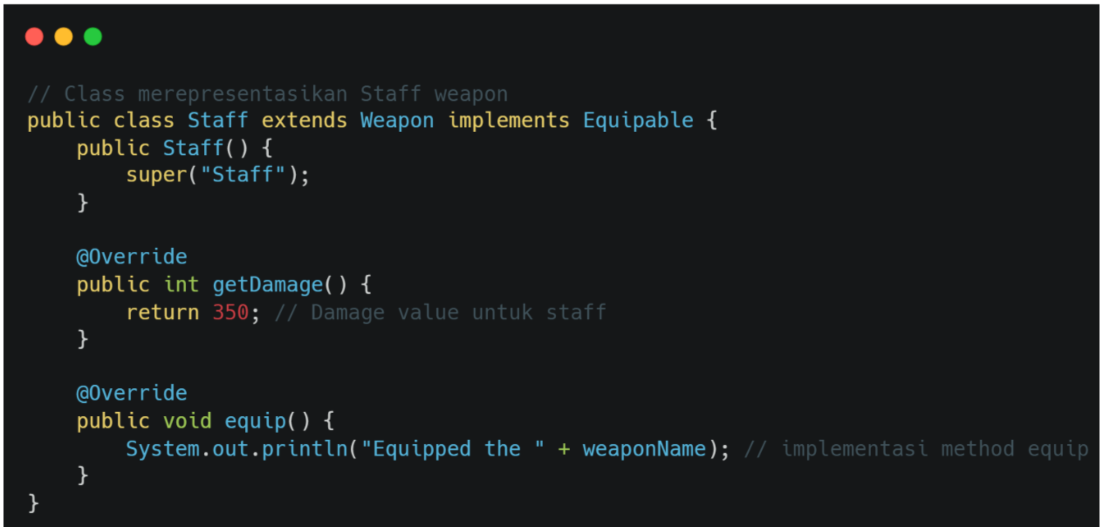
```java
// Class merepresentasikan warrior character
public class Mage extends Character implements Actions, Equipable {
    private Weapon weapon;

    public Mage(String name, int health, Weapon weapon) {
        super(name, health);
        this.weapon = weapon;
    }

    @Override //implementasi interface actions
    public void attack() {
        System.out.println(name + " casts a spell with " + weapon.weaponName + " for " + weapon.getDamage() + "
damage.");
    }

    @Override // implementasi interface equipable
    public void equip() {
        System.out.println("Equipped the " + weapon.weaponName);
    }
}
```

Sword.java



```java
// Class merepresentasikan warrior character
public class Sword extends Weapon implements Equipable {
    public Sword() {
        super("Sword");
    }

    @Override
    public int getDamage() {
        return 250; // Damage value untuk sword
    }

    @Override
    public void equip() {
        System.out.println("Equipped the " + weaponName); // implementasi method equip
    }
}
```

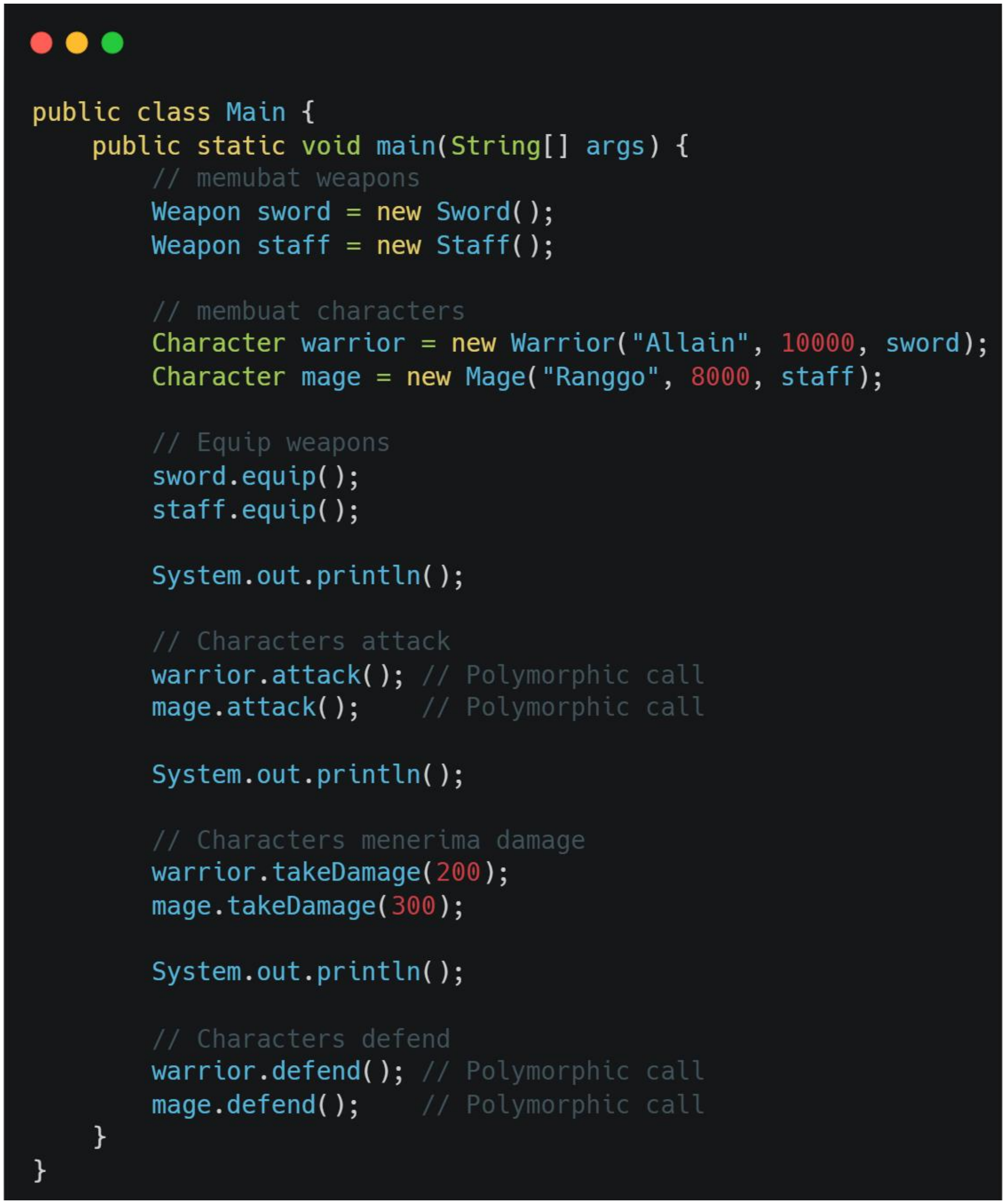
Staff.java



```java
// Class merepresentasikan Staff weapon
public class Staff extends Weapon implements Equipable {
    public Staff() {
        super("Staff");
    }

    @Override
    public int getDamage() {
        return 350; // Damage value untuk staff
    }

    @Override
    public void equip() {
        System.out.println("Equipped the " + weaponName); // implementasi method equip
    }
}
```

4. MAIN FILE

Main.java

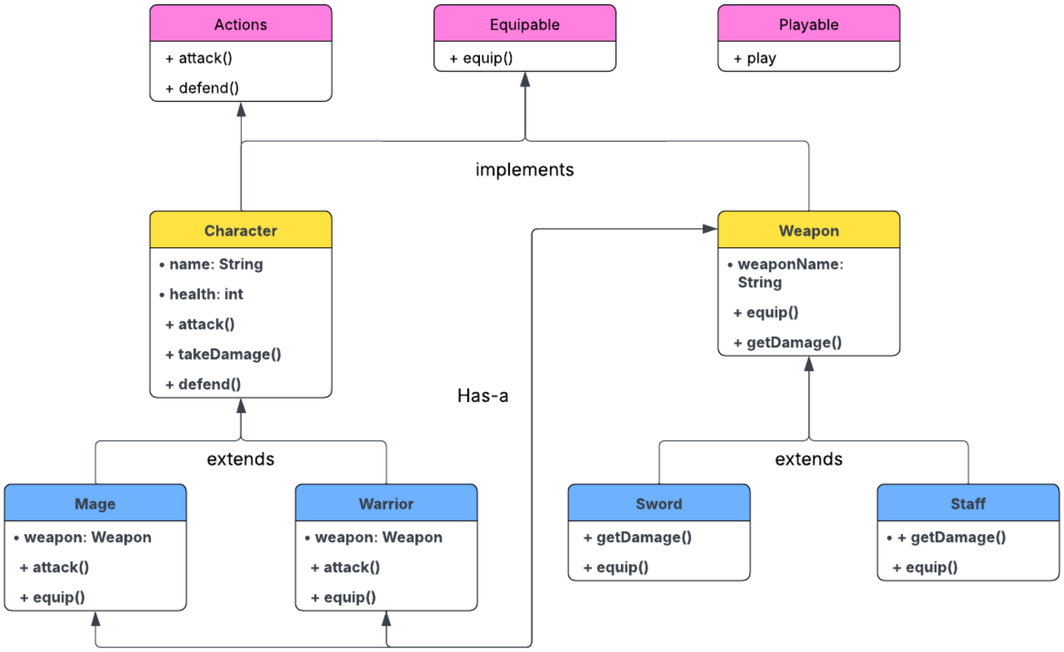


```java
public class Main {
    public static void main(String[] args) {
        // memubat weapons
        Weapon sword = new Sword();
        Weapon staff = new Staff();

        // membuat characters
        Character warrior = new Warrior("Allain", 10000, sword);
        Character mage = new Mage("Ranggo", 8000, staff);

        // Equip weapons
        sword.equip();
        staff.equip();

        System.out.println();

        // Characters attack
        warrior.attack(); // Polymorphic call
        mage.attack();    // Polymorphic call

        System.out.println();

        // Characters menerima damage
        warrior.takeDamage(200);
        mage.takeDamage(300);

        System.out.println();

        // Characters defend
        warrior.defend(); // Polymorphic call
        mage.defend();    // Polymorphic call
    }
}
```

Penjelasan Singkat

1. **Inheritance**:
   - Warrior dan Mage mewarisi Character.
   - Sword dan Staff mewarisi Weapon.
2. **Interface Implementation**:
   - Warrior, Mage, Sword, dan Staff mengimplementasikan Equipable.
   - Warrior dan Mage mengimplementasikan Actions.
3. **Polymorphism**:
   - attack() diimplementasikan di Warrior dan Mage dengan perilaku yang berbeda.
   - equip() diimplementasikan di Warrior, Mage, Sword, dan Staff dengan perilaku yang berbeda.
   - Polymorphism memungkinkan method yang sama (attack() atau equip()) memiliki banyak bentuk (perilaku) tergantung pada class yang mengimplementasikannya.
4. **Composition**:
   - Warrior memiliki Sword.
   - Mage memiliki Staff.

DIAGRAM RELATIONSHIPS


Actions
+ attack()
+ defend()
Equipable
+ equip()
Playable
+ play
implements
Character
• name: String
• health: int
+ attack()
+ takeDamage()
+ defend()
Weapon
• weaponName: String
+ equip()
+ getDamage()
Has-a
extends
Mage
• weapon: Weapon
+ attack()
+ equip()
Warrior
• weapon: Weapon
+ attack()
+ equip()
extends
Sword
+ getDamage()
+ equip()
Staff
• + getDamage()
+ equip()