**LAPORAN PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK**

**GETTER & SETTER**

Logo

Description automatically generated with medium confidence

**Disusun Oleh:**

**Rayhan Surya Destian (105222024)**

**PROGRAM STUDI ILMU KOMPUTER**

**FAKULTAS SAINS DAN KOMPUTER**

**PEMROGRAMAN BERORIENTASI OBJEK**

**UNIVERSITAS PERTAMINA**

**2023/2024**

1. **Pendahuluan**

Pembuatan game sederhana berorientasi objek ini melibatkan penggunaan empat kelas dalam bahasa pemrograman Java: Player, Weapon, Armor, dan Main. Setiap kelas memodelkan aspek tertentu dari game dimana dua pemain saling menyerang dengan bantuan armor dan senjata mereka. Game ini menunjukkan implementasi konsep OOP seperti enkapsulasi, inheritance, dan polymorphism.

1. **Variabel**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Variabel | Tipe data | Fungsi |
| 1 | name | String | Menyimpan nama dari objek Player, Weapon, atau Armor. |
| 2 | baseHealth | int | Health player |
| 3 | baseAttack | int | Attack player |
| 4 | incrementHealth | int | Nilai peningkatan health Player setiap levelup |
| 5 | incrementAttack | int | Nilai peningkatan attack Player setiap levelup |
| 6 | level | int | Level player saat ini |
| 7 | currentHealth | int | Health player saat ini |
| 8 | isAlive | boolean | Status apakah player masih hidup atau mati |
| 9 | armor | Armor | Reference ke objek Armor yg dipakai player |
| 10 | weapon | Weapon | Reference ke objek Weapon yg dipakai player |
| 11 | attack | int | Kekuatan attack Weapon |
| 12 | strength | int | Kekuatan health & defense Armor |
| 13 | health | int | Extra health dari Armor |

1. **Constructor dan Method**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Metode | Jenis Metode | Fungsi |
| 1 | Player(String) | Constructor | Inisialisasi Player dgn nilai awal |
| 2 | setArmor(Armor) | Functional | Menetapkan & update health Player dengan Armor |
| 3 | setWeapon(Weapon) | Functional | Set Weapon ke Player |
| 4 | display() | Procedural | Tampilkan detail Player |
| 5 | attack(Player) | Procedural | Melakukan serangan dan update level |
| 6 | defence(int) | Procedural | Hitung damage dan update health |
| 7 | maxHealth() | Functional | Hitung max health Player |
| 8 | Weapon(String, int) | Constructor | Buat Weapon dengan nama dan damage |
| 9 | getAttack() | Functional | Get attack strength dari Weapon |
| 10 | Armor(String, int, int) | Constructor | Buat Armor dengan nama dan atribut |
| 11 | getAddHealth() | Functional | Kalkulasi tambahan health dari Armor |
| 12 | getDefencePower() | Functional | Kalkulasi defence strength dari Armor |

1. **Dokumentasi dan Pembahasan Code**

Proyek ini melibatkan pembuatan empat kelas yang bertujuan untuk mensimulasikan pertarungan antara dua pemain yang menggunakan senjata dan armor. Class Player adalah inti dari simulasi, menyimpan atribut-atribut seperti health, attack, level, dan status hidup, serta menangani logika serangan dan pertahanan. Armor dan senjata diwakili oleh class Armor dan Weapon, yang keduanya menyediakan metode untuk meningkatkan kekuatan defence dan attack. Class Main berfungsi sebagai driver yang menginisialisasi objek-objek dan menjalankan simulasi pertarungan.

Metode attack() dan defence() dalam class Player adalah pusat interaksi antar pemain, dimana attack() meningkatkan level pemain yang menyerang dan defence() mengurangi health berdasarkan perbandingan antara kekuatan attack dan defence. Health pemain dihitung ulang setelah setiap serangan untuk mencerminkan efek dari armor yang dipakai. Proses ini berulang dalam class Main, yang mengatur alur pertarungan dan menampilkan status terkini dari setiap pemain.

1. **Kesimpulan**

Proyek ini menunjukkan penggunaan konsep pemrograman berorientasi objek dalam game sederhana, menyoroti pentingnya enkapsulasi dan interaksi antar objek. Demonstrasi melalui class Player, Weapon, dan Armor mengilustrasikan bagaimana data dan logika dapat dikelola secara efektif untuk simulasi yang koheren dan modular, memfasilitasi pengembangan dan pemeliharaan software.

1. **Daftar Pustaka**

-