

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK (PBO) – [TUGAS 2]**



Disusun Oleh

NADINE AURA RAHMADHANI 123140195

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SUMATERA**

2025

SOAL

membuat sebuah permainan sederhana tentang pertarungan Robot.

Kalian akan membuat kelas Robot yang terdiri dari beberapa properti seperti attack, Hp, dll., serta beberapa metode seperti attack_enemy() atau regen_health().

Permainan ini akan berakhir ketika salah satu robot memiliki Hp = 0.

Kalian bisa lebih kreatif dengan menambahkan konsep seperti attack_accuracy agar serangan dapat meleset dalam beberapa kesempatan atau menambahkan mekanisme skill seperti stun, silence, dll., pada musuh. (Bagian ini opsional).

Kalian mungkin perlu dua kelas:

Kelas Robot: Berisi mekanisme attack, hp, dan mekanisme pertarungan robot.

Kelas Game: Berfungsi untuk menentukan jumlah ronde serta mengatur jalannya permainan.

Kalian bisa melihat contoh cara kerja program di bawah ini. ini cuma contoh, kalian bisa buat versi kalian sendiri, yang lebih sederhana atau kompleks, tergantung kreativitas kalian.

Jawab :

```
class Robot:
    def __init__(self, nama, hp, attack):
        self.nama = nama
        self.hp = hp
        self.attack = attack

    def serang(self, musuh):
        musuh.hp -= self.attack
        print(f"{self.nama} menyerang {musuh.nama} dengan {self.attack} damage!")

class Game:
    def __init__(self, robot1, robot2):
        self.robot1 = robot1
        self.robot2 = robot2
        self.ronde = 1

    def mulai(self):
        while self.robot1.hp > 0 and self.robot2.hp > 0:
            print(f"\nRound-{self.ronde} =====")
            print(f"{self.robot1.nama} [HP: {self.robot1.hp}] vs {self.robot2.nama} [HP: {self.robot2.hp}]")

            input(f"{self.robot1.nama}, tekan ENTER untuk menyerang...")
            self.robot1.serang(self.robot2)

            if self.robot2.hp <= 0:
                print(f"\n{self.robot1.nama} menang!")
```

```

print(f"\nRound- {self.ronde} =====")
print(f"{self.robot1.nama} [HP: {self.robot1.hp}] vs {self.robot2.nama} [HP: {self.robot2.hp}]")

input(f"{self.robot1.nama}, tekan ENTER untuk menyerang...")
self.robot1.serang(self.robot2)

if self.robot2.hp <= 0:
    print(f"\n{self.robot1.nama} menang!")
    break

input(f"{self.robot2.nama}, tekan ENTER untuk menyerang...")
self.robot2.serang(self.robot1)

if self.robot1.hp <= 0:
    print(f"\n{self.robot2.nama} menang!")
    break

self.ronde += 1

robot1 = Robot("Drogon", 300, 50)
robot2 = Robot("Caraxes", 650, 30)

game = Game(robot1, robot2)
game.mulax()
```

Output Hasil (Screenshot)

Round-1
Drogon [HP: 300] vs Caraxes [HP: 650]
Drogon, tekan ENTER untuk menyerang...
Drogon menyerang Caraxes dengan 50 damage!
Caraxes, tekan ENTER untuk menyerang...
Caraxes menyerang Drogon dengan 30 damage!

Round-2
Drogon [HP: 270] vs Caraxes [HP: 600]
Drogon, tekan ENTER untuk menyerang...
Drogon menyerang Caraxes dengan 50 damage!
Caraxes, tekan ENTER untuk menyerang...
Caraxes menyerang Drogon dengan 30 damage!

Round-3
Drogon [HP: 240] vs Caraxes [HP: 550]
Drogon, tekan ENTER untuk menyerang...
Drogon menyerang Caraxes dengan 50 damage!
Caraxes, tekan ENTER untuk menyerang...
Caraxes menyerang Drogon dengan 30 damage!

Round-4
Drogon [HP: 210] vs Caraxes [HP: 500]
Drogon, tekan ENTER untuk menyerang...
Drogon menyerang Caraxes dengan 50 damage!
Caraxes, tekan ENTER untuk menyerang...

Lampiran

buatkan saya codingan membuat sebuah game tentang pertarungan robot dimana Kalian akan membuat kelas Robot yang terdiri dari beberapa properti seperti attack, Hp, dll., serta beberapa metode seperti attack_enemy() atau regen_health().
Perm. ↓ ini akan berakhir ketika salah satu

```

1  import random
2
3  class Robot:
4      def __init__(self, name, attack, hp):
5          self.name = name
6          self.attack = attack
7          self.hp = hp
8
9      def attack_enemy(self, enemy):
10         damage = random.randint(self.attack - 5, self.attack + 5)
11         enemy.hp -= damage
12         print(f"{self.name} menyerang {enemy.name} dan memberikan {damage} damage!")
13
14     def regen_health(self):
15         heal = random.randint(5, 15)
16         self.hp += heal
17         print(f"{self.name} meregenerasi {heal} HP!")
18
19     def battle(robot1, robot2):
20         round = 1
21         while robot1.hp > 0 and robot2.hp > 0:
22             print(f"\n--- Round {round} ---")
23
24             if random.choice([True, False]):
25                 robot1.attack_enemy(robot2)

```