

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



Disusun Oleh

Andryano Shevchenko Limbong      123140205

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI SUMATERA  
2025**

## Soal

### a. Input soal

## Tugas

- a. Minggu ini hanya terdiri dari 1 Problem Set. Kalian perlu membuat sebuah permainan sederhana tentang pertarungan Robot.
  - i. Kalian akan membuat kelas Robot yang terdiri dari beberapa properti seperti attack, Hp, dll., serta beberapa metode seperti attack\_enemy() atau regen\_health().
  - ii. Permainan ini akan berakhir ketika salah satu robot memiliki Hp = 0.
- b. 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).
- c. Kalian mungkin perlu dua kelas:
  - i. Kelas Robot: Berisi mekanisme attack, hp, dan mekanisme pertarungan robot.
  - ii. Kelas Game: Berfungsi untuk menentukan jumlah ronde serta mengatur jalannya permainan.
- d. 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.

```
1 Round-1 =====
2 Atreus [500|10]
3 Daedalus [750|8]
4
5 1. Attack    2. Defense    3. Giveup
6 Atreus, pilih aksi: 1
7
8 1. Attack    2. Defense    3. Giveup
9 Daedalus, pilih aksi: 1
10
11 ----- Daedalus gagal menyerang -----
12
13 Round-2 =====
14 Atreus [500|10]
15 Daedalus [666|7]
16
17 1. Attack    2. Defense    3. Giveup
18 Atreus, pilih aksi: 3
19
20 1. Attack    2. Defense    3. Giveup
21 Daedalus, pilih aksi: 1
22
23 Daedalus menang!
```

## **b. Penjelasan**

Pada Codingan yang saya buat terdapat beberapa bagian yang dapat saya jelaskan, sebagai berikut

### **1. Class and Method(Fungsi)**

- a. Class yang merupakan atribut dasar yang akan dimiliki oleh karakter pemain dan karakter musuh, dapat dilihat pada line codingan 4 - 12:
  - i. name: Nama karakter
  - ii. hp: Health Points (Nyawa)
  - iii. chakra: Sumber energi untuk jurus
  - iv. attack: Kekuatan serangan fisik
  - v. defense: Kemampuan bertahan
  - vi. speed: Penentu urutan giliran
  - vii. jutsu: Kamus berisi jurus-jurus spesial
- b. Method (Fungsi) merupakan bagian paling besar dan penting keberadaannya di dalam Turn Based Game dengan tema anime Naruto yang saya buat dengan beberapa penjelasan sebagai berikut
  - i. def is\_alive(self): return self.hp > 0 : Untuk menentukan apakah karakter mati atau tidak
  - ii. def battle(player, enemy): Untuk mengoutput proses pertarungan antara karakter pada terminal, sekaligus proses inputan dan putputan pemilihan Serangan Fisik, Jutsu atau Isi Chakra oleh pemain untuk memenangkan permainan, di dalam fungsi def battle(player, enemy) juga dapat memanggil def display\_status() dari karakter yang bertarung dan apabila karakter yang bertarung sudah kehabisan Health maka, fungsi ini akan menghentikan program dan mendeklarasikan pemenang dari pertarungan tersebut.
  - iii. def main(): Untuk mengoutput karakter apa saja yang tersedia di terminal yang berfungsi membantu user untuk memilih karakter yang ingin dimainkan, di dalam def main(): juga disimpan atribut dan karakteristik dari setiap karakter yang ada di dalam program, baik karakter yang bisa dimainkan atau karakter musuh yang tidak bisa diakses oleh user.
  - iv. def physical\_attack(): Serangan fisik dasar dengan rumus damage = attack - defense
  - v. def use\_jutsu(): Menggunakan jurus dengan pengecekan ketersediaan chakra
  - vi. def recharge\_chakra(): Mengisi 10 chakra
  - vii. def display\_status(): Menampilkan status HP dan chakra

### **2. Mekanisme Pertarungan**

- a. Berbasis Turn Based Game, yang artinya mekanik adalah sistem giliran antara karakter pemain dengan karakter musuh yang ditentukan oleh atribut speed, sehingga giliran karakter pertama kali akan ditentukan oleh codingan pada line

47 - 48.

```
while player.is_alive() and enemy.is_alive():  
    if player.speed >= enemy.speed:
```

- b. Aksi yang tersedia pada karakter ada 3, yaitu, Serangan Fisik, Jutsu (Rasengan, Chidori, Okasho dan Healing) dan Isi Chakra +10
- c. Karakter yang ada di dalam game, ada 3 karakter yang bisa dimainkan oleh user:

```
def main():  
    # Definisikan karakter  
    naruto = Character("Naruto", 100, 50, 20, 5, 10, {  
        "Rasengan": {"damage": 30, "cost": 20},  
        "Healing": {"damage": -20, "cost": 15}  
    })  
  
    sasuke = Character("Sasuke", 90, 60, 25, 5, 12, {  
        "Chidori": {"damage": 35, "cost": 25},  
        "Gokakyu": {"damage": 25, "cost": 15}  
    })  
  
    sakura = Character("Sakura", 80, 70, 15, 10, 8, {  
        "Okasho": {"damage": 25, "cost": 20},  
        "Healing": {"damage": -30, "cost": 20}  
    })  
  
    kakashi = Character("Kakashi", 85, 65, 22, 6, 11, {  
        "Chidori": {"damage": 30, "cost": 20},  
        "Gokakyu": {"damage": 25, "cost": 15}  
    })  
  
    # Pilih karakter pemain  
    print("\n=====")  
    print("                PILIH KARAKTER                ")  
    print("=====")  
    print("1: Naruto (Jutsu: Rasengan dan Healing)")  
    print("2: Sasuke (Jutsu: Chidori dan Gokakyu)")  
    print("3: Sakura (Jutsu: Okasho dan Healing)")  
    print("4: Kakashi (Jutsu: Chidori dan Gokakyu)")  
    choice = input("\nMasukkan pilihan (1-4): ")
```

- d. Ada 1 karakter khusus yang dijadikan sebagai musuh:

```
# Definisikan musuh  
enemy = Character("Orochimaru", 95, 55, 18, 7, 9, {  
    "Snake Bite": {"damage": 28, "cost": 15},  
    "Healing": {"damage": -25, "cost": 20}  
})
```

### c. Source Code

```
import random
import time

class Character:
    def __init__(self, name, hp, chakra, attack, defense, speed, jutsu):
        self.name = name
        self.hp = hp
        self.chakra = chakra
        self.attack = attack
        self.defense = defense
        self.speed = speed
        self.jutsu = jutsu

    def is_alive(self):
        return self.hp > 0

    def physical_attack(self, target):
        damage = self.attack - target.defense
        if damage < 0:
            damage = 0
        target.hp -= damage
        print(f"{self.name} menyerang {target.name} dan menyebabkan {damage} damage!")

    def use_jutsu(self, target, jutsu_name):
        if jutsu_name in self.jutsu and self.chakra >= self.jutsu[jutsu_name]['cost']:
            damage = self.jutsu[jutsu_name]['damage']
            target.hp -= damage
            self.chakra -= self.jutsu[jutsu_name]['cost']
            print(f"{self.name} menggunakan {jutsu_name} pada {target.name} dan menyebabkan {damage} damage!")
        else:
            print(f"{self.name} tidak memiliki cukup chakra untuk menggunakan {jutsu_name}!")

    def recharge_chakra(self):
        self.chakra += 10
        print(f"{self.name} mengisi chakra! Chakra sekarang: {self.chakra}")

    def display_status(self):
        print(f"{self.name} - HP: {self.hp}, Chakra: {self.chakra}")
        print("-----")

def battle(player, enemy):
    print("\n=====")
    print("          PERTEMPURAN DIMULAI          ")
    print("-----\n")
    time.sleep(1)

    while player.is_alive() and enemy.is_alive():
        if player.speed >= enemy.speed:
            # Giliran pemain
            print("\n----- GILIRAN PEMAIN -----")
            action = input("Pilih aksi (1: Serangan Fisik, 2: Jutsu, 3: Isi Chakra): ")

            if action == '1':
                player.physical_attack(enemy)
            elif action == '2':
                jutsu_list = ", ".join(player.jutsu.keys())
                jutsu_name = input(f"Pilih jutsu ({jutsu_list}): ").strip()
                player.use_jutsu(enemy, jutsu_name)
            elif action == '3':
                player.recharge_chakra()
            else:
                print("\nAksi tidak valid!")

            # Giliran musuh jika masih hidup
            if enemy.is_alive():
                print("\n----- GILIRAN MUSUH -----")
                enemy.physical_attack(player)
            else:
                break
        else:
            # Giliran musuh lebih dulu
            print("\n----- GILIRAN MUSUH -----")
            enemy.physical_attack(player)
```

```

# Giliran pemain jika masih hidup
if player.is_alive():
    print("\n----- GILIRAN PEMAIN -----")
    action = input("Pilih aksi (1: Serangan Fisik, 2: Jutsu, 3: Isi Chakra): ")

    if action == '1':
        player.physical_attack(enemy)
    elif action == '2':
        jutsu_list = ', '.join(player.jutsu.keys())
        jutsu_name = input(f"Pilih jutsu ({jutsu_list}): ").strip()
        player.use_jutsu(enemy, jutsu_name)
    elif action == '3':
        player.recharge_chakra()
    else:
        print("\nAksi tidak valid!")

else:
    break

# Menampilkan status setelah satu ronde
print("\n----- STATUS -----")
player.display_status()
enemy.display_status()
print("-----")
time.sleep(1)

# Hasil akhir pertempuran
print("\n-----")
if player.is_alive():
    print(f"        {player.name} MENANG!      ")
else:
    print(f"        {enemy.name} MENANG!      ")
print("-----\n")

def main():
    # Definisikan karakter
    naruto = Character('Naruto', 100, 50, 20, 5, 10, {
        "Rasengan": {"damage": 30, "cost": 20},
        "Healing": {"damage": -20, "cost": 15}
    })

    sasuke = Character('Sasuke', 90, 60, 25, 5, 12, {
        "Chidori": {"damage": 35, "cost": 25},
        "Gokakkyu": {"damage": 25, "cost": 15}
    })

    sakura = Character('Sakura', 80, 70, 15, 10, 8, {
        "Okasho": {"damage": 25, "cost": 20},
        "Healing": {"damage": -30, "cost": 20}
    })

    kakashi = Character('Kakashi', 85, 65, 22, 6, 11, {
        "Chidori": {"damage": 30, "cost": 20},
        "Gokakkyu": {"damage": 25, "cost": 15}
    })

    # Pilih karakter pemain
    print("\n-----")
    print("          PILIH KARAKTER          ")
    print("-----")
    print("1: Naruto (Jutsu: Rasengan dan Healing)")
    print("2: Sasuke (Jutsu: Chidori dan Gokakkyu)")
    print("3: Sakura (Jutsu: Okasho dan Healing)")
    print("4: Kakashi (Jutsu: Chidori dan Gokakkyu)")
    choice = input("\nMasukkan pilihan (1-4): ")

    if choice == '1':
        player = naruto
    elif choice == '2':
        player = sasuke
    elif choice == '3':
        player = sakura
    elif choice == '4':
        player = kakashi
    else:
        print("\nPilihan tidak valid! Menggunakan Naruto sebagai default.")
        player = naruto

    # Definisikan musuh
    enemy = Character('Orochimaru', 95, 55, 18, 7, 9, {
        "Snake Bite": {"damage": 20, "cost": 15},
        "Healing": {"damage": -25, "cost": 20}
    })

    print(f"\n-----")
    print(f"        {player.name} vs {enemy.name}      ")
    print("-----\n")
    time.sleep(1)

    # Mulai pertarungan
    battle(player, enemy)

if __name__ == "__main__":
    main()

```

d. Hasil

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS D:\VSCode\Phyton> python -u "d:\VSCode\Phyton\tempCodeRunnerFile.py"

=====

PILIH KARAKTER

=====

1: Naruto (Jutsu: Rasengan dan Healing)  
2: Sasuke (Jutsu: Chidori dan Gokakyu)  
3: Sakura (Jutsu: Okasho dan Healing)  
4: Kakashi (Jutsu: Chidori dan Gokakyu)

Masukkan pilihan (1-4): 4

=====

Kakashi vs Orochimaru

=====

=====

PERTEMPURAN DIMULAI

=====

===== GILIRAN PEMAIN =====

Pilih aksi (1: Serangan Fisik, 2: Jutsu, 3: Isi Chakra): 2

Pilih jutsu (Chidori, Gokakyu): Chidori

Kakashi menggunakan Chidori pada Orochimaru dan menyebabkan 30 damage!

===== GILIRAN MUSUH =====

Orochimaru menyerang Kakashi dan menyebabkan 12 damage!

===== STATUS =====

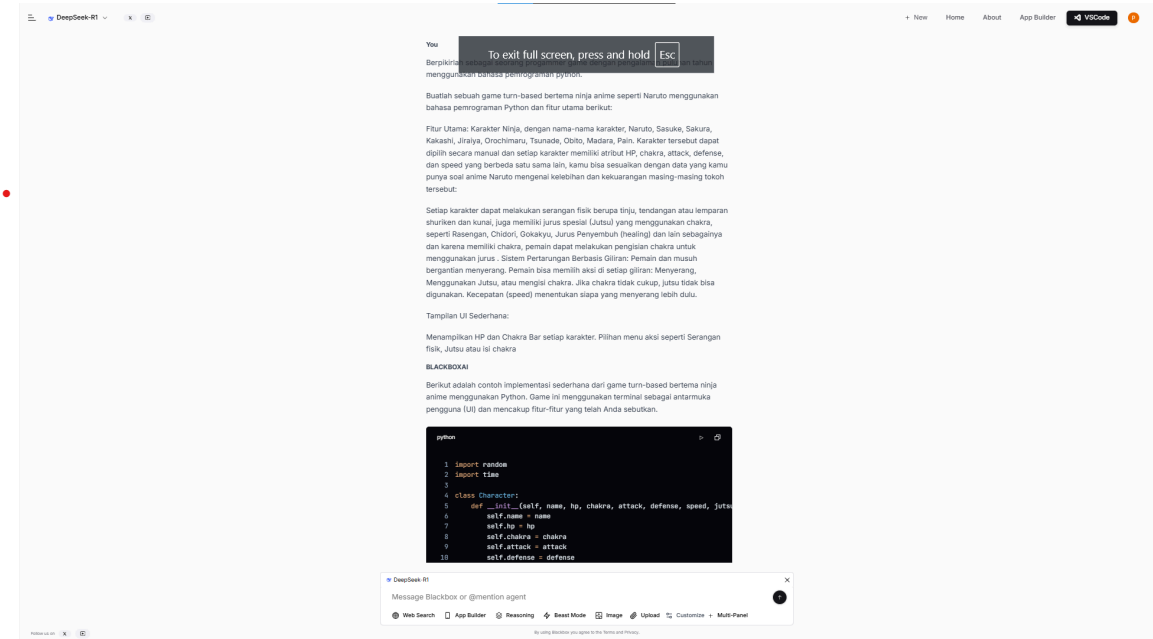
Kakashi - HP: 73, Chakra: 45

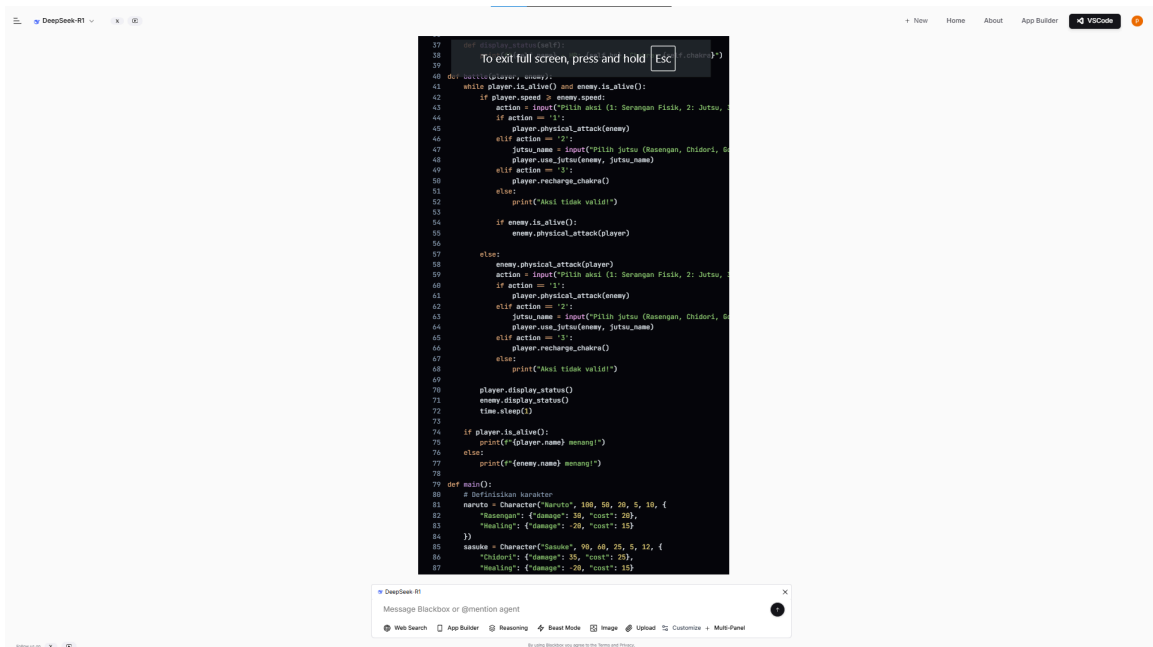
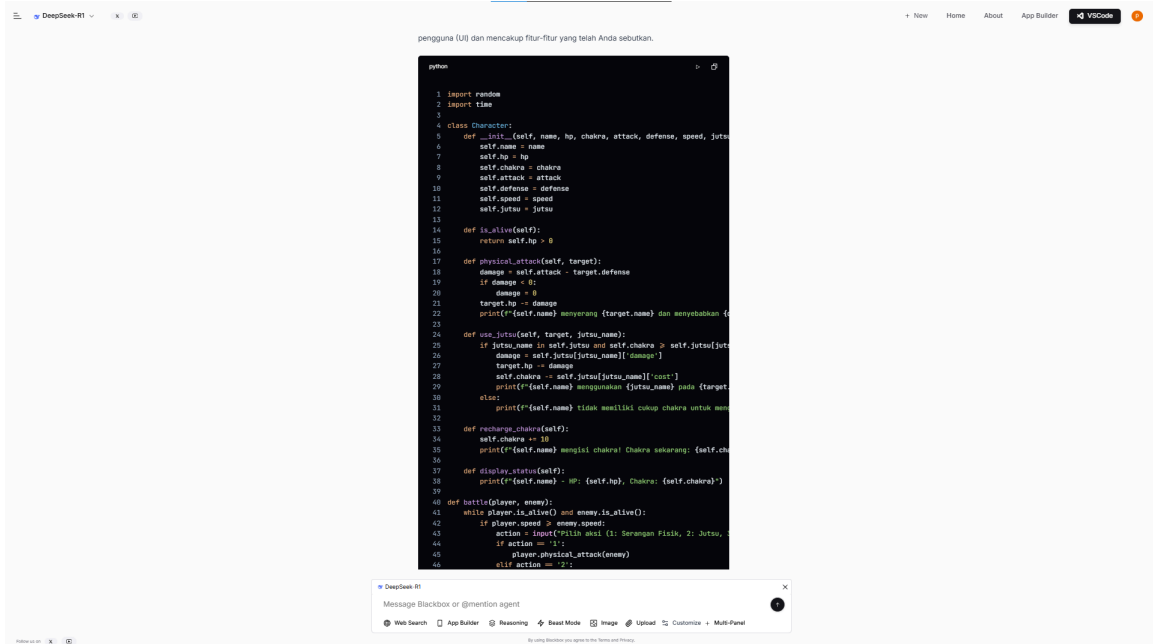
-----

Orochimaru - HP: 65, Chakra: 55

-----

Lampiran







SELECT MODELS

+ NewHomeAboutApp BuilderVS Code

```
87     "healing": {"damage": -20, "cost": 15}
88 })
89 sakura = Character("Sakura", 80, 70, 15, 10, 8, {
90     "chakra": {"damage": 25, "cost": 20},
91     "healing": {"damage": -30, "cost": 20}
92 })
93 kakashi = Character("Kakashi", 85, 65, 22, 4, 11, {
94     "chakra": {"damage": 30, "cost": 20},
95     "healing": {"damage": -20, "cost": 15}
96 })
97
98 # Pilih karakter pemain
99 print("Pilih karakter:")
100 print("1: Naruto")
101 print("2: Sasuke")
102 print("3: Sakura")
103 print("4: Kakashi")
104 choice = input("Masukkan pilihan (1-4): ")
105
106 if choice == '1':
107     player =
```

You

continue

**BLACKBOXAI**

Tentu! Mari kita lanjutkan dengan kode yang telah kita mulai. Di bawah ini adalah kelanjutan dari fungsi "main()" dan penutupan program:

```
python
1     player = naruto
2     elif choice == '2':
3         player = sasuke
4     elif choice == '3':
5         player = sakura
6     elif choice == '4':
7         player = kakashi
8     else:
9         print("Pilihan tidak valid! Menggunakan Naruto sebagai default.")
10        player = naruto
11
12 # Definisi musuh
13 enemy = Character("Orochimaru", 95, 55, 10, 7, 9, {
14     "chakra": {"damage": 25, "cost": 15},
15     "healing": {"damage": -25, "cost": 20}
16 })
17
18 print(f"({player.name}) vs ({enemy.name})")
```

Message Blackbox or @mention agent

Web SearchApp BuilderReasoningBest ModeImagesUploadCustomizeMulti-Panel

By using Blackbox you agree to the Terms and Privacy

SELECT MODELS

+ NewHomeAboutApp BuilderVS Code

```
12 # Definisi musuh
13 enemy = Character("Orochimaru", 95, 55, 10, 7, 9, {
14     "chakra": {"damage": 25, "cost": 15},
15     "healing": {"damage": -25, "cost": 20}
16 })
17
18 print(f"({player.name}) vs ({enemy.name})")
19 time.sleep(1)
20
21 # Mulai pertarungan
22 battle(player, enemy)
23
24 if __name__ == "__main__":
25     main()
```

**Penjelasan Kode:**

- Karakter:** Kami mendefinisikan beberapa karakter dengan atribut yang berbeda. Setiap karakter memiliki HP, chakra, attack, defense, speed, dan jutsu yang berbeda.
- Sistem Pertarungan:** Dalam fungsi "battle", pemain dan musuh bergantian menyerang berdasarkan kecepatan. Pemain dapat memilih untuk menyerang secara fisik, menggunakan jutsu, atau mengisr chakra.
- Jutsu:** Setiap jutsu memiliki biaya chakra dan damage yang berbeda. Jika chakra tidak cukup, jutsu tidak dapat digunakan.
- Status Karakter:** Status HP dan chakra ditampilkan setelah setiap giliran.
- Antarmuka Pengguna:** Pengguna dapat memilih karakter dan aksi melalui input terminal.

**Cara Menjalankan:**

- Pastikan Anda memiliki Python terinstal di komputer Anda.
- Salin kode di atas ke dalam file Python (misalnya "ninja\_game.py").
- Jalankan file tersebut menggunakan terminal atau command prompt dengan perintah "python ninja\_game.py".

**Catatan:**

- Anda dapat menambahkan lebih banyak karakter dan jutsu sesuai keinginan.
- Anda juga dapat memperluas logika permainan dengan menambahkan lebih banyak fitur, seperti item, level, atau sistem pengalaman.
- Game ini adalah contoh dasar dan dapat dikembangkan lebih lanjut untuk menambahkan fitur-fitur yang lebih kompleks.

Message Blackbox or @mention agent

Web SearchApp BuilderReasoningBest ModeImagesUploadCustomizeMulti-Panel

By using Blackbox you agree to the Terms and Privacy



SELECT MODELS

+ New

Home

About

App Builder

VSCode

```
44 player.display_status()
45 enemy.display_status()
46 print("=====")
47 time.sleep(1)
48
49 # Hasil akhir pertempuran
50 print("la=====")
51 if player.is_alive():
52     print(f"      (player.name) REMANG!")
53 else:
54     print(f"      (enemy.name) REMANG!")
55 print("la=====")

def main(): # Definiskan karakter naruto = Character("Naruto", 100, 50, 20, 5, 10, {
    "Rasengan": ("damage": 30, "cost": 20), "Healing": ("damage": -20, "cost": 15) })
sasuke = Character("Sasuke", 80, 60, 25, 5, 12, {"Chidori": ("damage": 35, "cost": 25),
    "Gokakyu": ("damage": 25, "cost": 15) }) sakura = Character("Sakura", 80, 70, 15, 10, 8,
    {"Okasho": ("damage": 25, "cost": 20), "Healing": ("damage": -30, "cost": 20) })
kakashi = Character("Kakashi", 85, 85, 22, 6, 11, {"Chidori": ("damage": 30, "cost": 20),
    "Gokakyu": ("damage": 25, "cost": 15) })

1 # Pilih karakter pemain
2 print("la=====")
3 print("      PILIH KARAKTER")
4 print("=====")
5 print("1: Naruto (Jutsu: Rasengan dan Healing)")
6 print("2: Sasuke (Jutsu: Chidori dan Gokakyu)")
7 print("3: Sakura (Jutsu: Okasho dan Healing)")
8 print("4: Kakashi (Jutsu: Chidori dan Gokakyu)")
9 choice = input("Masukkan pilihan (1-4): ")
10
11 if choice == '1':
12     player = naruto
13 elif choice == '2':
14     player = sasuke
15 elif choice == '3':
16     player = sakura
17 elif choice == '4':
18     player = kakashi
19 else:
20     print("Pilihan tidak valid! Menggunakan Naruto sebagai default.")
21     player = naruto
22
23 # Definiskan musuh
24 enemy = Character("Orochimaru", 80, 65, 18, 7, 9, {
25     "Sake Bile": ("damage": 28, "cost": 15),
26     "Healing": ("damage": -25, "cost": 20) })

Message Blackbox or @mention agent
Web Search App Builder Reasoning Best Mode Image Upload Customize Multi-Panel
```

Footer on 10

By using Blackbox you agree to the Terms and Privacy

SELECT MODELS

+ New

Home

About

App Builder

VSCode

```
26 "Healing": ("damage": -25, "cost": 20)
27 })
28
29 print(f"la=====")
30 print(f"      (player.name) vs (enemy.name) ")
31 print("=====la")
32 time.sleep(1)
33
34 # Mulai pertarungan
35 battle(player, enemy)

if name == "main": main()

perbaiki code di atas sehingga tidak mengeluarkan output:

=====
===== PILIH KARAKTER
=====
1: Naruto (Jutsu: Rasengan dan Healing) 2: Sasuke (Jutsu: Chidori dan Gokakyu) 3:
Sakura (Jutsu: Okasho dan Healing)
4: Kakashi (Jutsu: Chidori dan Gokakyu)

Masukkan pilihan (1-4): 3

=====
===== Sakura vs
Orochimaru
=====
===== PERTEMPURAN
DIMULAI

Orochimaru menyerang Sakura dan menyebabkan 8 damage!

Pilih aksi (1: Serangan Fisik, 2: Jutsu, 3: Isi Chakra): 2

Pilih jutsu: Okasho

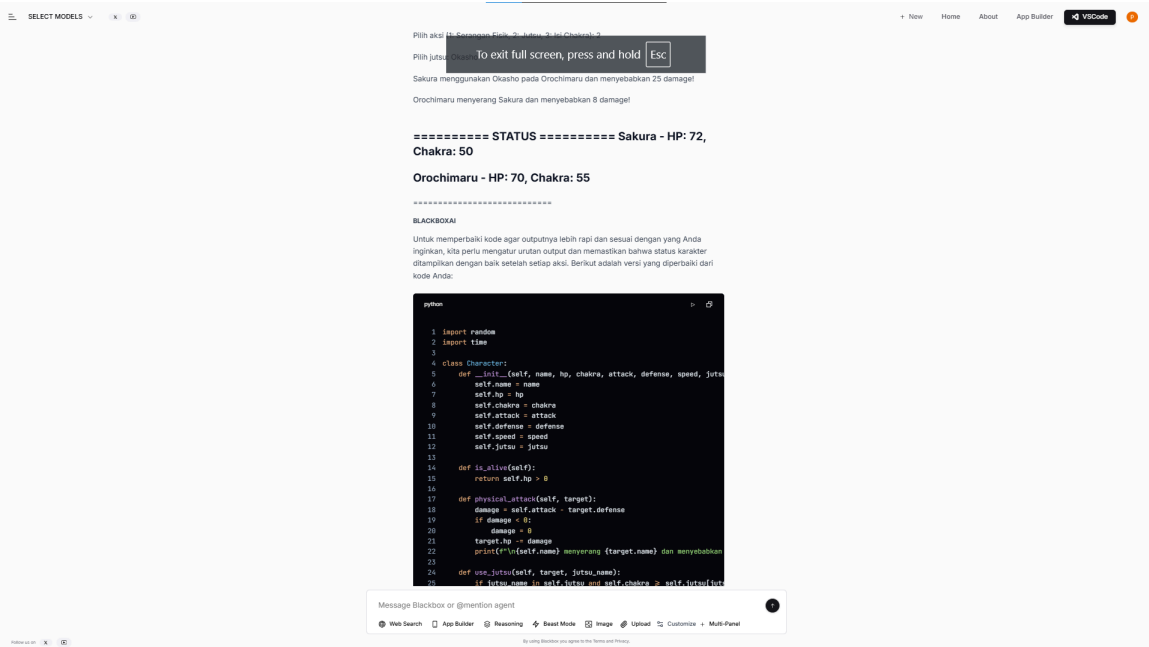
Sakura menggunakan Okasho pada Orochimaru dan menyebabkan 25 damage!

===== STATUS ===== Sakura - HP: 72,
Chakra: 50

Message Blackbox or @mention agent
Web Search App Builder Reasoning Best Mode Image Upload Customize Multi-Panel
```

Footer on 10

By using Blackbox you agree to the Terms and Privacy



SELECT MODELS

+ New

Home

About

App Builder

VSCODE

```
24 def use_jutsu(self, target, jutsu_name):
25     if jutsu_name in self.jutsu and self.charge > self.jutsu[jutsu_name]['damage']:
26         self.charge -= self.jutsu[jutsu_name]['damage']
27         target.hp -= damage
28         self.charge = self.jutsu[jutsu_name]['cost']
29         print(f"({self.name}) menggunakan {jutsu_name} pada (target)
30     else:
31         print(f"({self.name}) tidak memiliki cukup charge untuk =
32
33 def recharge_charge(self):
34     self.charge = 10
35     print(f"({self.name}) mengisi charge! Charge sekarang: {self.charge}
36
37 def display_status(self):
38     print(f"({self.name}) - HP: {self.hp}, Charge: {self.charge}
39     print("-----")
40
41 def battle(player, enemy):
42     print("=====")
43     print("          PERTEMPURAN DIMULAI          ")
44     print("=====")
45     time.sleep(1)
46
47 while player.is_alive() and enemy.is_alive():
48     if player.speed > enemy.speed:
49         # Siliran pemain
50         action = input("Pilih aksi (1: Serangan Fisik, 2: Jutsu, 3: Recharge): ")
51
52         if action == '1':
53             player.physical_attack(enemy)
54         elif action == '2':
55             jutsu_name = input("Pilih jutsu: ")
56             player.use_jutsu(enemy, jutsu_name)
57         elif action == '3':
58             player.recharge_charge()
59         else:
60             print("Pilih tidak valid!")
61
62         # Siliran musuh jika masih hidup
63         if enemy.is_alive():
64             enemy.physical_attack(player)
65         else:
66             break
67
68     else:
69         # Siliran musuh lebih dulu
70         enemy.physical_attack(player)
71
72         # Siliran pemain jika masih hidup
73         if player.is_alive():
74             action = input("Pilih aksi (1: Serangan Fisik, 2: Jutsu, 3: Recharge): ")
75             if action == '1':
76                 player.physical_attack(enemy)
77             elif action == '2':
78                 jutsu_name = input("Pilih jutsu: ")
79                 player.use_jutsu(enemy, jutsu_name)
80             elif action == '3':
81                 player.recharge_charge()
82             else:
83                 print("Pilih tidak valid!")
84
85             # Menampilkan status setelah satu ronde
86             print("===== STATUS =====")
87             player.display_status()
88             enemy.display_status()
89             print("=====")
90             time.sleep(1)
91
92         # Hasil akhir pertempuran
93         print("=====")
94         if player.is_alive():
95             print(f"({player.name}) MENANG!")
96         else:
97             print(f"({enemy.name}) MENANG!")
98         print("=====")
99
100 def main():
101     # Definisi karakter
102     naruto = Character("Naruto", 100, 50, 20, 5, 10, {
103         "Rasengan": {"damage": 30, "cost": 20},
104         "Mending": {"damage": 20, "cost": 10}
105     })
106     sasuke = Character("Sasuke", 90, 40, 25, 5, 12, {
107         "Chidori": {"damage": 35, "cost": 20},
108         "Sukunagi": {"damage": 25, "cost": 10}
109     })
110     sakura = Character("Sakura", 80, 70, 15, 10, 8, {
111         "Danzou": {"damage": 25, "cost": 10}
112     })
113
114     # Pertempuran
115     battle(naruto, sakura)
116     battle(sasuke, sakura)
117
118 if __name__ == '__main__':
119     main()
```

Message Blackbox or @mention agent

Web Search

App Builder

Reasoning

Best Mode

Image

Upload

Customize

Multi-Panel

By using Blackbox you agree to the Terms and Privacy

SELECT MODELS

+ New

Home

About

App Builder

VSCODE

```
76         if action == '1':
77             player.physical_attack(enemy)
78         elif action == '2':
79             jutsu_name = input("Pilih jutsu: ")
80             player.use_jutsu(enemy, jutsu_name)
81         elif action == '3':
82             player.recharge_charge()
83         else:
84             print("Pilih tidak valid!")
85
86         # Menampilkan status setelah satu ronde
87         print("===== STATUS =====")
88         player.display_status()
89         enemy.display_status()
90         print("=====")
91         time.sleep(1)
92
93         # Hasil akhir pertempuran
94         print("=====")
95         if player.is_alive():
96             print(f"({player.name}) MENANG!")
97         else:
98             print(f"({enemy.name}) MENANG!")
99         print("=====")
100
101 def main():
102     # Definisi karakter
103     naruto = Character("Naruto", 100, 50, 20, 5, 10, {
104         "Rasengan": {"damage": 30, "cost": 20},
105         "Mending": {"damage": 20, "cost": 10}
106     })
107     sasuke = Character("Sasuke", 90, 40, 25, 5, 12, {
108         "Chidori": {"damage": 35, "cost": 20},
109         "Sukunagi": {"damage": 25, "cost": 10}
110     })
111     sakura = Character("Sakura", 80, 70, 15, 10, 8, {
112         "Danzou": {"damage": 25, "cost": 10}
113     })
114
115     # Pertempuran
116     battle(naruto, sakura)
117     battle(sasuke, sakura)
118
119 if __name__ == '__main__':
120     main()
```

You

selesaikan codingan di atas

BLACKBOXAI

Tentu! Berikut adalah kelanjutan dan penyelesaian dari kode yang telah Anda mulai. Saya akan melengkapi bagian yang hilang dan memastikan bahwa semua karakter dan logika permainan berfungsi dengan baik.

python

Message Blackbox or @mention agent

Web Search

App Builder

Reasoning

Best Mode

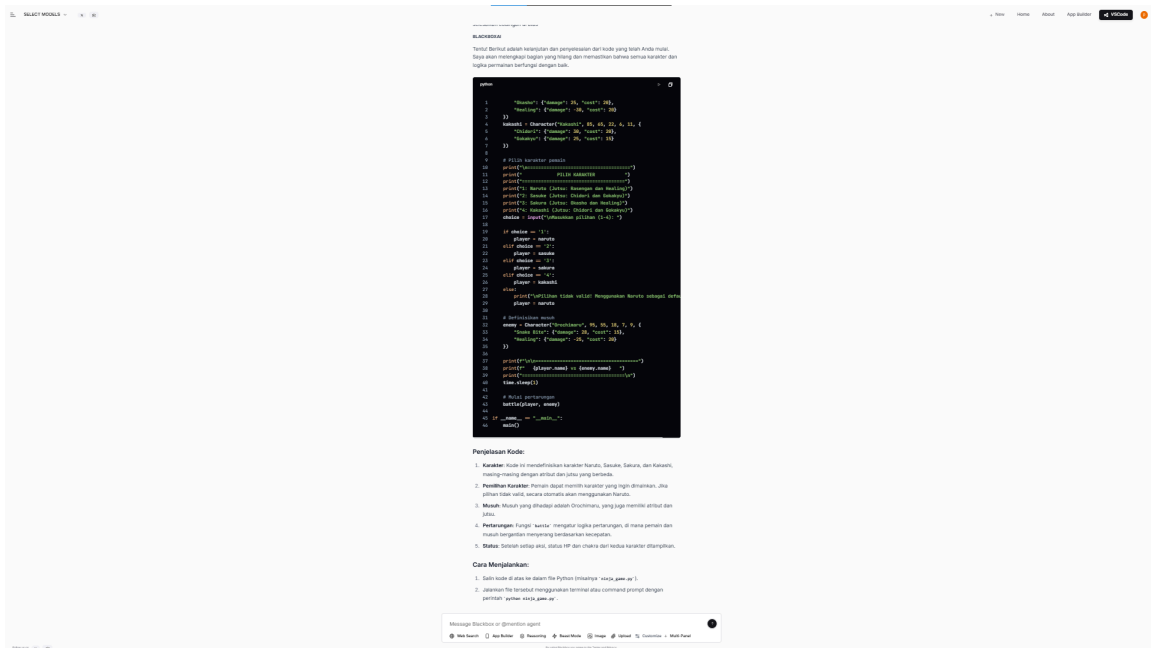
Image

Upload

Customize

Multi-Panel

By using Blackbox you agree to the Terms and Privacy



(Jika menggunakan LLM atau Referensi website dalam pembuatan laporan, baik untuk generate code ataupun penulisan text silahkan sertakan dokumentasinya bisa berupa link ataupun screenshot percakapan. Jika tidak melampirkan dan ketahuan menggunakan nilai = 0)

Contoh :

1. [Link Percakapan LLM](#)
2. [Web Referensi - DuniaIlkom](#)