

Fantasi Terakhir

Lab 9 - Dasar-Dasar Pemrograman 1 Gasal 2020/2021



Source: Final Fantasy VII (1997) © Square Enix

Cerita ini hanya fiktif belaka. Jika ada kesamaan nama tokoh, tempat kejadian ataupun cerita, itu adalah kebetulan semata dan tidak ada unsur kesengajaan.

Setelah senang bisa melakukan *speedrun* game Meong Brosss, Abi akhirnya ingin mencari game baru. Abi membeli konsol baru bernama Pacilstation 5 dengan game RPG yang sedang trending waktu itu bernama “Fantasi Terakhir”. Game ini merupakan game turn-based RPG dengan grafik yang sangat keren. Namun, Abi masih pemula dalam memainkan turn-based RPG sehingga ia selalu kalah. Abi pun akhirnya bingung dan menyerah memainkannya.

Tiba-tiba, Abi teringat temannya yang membantu dia melakukan *speedrun* Meong Bross waktu itu yang bernama DekDepe. Sayangnya, laptop DekDepe rusak dan DekDepe tidak bisa membantunya. DekDepe memberi kontak salah satu temannya yang bernama Izuri yang memiliki toko boba yang sukses. Konon, Izuri mempunyai seorang *programmer* berinisial “S” yang dapat memecahkan masalah tersebut.

Task: fantasi_terakhir (Poin: 100)

Silahkan lengkapi template yang telah diberikan dengan aturan sebagai berikut:

1. Kalian wajib menggunakan template. *Download template* tersebut [di sini](#).
2. Fantasi Terakhir memiliki 3 *class* yaitu **Warrior**, **Tank**, **Support** dan ketiganya merupakan *child class* dari *class* **Hero**.
3. Lengkapi method *constructor*, *setter*, dan *getter* untuk masing-masing *class*. Terdapat beberapa catatan khusus sebagai berikut:
 - a. Untuk setiap *constructor* yang dimiliki *child class* kalian wajib memanggil *constructor parent class*-nya.
HINT: Gunakan method **super()**
 - b. Untuk method **set_hp()** kalian tidak bisa mengubah **hp** menjadi negatif. Jika method tersebut dipanggil menggunakan argumen bilangan negatif maka **hp** akan diubah menjadi **0**.
4. Setiap *Child Class* memiliki **atribut_khusus** sebagai berikut:

Class	atribut_khusus	Fungsi
Warrior	extra_attack	Ketika melakukan query ATTACK, karakter akan melakukan damage sebesar attack + extra_attack kepada karakter yang di-attack. Apabila atribut_khusus bernilai DEFAULT maka extra_attack diset menjadi 20.
Tank	shield	Ketika terkena query ATTACK, maka shield karakter dahulu yang berkurang. Jika shield sudah 0, maka baru HP karakter yang akan berkurang. Apabila atribut_khusus bernilai DEFAULT maka shield diset menjadi 20.
Support	heal_amount	Karakter dapat melakukan query HEAL, karakter akan menambahkan HP sebesar heal_amount kepada karakter yang di-heal. Apabila atribut_khusus bernilai DEFAULT maka heal_amount diset menjadi 20.

Alur Program:

1. Pertama, program Anda akan meminta masukan **banyaknya karakter** (Contoh interaksi dapat dilihat setelah penjelasan alur program). Kemudian Anda perlu memasukkan setiap data karakter dengan format sebagai berikut:

<TIPE> <NAMA> <HP> <ATTACK> <ATRIBUT_KHUSUS>

Keterangan:

- **<TIPE>** yang diberikan dijamin merupakan salah satu di antara **WARRIOR**, **SUPPORT**, dan **TANK**.
 - **<NAMA>** yang diberikan dijamin hanya berisi **alfabet** yang ditulis dengan **huruf kapital**.
 - **<HP>** yang diberikan merupakan bilangan bulat positif.
 - **<ATTACK>** yang diberikan merupakan bilangan bulat positif.
 - **<ATRIBUT_KHUSUS>** yang diberikan merupakan sebuah bilangan bulat positif atau dapat juga berupa string **DEFAULT**. Jika berupa string **DEFAULT** maka nilai atribut khususnya di-set sebesar **20**.
 - Seluruh karakter yang ditambahkan tersebut akan disimpan ke dalam suatu list.
2. Program akan terus meminta query/perintah yang akan dilaksanakan. Terdapat tiga macam query sebagai berikut:

- **ATTACK KARAKTER1 KARAKTER2**
KARAKTER1 akan menyerang KARAKTER2 sebanyak nilai atribut **attack** (Khusus untuk class Warrior maka ditambah dengan nilai **extra_attack**). Apabila **hp** KARAKTER2 setelah diserang habis atau **0** maka KARAKTER2 akan dihapus dari list karakter.
- **HEAL KARAKTER1 KARAKTER2**
KARAKTER1 akan melakukan heal kepada KARAKTER2 sehingga hp dari KARAKTER2 akan bertambah sebanyak nilai atribut **heal** KARAKTER 1. Apabila KARAKTER1 bukan **Support** maka program tidak akan melakukan query tersebut. Selain itu, karakter tidak dapat melakukan heal ke **dirinya sendiri**.
- **GABUNGAN KARAKTER1 KARAKTER2**
KARAKTER1 dan KARAKTER2 akan bergabung menjadi karakter yang baru bernama KARAKTER1_KARAKTER2 dengan atribut baru merupakan hasil penjumlahan atribut-atribut KARAKTER1 dan KARAKTER2. KARAKTER1 dan KARAKTER2 harus merupakan tipe yang sama.
- **EXIT**
Keluar dari program

Apabila nama karakter yang dimasukkan tidak terdapat dalam list karakter atau query tidak memenuhi syarat maka query tersebut tidak diproses dan akan dicetak output sesuai contoh interaksi. Selain itu, apabila query yang dimasukkan tidak termasuk dalam daftar query tersebut maka query tersebut juga tidak akan dianggap.

Apabila user memasukkan perintah **EXIT** maka program berhenti dan dicetak seluruh karakter yang masih hidup dengan format tabel seperti contoh di bawah ini.

Karakter yang masih hidup:

Nama	Tipe	HP

SNOW	Tank	190
GLADIO	Tank	125
TIFA_CLOUD	Warrior	135

Contoh Interaksi:

Untuk memudahkan dalam melakukan *testing*, seluruh input (tanpa *prompt*) untuk contoh interaksi di bawah dapat dilihat [di sini](#).

Contoh Interaksi 1:

Input:

```
Masukkan jumlah hero : 4
Masukkan detail hero : SUPPORT AERITH 100 15 20
Masukkan detail hero : SUPPORT TIFA 110 10 15
Masukkan detail hero : WARRIOR CLOUD 120 30 5
Masukkan detail hero : TANK BARRET 150 5 10
Masukkan perintah : HEAL AERITH CLOUD
AERITH berhasil meng-heal CLOUD
Nyawa CLOUD menjadi 140
Masukkan perintah : ATTACK CLOUD BARRET
CLOUD berhasil menyerang BARRET
Nyawa BARRET tersisa 125
Masukkan perintah : EXIT
```

Output:

```
Karakter yang masih hidup:
-----
Nama                | Tipe        | HP
-----
AERITH              | Support     | 100
TIFA                 | Support     | 110
CLOUD                | Warrior     | 140
BARRET              | Tank        | 125
```

Contoh Interaksi 2:

Input:

```
Masukkan jumlah hero : 2
Masukkan detail hero : SUPPORT AERITH 100 15 20
Masukkan detail hero : WARRIOR CLOUD 120 30 DEFAULT
Masukkan perintah : ATTACK CLOUD AERITH
CLOUD berhasil menyerang AERITH
Nyawa AERITH tersisa 50
Masukkan perintah : ATTACK CLOUD AERITH
CLOUD berhasil menyerang AERITH
Nyawa AERITH tersisa 0
Masukkan perintah : HEAL AERITH CLOUD
Karakter tidak ditemukan
Masukkan perintah : EXIT
```

Output:

```
Karakter yang masih hidup:
-----
Nama                | Tipe      | HP
-----
CLOUD                | Warrior   | 120
```

Contoh Interaksi 3:

Input:

```
Masukkan jumlah hero : 4
Masukkan detail hero : SUPPORT AERITH 100 15 DEFAULT
Masukkan detail hero : SUPPORT TIFA 110 10 15
Masukkan detail hero : SUPPORT YUNA 120 10 15
Masukkan detail hero : WARRIOR CLOUD 120 30 5
Masukkan perintah : GABUNGKAN TIFA CLOUD
Gagal menggabungkan karena tipe kedua karakter berbeda
Masukkan perintah : GABUNGKAN AERITH TIFA
AERITH berhasil bergabung dengan TIFA menjadi AERITH_TIFA
Masukkan perintah : ATTACK AERITH_TIFA CLOUD
AERITH_TIFA berhasil menyerang CLOUD
Nyawa CLOUD tersisa 95
Masukkan perintah : GABUNGKAN AERITH_TIFA YUNA
AERITH_TIFA berhasil bergabung dengan YUNA menjadi AERITH_TIFA_YUNA
Masukkan perintah : EXIT
```

Output:

```
Karakter yang masih hidup:
-----
Nama                | Tipe        | HP
-----
CLOUD                | Warrior     | 95
AERITH_TIFA_YUNA     | Support     | 330
```

Contoh Interaksi 4:

Input

```
Masukkan jumlah hero : 6
Masukkan detail hero : WARRIOR CLOUD 70 10 5
Masukkan detail hero : WARRIOR TIFA 65 15 DEFAULT
Masukkan detail hero : SUPPORT AERITH 30 5 50
Masukkan detail hero : SUPPORT YUNA 40 10 DEFAULT
Masukkan detail hero : TANK SNOW 120 10 20
Masukkan detail hero : TANK GLADIO 125 5 DEFAULT
Masukkan perintah : ATTACK CLOUD SNOW
CLOUD berhasil menyerang SNOW
Nyawa SNOW tersisa 120
Masukkan perintah : ATTACK YUNA SNOW
YUNA berhasil menyerang SNOW
Nyawa SNOW tersisa 115
Masukkan perintah : HEAL GLADIO YUNA
GLADIO bukan merupakan Support
Masukkan perintah : HEAL AERITH GLADIO
AERITH berhasil meng-heal GLADIO
Nyawa GLADIO menjadi 175
Masukkan perintah : HEAL SNOW TIFA
SNOW bukan merupakan Support
Masukkan perintah : HEAL AERITH AERITH
AERITH tidak dapat meng-heal dirinya sendiri
Masukkan perintah : HEAL AERITH CLOUD
AERITH berhasil meng-heal CLOUD
Nyawa CLOUD menjadi 120
Masukkan perintah : ATTACK TIFA AERITH
TIFA berhasil menyerang AERITH
Nyawa AERITH tersisa 0
Masukkan perintah : ATTACK LIGHTNING SNOW
Masukkan perintah : EXIT
```

Output:

Karakter yang masih hidup:

Nama	Tipe	HP
CLOUD	Warrior	120
TIFA	Warrior	65
YUNA	Support	40
SNOW	Tank	115
GLADIO	Tank	175

Contoh Interaksi 5;

Input:

```
Masukkan jumlah hero : 6
Masukkan detail hero : WARRIOR TIFA 70 10 5
Masukkan detail hero : WARRIOR CLOUD 65 15 DEFAULT
Masukkan detail hero : SUPPORT AERITH 30 5 50
Masukkan detail hero : SUPPORT YUNA 40 10 DEFAULT
Masukkan detail hero : TANK SNOW 120 10 20
Masukkan detail hero : TANK GLADIO 125 5 DEFAULT
Masukkan perintah : GABUNGAN TIFA AERITH
Gagal menggabungkan karena tipe kedua karakter berbeda
Masukkan perintah : GABUNGAN AERITH TIFA
Gagal menggabungkan karena tipe kedua karakter berbeda
Masukkan perintah : GABUNGAN AERITH YUNA
AERITH berhasil bergabung dengan YUNA menjadi AERITH_YUNA
Masukkan perintah : ATTACK CLOUD AERITH_YUNA
CLOUD berhasil menyerang AERITH_YUNA
Nyawa AERITH_YUNA tersisa 35
Masukkan perintah : HEAL AERITH_YUNA SNOW
AERITH_YUNA berhasil meng-heal SNOW
Nyawa SNOW menjadi 190
Masukkan perintah : GABUNGAN TIFA CLOUD
TIFA berhasil bergabung dengan CLOUD menjadi TIFA_CLOUD
Masukkan perintah : ATTACK TIFA_CLOUD AERITH_YUNA
TIFA_CLOUD berhasil menyerang AERITH_YUNA
Nyawa AERITH_YUNA tersisa 0
Masukkan perintah : EXIT
```

Output:

Karakter yang masih hidup:

Nama	Tipe	HP
SNOW	Tank	190
GLADIO	Tank	125
TIFA_CLOUD	Warrior	135

Task: bonus (Poin: 0)

Soal bonus ini tidak wajib untuk dikerjakan, namun tentu saja akan mendapat pengalaman berharga mengenai konsep multiple inheritance yang ada di python.

Untuk mengerjakan soal Bonus ini, terdapat beberapa hal yang perlu dikerjakan, yaitu:

1. Buatlah sebuah class baru bernama **Boss** yang mewarisi ketiga child class sebelumnya (Support, Warrior, dan Tank).
2. Class **Boss** ini memiliki ketiga atribut khusus dan semua spesifikasi *query* yang dimiliki ketiga child class sebelumnya (ATTACK, HEAL, dan GABUNGGAN).
3. Tentunya fungsi **main()** akan sedikit mengalami perubahan, sesuaikan kembali fungsi main agar program tetap berjalan sebagaimana mestinya.
4. Untuk menambahkan tipe hero **Boss** ini gunakan format sebagai berikut,

**BOSS <NAMA> <HP> <ATTACK> <HEAL_AMOUNT> <EXTRA_ATTACK>
<SHIELD>**

Keterangan:

- Untuk atribut nama, hp, dan attack sama seperti hero lainnya
- Untuk atribut heal_amount, extra_attack, dan shield dapat bernilai **DEFAULT** (bisa saja hanya sebagian yang **DEFAULT**). Apabila nilainya berupa string **DEFAULT** maka atribut tersebut akan di-set sebesar **25**.

HINT: Override fungsi `__init__()`, `__str__()`, dan `__add__()` pada class Boss

Contoh Interaksi

Input:

```
Masukkan jumlah hero : 3
Masukkan detail hero : BOSS DINO 200 40 15 DEFAULT 30
Masukkan detail hero : WARRIOR ZILONG 100 20 DEFAULT
Masukkan detail hero : BOSS LORD 250 35 DEFAULT 10 DEFAULT
Masukkan perintah : ATTACK ZILONG LORD
Masukkan perintah : HEAL DINO ZILONG
Masukkan perintah : GABUNGGAN DINO LORD
Masukkan perintah : ATTACK DINO_LORD ZILONG
Masukkan perintah : EXIT
```

Output:

```
Karakter yang masih hidup:
-----
Nama          | Tipe      | HP
-----
ZILONG        | Warrior   | 5
DINO_LORD     | Boss      | 435
```

Deliverable

Buatlah file .zip dengan format nama **[KodeAsdos]_[Nama]_[NPM]_[Kelas]_lab9.zip** yang berisi file python dengan format nama **[Nama]_[Kelas]_fantasi_terakhir.py**

Contoh:

LouisAkbar_A_fantasi_terakhir.py

DNS_LouisAkbar_1234567890_lab9.zip