

Laporan Tugas Besar
IF1210 Dasar Pemrograman
Semester II 2023/2024

Kelompok K06-H:

Rafi Ihsan Alfathin	16523076
Raynee Soe Putra	16523086
Muhammad Fadhil Rafah	16523266
Muhammad Aqmar Fayyaz Zakaria	19623086
Sabilul Huda	19623226



SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2024

Pernyataan Kelompok

“Saya menyatakan bahwa saya mengerjakan tugas besar ini dengan sejujur-jujurnya, tanpa menggunakan cara yang tidak dibenarkan. Apabila di kemudian hari diketahui saya mengerjakan tugas besar ini dengan cara yang tidak jujur, saya bersedia mendapatkan konsekuensinya, yaitu mendapatkan nilai E pada mata kuliah IF1210 Dasar Pemrograman Semester 2 2023/2024.”

Yang mengeluarkan pernyataan,

Rafi Ihsan Alfathin	16523076
Raynee Soe Putra	16523086
Muhammad Fadhil Rafah	16523266
Muhammad Aqmar Fayyaz Zakaria	19623086
Sabilul Huda	19623226

Daftar Isi

Pernyataan Kelompok.....	2
Daftar Isi.....	3
Daftar Tabel.....	4
Daftar Gambar.....	5
I. Deskripsi Persoalan.....	6
II. Deskripsi Pembagian Kerja Anggota Kelompok.....	11
III. Checklist Hasil Pengerjaan Tugas Besar.....	13
IV. Desain Perintah (Command)	15
V. Desain Kamus Data	18
VI. Desain Dekomposisi Algoritmik dan Fungsional Program.....	19

Daftar Tabel

Tabel 2.1 Pembagian Kerja Anggota Kelompok Berdasarkan Fitur Program.....	10
Tabel 2.2 Daftar Pembagian Pembuatan Laporan.....	11
Tabel 2.3 Daftar <i>Checklist</i> Hasil Rancangan, Implementasi, dan Testing.....	12

Daftar Gambar

Gambar 1 Agent P.....	6
Gambar 2 Dr. Asep Spakbor.....	6
Gambar 7.1 Tampilan Awal.....	67
Gambar 7.2 Saat pilih login.....	67
Gambar 7.3 Saat pilih login lalu berhasil login.....	67
Gambar 7.4 saat pilih register pada lalu berhasil register.....	68
Gambar 7.5 Saat sudah ada user lalu masukkan command help.....	68
Gambar 7.6 Saat save folder.....	69
Gambar 7.7 Tampilan dasar menu help sebelum register/login.....	69
Gambar 7.8 Tampilan dasar menu help saat masuk dari main menu.....	69
Gambar 7.9 Tampilan awal login.....	70
Gambar 7.10 Saat logout berhasil.....	70
Gambar 7.11 saat pilih inventory pada help.....	70
Gambar 7.12 Monster inventory.....	71
Gambar 7.13 detail monster.....	71
Gambar 7.14 Saat back kembali ke menu awal inventory.....	71
Gambar 7.15 Inventory item.....	71
Gambar 7.16 Detail potion.....	71
Gambar 7.17 Keluar dari fungsi inventory.....	72
Gambar 7.18 Tampilan laboratory.....	72
Gambar 7.19 Tampilan battle dengan awal monstert.....	74
Gambar 7.20 Ketika saat battle fungsi attack.....	75
Gambar 7.21 Ketika menggunakan potion.....	75
Gambar 7.22 Quit saat battle.....	76
Gambar 7.23 Tampilan Arena Gambar.....	76
Gambar 7.24 Tampilan shop Gambar	77
Gambar 7.25 Tampilan pembelian shop gagal	77
Gambar 7.26 Keluar dari shop.....	77
Gambar 8.1 Dokumentasi asistensi 1.....	79

Gambar 8.2 Dokumentasi asistensi 2.....	79
Gambar 8.3 Dokumentasi kerja kelompok.....	80

I. Deskripsi Persoalan



Gambar 1. Agent P

Di sebuah kota bernama Danville, hidup seorang ilmuwan jahat bernama Dr. Asep Spakbor. Ia merupakan ancaman yang besar bagi kota Danville karena telah menciptakan monster-monster jahat yang membuat warga kota Danville ketakutan. Namun, Perry, yang dikenal sebagai Agent P, berhadapan dengan Dr. Asep Spakbor. Karena Dr. Asep Spakbor dengan monster-monsternya begitu kuat dan sulit untuk dilawan, Agent P memutuskan untuk mencari monster-monster sendiri sehingga bisa menyaingi Dr. Asep Spakbor. Kami sebagai mahasiswa TPB STEI 2023 ditugaskan untuk membantu Agent P mencari dan melatih monster-monsternya sehingga dapat mengalahkan Dr. Asep Spakbor demi menjaga kota Danville.



Gambar 2. Dr. Asep Spakbor

Sesuai dengan arahan "Spesifikasi Tugas Besar IF1210", agen harus memenuhi persyaratan fungsional tertentu. Interface atau tampilan sistem dapat dirancang sedemikian rupa sehingga keluaran nya sedikit berbeda dari contoh; memenuhi spesifikasi adalah hal yang paling penting. Evaluasi akan mempertimbangkan daya cipta antarmuka. Perlu disebutkan bahwa penekanannya adalah pada alur program dan verifikasi tindakan yang diambil. Pemrograman dapat dibuat lebih singkat dan mudah dengan membangun program

modular dengan fungsi dan operasi yang berbeda. Oleh karena itu, program ini tidak akan menjadi panjang atau rumit karena banyaknya kemampuan yang harus diterapkan.

Pada cerita di Kota Danville ini, terdapat program berisi fungsional–fungsional yang membangun cerita perlawanan Agent P dengan Dr. Asep Spakbor, yaitu:

- a. **F00 - Random Number Generator (RNG)**, akan digunakan untuk menghasilkan bilangan acak dari sebuah *range* angka. Fitur ini akan digunakan untuk berbagai fitur, terutama pada monster dan *battle*. Implementasi RNG menggunakan algoritma *Linear Congruential Generator* (LCG), tidak/dilarang menggunakan library random python.
- a. **F01 - Register**, saat awal menggunakan sistem, pengguna harus melakukan registrasi akun terlebih dahulu dengan memasukkan nama pengguna (*username*) dan kata sandi (*password*) yang digunakan untuk mengakses sistem. Secara otomatis, pengguna akan mendapatkan peran sebagai “Agent” dan memiliki O.W.C.A Coins dengan saldo sebesar 0.
- b. **F02 - Login**, Agent maupun Admin dapat untuk melakukan login ke dalam sistem. *Login* dilakukan dengan memasukkan nama pengguna (*username*) dan kata sandi (*password*) yang telah didaftar pada saat registrasi akun di *Register*. Terdapat 3 jenis kesalahan yang dapat terjadi ketika login; kesalahan *username*, kesalahan *password*, dan pemanggilan *command* ketika sudah *logged in*.
- c. **F03 - Logout**, prosedur ini dapat digunakan oleh pengguna untuk keluar dari akun yang sedang dipakai. Untuk mengakses akun yang telah di-*logout*, pengguna dapat melakukan *login* kembali menggunakan akun yang berbeda.
- d. **F04 - Menu & Help**, mencakup setiap perintah yang tersedia berdasarkan status login. Selain itu, perintah yang dapat digunakan sebelum login dapat ditampilkan sebelum login..
- e. **F14 - Load**, dapat memuat data yang sesuai dengan struktur data eksternal. Pada saat pengguna memulai program, dengan menamakan *folder* yang berisi *file* penyimpanan, *Load* akan dijalankan sekali
- f. **F15 - Save**, prosedur ini digunakan untuk menyimpan data ke dalam *file* yang sesuai dengan struktur data eksternal. Dalam prosedur ini, program akan meminta nama *folder* yang akan digunakan sebagai tempat penyimpanan *file*. Berikut adalah ketentuannya: Jika nama *folder* tidak ditemukan, program akan membuat *folder* sesuai dengan masukan dan memberikan pesan bahwa ia membuat *folder* baru dan berhasil menyimpan data di *folder* tersebut. Jika

nama *folder* ditemukan namun belum ada *file* pada folder tersebut, maka program akan menaruh *file* baru pada *folder* tersebut. Jika nama *folder* dan *file* sudah ada, program akan mengganti *file* pada *folder* tersebut dengan yang lebih baru dan tidak perlu memberikan pesan tambahan (*overwrite/replace*).)

- g. **F16 - Exit**, berfungsi untuk meninggalkan permainan, dimana sebelum *Exit*, pengguna dapat melakukan penyimpanan file dengan menggunakan *Save*.

Terdapat fungsional-fungsional yang dapat dijalankan oleh Agent:

- i. **F05 - Monster**, adalah monster yang bisa dipanggil atau digunakan oleh Agent dalam *Battle*. Monster dan atribut (Tipe, ATK, DEF, dan HP) dikompilasi ke dalam file CSV yang disimpan dalam sebuah *database* "owca-dex". Informasi level monster disimpan dalam *inventory* Agent.

- j. **F06 - Potion**, item ini disimpan dalam *inventory* Agent dan dapat digunakan pada saat *battle*. Terdapat beberapa jenis *potion*; *Strength Potion*, *Resilience Potion*, dan *Healing Potion*.

- k. **F07 - Inventory**, tersimpan dalam file bentuk csv, *inventory* ini berisi *monster* dan *potion* beserta statistik masing-masing (Monster: Name, ATK Power, DEF Power, HP, Level. Potion: Type, Quantity).

- l. **F08 - Battle**, Agent dapat memilih untuk bertarung melawan monster secara acak. Prosedur ini bersifat *turn-based IVI game*. Setiap pertempuran akan terdiri dari serangkaian putaran dimana Agent dan monster bergantian melakukan tindakan. Berikut adalah alur pertarungan.

1. Muncul monster (musuh) secara acak (RNG) *database* Monster; Agent memilih monster (Agent) yang ingin dipertarungkan
2. Setiap putaran Agent memiliki pilihan untuk "Attack", "Use Potion", atau "Quit"; Monster (musuh) hanya bisa "Attack"
3. Kondisi kemenangan adalah saat nyawa monster (musuh) habis; Agent mendapatkan OC (OC yang diterima acak (RNG), misal 5-30)
4. Kondisi kekalahan adalah saat nyawa monster (Agent) habis; Agent keluar dari pertempuran, nyawa monster yang dipertarungkan kembali penuh (tidak hilang dari *inventory*)

- m. **F09 - Arena**, merupakan tempat untuk melatih monster untuk membantu Agent lebih siap melawan Dr. Asep Spakbor. Arena ini memiliki beberapa detail;

1. Sesi latihan terdapat dalam 5 *stage* dengan *stage* merepresentasikan level monster tertentu. Monster yang muncul pada tiap *stage* terpilih secara acak dari monster-monster yang terdapat pada *database*.
2. Mekanisme pertarungan pada tiap stage sama seperti *battle* biasa.

3. Apabila agent kalah pada suatu *stage* (monster miliknya mati), maka sesi latihan berakhir dan *game over*. Agent dapat keluar dari pertarungan dan akan mendapatkan status *game over* juga.
4. Setelah sesi latihan berakhir yaitu Agent berhasil memenangkan kelima *stage* atau terjadi *game over*, hasil *stats* sesi latihan akan ditampilkan pada *scoreboard* yang setidaknya terdiri atas informasi total hadiah yang diterima, berhasil menang hingga *stage* berapa, total *damage* yang diberikan, dan total *damage* yang diterima. Informasi lain dapat ditambahkan (opsional). Jumlah hadiah pada setiap *stage* dibebaskan (hardcode pada program).

k. **F10 - Shop & Currency**, sebuah fungsional yang dimana Agent mendapatkan akses untuk melihat Shop dan membeli *monster* dan *potion* dengan O.W.C.A Coins yang dimiliki setelah memenangkan *battle*, *arena*, ataupun *mini games*.

l. **F11 - Laboratory**, pada *command* ini, Agent dapat *upgrade* monster yang dimiliki di *inventory*, yaitu menguatkan kemampuan monster dalam *battle* dengan menaikkan level monster. *Upgrade* dapat dilakukan apabila O.W.C.A Coins yang dimiliki oleh Agent cukup. Semakin tinggi level monster, semakin mahal biaya *upgrade*.

a. Terdapat beberapa fungsional yang dapat dieksekusi oleh Admin:

i. **F12 - Shop Management**, *item* atau monster dan *potion* yang ada di shop (pada *shop & currency*) dapat diatur oleh Admin yang memperbolehkan untuk melakukan perubahan, penambahan, dan penghapusan *item* pada shop. *Item* yang dijual di *shop* memiliki database sendiri. Monster dan *potion* yang ditampilkan pada *shop* berasal dari *database*.

ii. **F13 - Monster Management**, fungsional ini memperbolehkan Admin untuk mengatur monster yang ada dalam *database* yaitu dengan melakukan implementasi fitur *monster management*. Validasi input diwajibkan, sesuai yang ada di dalam *database*.

II. Deskripsi Pembagian Kerja Anggota Kelompok

Tabel 2.1. Pembagian Kerja Anggota Kelompok Berdasarkan Fitur Program

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F00 – Random Number Generator	Fungsi random_uniform_sample	19623226	16523086, 19623226	19623226
F01 – Register	Fungsi register	19623226	19623226	19623226 16523086
F02 – Login	Fungsi login	16523076	16523076	19623086 16523086
F03 – Logout	Fungsi logout	16523076	16523076	19623086 16523086
F04 – Menu & Help	Fungsi menu_help	16523076, 19623086	16523076, 19623086	16523076, 19623086
F05 – Monster		19623226	19623226	19623226
F06 – Potion		16523086	19623226	19623226
F07 – Inventory	Fungsi inventory	19623086	16523086, 16523076	16523076
F08 – Battle	Fungsi battle	16523086	19623226	19623226, 19623086
F09 – Arena	Fungsi arena	19623226	19623226	19623226, 19623086
F10 – Shop & Currency	Fungsi toInteger Fungsi is_digit Fungsi isIn Fungsi itsIndex Fungsi shop_currency	19623226	16523266, 19623226	16523076, 19623086

F11 – Laboratory	Fungsi monster_upgrade Fungsi laboratory	16523076	16523266, 16523076	16523076, 19623086
F12 – Shop Management	Fungsi shop_management	16523266	16523266	16523076, 19623086
F13 – Monster Management	Fungsi strip Fungsi split Fungsi is_integer Fungsi id_baru Fungsi cek_sama Fungsi monster_management	16523266	16523266	19623226
F14 – Load	Fungsi load	19623226	19623226	19623226
F15 – Save	Fungsi save	19623226	19623226	19623226
F16 – Exit	Fungsi exit	19623226	19623226	19623226
Tipe Bentuk dan Penggabungan <i>Code</i>	Fungsi main	19623086	19623086, 16523076	19623086, 16523076

Tabel 2.2. Daftar Pembagian Pembuatan Laporan

No.	Bagian Laporan	NIM
1.	Halaman <i>Cover</i>	16523266
2.	Pernyataan Kelompok	16523266
3.	Daftar Isi	16523086
4.	Daftar Tabel	16523266
5.	Daftar Gambar	16523266
6.	Deskripsi Persoalan	16523266

7.	Deskripsi Pembagian Kerja Anggota Kelompok	16523266
8.	<i>Checklist</i> Hasil Pengerjaan Tugas Besar	16523076
9.	Desain Perintah (<i>Command</i>)	19623086
10.	Desain Kamus Data	-
11.	Desain Dekomposisi Algoritmik dan Fungsional Program	16523086
12.	Spesifikasi Program	16523076
13.	Hasil Pengujian Program	16523086
14.	Catatan dan Lampiran	16523076

III. *Checklist* Hasil Pengerjaan Tugas Besar

Tabel 3.1. Daftar *Checklist* Hasil Rancangan, Implementasi, dan Testing Setiap Primitif.

Fitur	Desain	Implementasi	Testing
F00 – Random Number Generator	v	v	v
F01 – Register	v	v	v
F02 – Login	v	v	v
F03 – Logout	v	v	v
F04 – Menu & Help	v	v	v
F05 – Monster	v	v	v
F06 – Potion	v	v	v
F07 – Inventory	v	v	v
F08 – Battle	v	v	v
F09 – Arena	v	v	v
F10 – Shop & Currency	v	v	v
F11 – Laboratory	v	v	v
F12 – Shop Management	v	x	x
F13 – Monster Management	v	x	x
F14 – Load	v	v	v
F15 – Save	v	v	v
F16 – Exit	v	v	v

Tipe Bentuk dan Penggabungan <i>Code</i>	v	v	v
--	---	---	---

****Keterangan:**

- v : Sudah selesai dikerjakan;
- x : Dikerjakan tapi belum selesai;
- : Tidak dikerjakan sama sekali.

IV. Desain Perintah (*Command*)

Berikut adalah beberapa desain perintah (*command*) utama dari program,

a. F01 - Register

```
function register(user: list, monster: list, monster_inv: list,
sudah_login: bool) -> tuple[list, list, bool]
    { I.S : Pengguna belum memiliki akun untuk melakukan login }
    { F.S : Data akun pengguna akan tersimpan di database yang dapat
digunakan oleh pengguna untuk login }
```

```
# Proses register berhasil
>>> REGISTER

Masukkan username: ininama
Masukkan password: inipassword

► Silahkan pilih salah satu monster sebagai monster awalmu! ◀

1. Charizard
2. Bulbasaur
3. Aspal

Monster Pilihanmu: 1
Berhasil melakukan register!

Selamat datang Agent ininama. Mari kita mengalahkan Dr. Asep Spakbor
dengan Charizard!
```

```
# Username tidak sesuai kriteria untuk register
>>> REGISTER
=====

Masukkan username: h@lloworld
Masukkan password: inipassword

Username hanya boleh berisi alfabet, angka, underscore, dan strip!
```

```
# Username sudah terpakai (tidak bisa register)
>>> REGISTER

Masukkan username: AgentPurry
Masukkan password: inipassword

Username AgentPurry sudah terpakai, silahkan gunakan username lain!
```


b. F02 - Login

```
function login(sudah_login: bool, user: list) -> tuple[bool, int, str, str, int]
    { I.S : Pengguna belum login ke dalam program }
    { F.S : Pengguna sudah login ke dalam program }
```

```
# Login berhasil sebagai agent
```

```
>>> Login
```

```
=====
```

```
Masukkan username: ininama
```

```
Masukkan password: inipassword
```

```
Selamat datang, Agent ininama!
```

```
Masukkan command "Help" untuk daftar command yang dapat kamu panggil.
```

```
Masukkan command "Exit" untuk keluar dari program.
```

```
# Login berhasil sebagai admin
```

```
>>> Login
```

```
=====
```

```
Masukkan username: admin1
```

```
Masukkan password: admin1
```

```
Selamat datang, admin1!
```

```
Masukkan command "Help" untuk daftar command yang dapat kamu panggil.
```

```
Masukkan command "Exit" untuk keluar dari program.
```

```
# Login tidak berhasil karena username tidak terdaftar
```

```
>>> Login
```

```
=====
```

```
Masukkan username: inisaya
```

```
Masukkan password: inipassword
```

```
Username tidak terdaftar!
```

```
Loading...
```

```
Masukkan command "Help" untuk daftar command yang dapat kamu panggil.
```

Masukkan command "Exit" untuk keluar dari program.

>>> Masukkan command :

Login tidak berhasil karena password salah

>>> **Login**

=====

Masukkan username: **ininama**

Masukkan password: **inisaya**

Password salah!

1. Login
2. Register
3. Help

>>> Pilih command yang akan kamu pilih :

c. F03 - Logout

```
function logout(statusLogin: bool) -> bool
{ I.S : Pengguna sudah login ke dalam program }
{ F.S : Pengguna keluar dari program (logout) }
```

Logout berhasil karena User sudahlogin

>>> **Logout**

=====

Logout berhasil!

Silahkan login lagi jika ingin mengakses kembali program

d. F04 - Menu & Help

```
procedure help_sebelumlogin()
{ I.S : Pengguna berada di halaman utama sebelum login }
{ F.S : Terdapat menu yang dapat dipilih oleh pengguna sebelum
login untuk membantu pengguna }
```

Menu help sebelum login sebagai admin maupun agent

>>> **HELP**

=====

===== HELP =====

Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu.

1. Login: Masuk ke dalam akun yang sudah terdaftar

2. Register: Membuat akun baru

>>> Pilih fungsi :

procedure help_setelahlogin_agent(uname: str)
 { I.S : Pengguna berada di halaman utama setelah login sebagai agent }
 { F.S : Terdapat menu yang dapat dipilih oleh pengguna setelah login untuk membantu pengguna }

Menu help setelah login sebagai agent

>>> **HELP**

=====

===== HELP =====

Halo Agent a. Kamu memanggil command HELP. Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian. Berikut adalah hal-hal yang dapat kamu lakukan sekarang:

1. Logout: Keluar dari akun yang sedang digunakan
2. Inventory: Lihat OWCA-dex dan peralatanmu, Agent a
3. Laboratory: Upgrade monster yang kamu miliki, Agent a!
4. Battle : Ayo bertarung di Arena, Agent a dan menangkan hadiahnya!
5. Arena : Ayo masuk ke Arena, Agent a dan latih monstermu!
6. Shop and Currency : Ayo belanjakan OWCA-mu dengan potion dan monster di sini, Agent a!

Footnote:

1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
2. Jangan lupa untuk memasukkan input yang valid

>>> Pilih fungsi :

procedure help_setelahlogin_admin()
 { I.S : Pengguna berada di halaman utama setelah login sebagai admin}
 { F.S : Terdapat menu yang dapat dipilih oleh admin setelah login untuk membantu admin }

>>> **HELP**

=====

===== HELP =====

Selamat datang, Admin. Berikut adalah hal-hal yang dapat kamu lakukan:

1. Logout: Keluar dari akun yang sedang digunakan
2. Shop Management: Lakukan manajemen pada SHOP sebagai tempat jual beli peralatan Agent
3. Monster Management: Lakukan manajemen pada SHOP untuk menambah monster

Footnote:

```
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang
terdaftar
2. Jangan lupa untuk memasukkan input yang valid

>>> Pilih fungsi :
```

e. F07 - Inventory

```
function inventory(userinventory : list, yourmonsterdata: list,
monsterdata: list, coin: int, user_idx : int)
{ I.S : Pengguna berada di halaman utama setelah login sebagai
agent }
{ F.S : Pengguna berada dalam menu inventory yang dapat digunakan
pengguna untuk melihat potion dan monster yang dimilikinya }
```

```
>>> INVENTORY
```

```
=====
jumlah OWCA coin anda saat ini : 1000
Silakan pilih jenis item yang ingin diketahui Anda
(Monster/Item/Back): Monster
1. Sanyaz
2. Charizard
3. Onix
Masukkan nomor monster untuk menampilkan detail monster
(1/2/dst/Back): 1
Nama      : Sanyaz
ATK Power : 1098.9
DEF Power : 1098.9
HP        : 1098.9
Level     : 3
Masukkan nomor monster untuk menampilkan detail monster
(1/2/dst/Back): 2
Nama      : Charizard
ATK Power : 29.4
DEF Power : 15.75
HP        : 115.5
Level     : 2
Masukkan nomor monster untuk menampilkan detail monster
(1/2/dst/Back): 3
Nama      : Onix
ATK Power : 19.0
DEF Power : 14.0
HP        : 110.0
Level     : 1
Masukkan nomor monster untuk menampilkan detail monster
```

```
(1/2/dst/Back): back
Silakan pilih jenis item yang ingin diketahui Anda
(Monster/Item/Back): item
1. Type: ATK potion
2. Type: HEAL potion
3. Type: DEF potion
Masukkan nomor potion untuk menampilkan detail item (1/2/3/Back): 1
Type: ATK potion
Quantity: 10
Masukkan nomor potion untuk menampilkan detail item (1/2/3/Back): 2
Type: HEAL potion
Quantity: 15
Masukkan nomor potion untuk menampilkan detail item (1/2/3/Back): 3
Type: DEF potion
Quantity: 20
Masukkan nomor potion untuk menampilkan detail item (1/2/3/Back):
back
Silakan pilih jenis item yang ingin diketahui Anda
(Monster/Item/Back): back
```

f. F08 - Battle

```
function battle(data : list, user_id : int)
{ I.S : Pengguna sudah login sebagai Agent }
{ F.S : OC pengguna bertambah ketika sudah menyelesaikan battle }
```

```
>>> BATTLE
```

```
=====
AYOO LAWAN MONSTER Pikachu !!!
```

```
Name      : Pikachu
ATK Power : 20.0
DEF Power : 20.0
HP        : 110.0
Level     : 1
```

```
=====
SEMANGAT MONSTER Sanyaz KAMU PASTI BISA !!!
```

```
Name      : Sanyaz
ATK Power : 1198.8
DEF Power : 1198.8
```

HP : 1198.8

Level : 3

===== TURN 1 (Sanyaz) =====

1. Attack
2. Use Potion
3. Quit

Pilih perintah: **2**

===== POTION LIST =====

1. Strength Potion (Qty : 10) - Increases ATK Power
2. Healing Potion (Qty : 15) - Restores Health
3. Resilience Potion (Qty : 20) - Increases DEF Power
4. Cancel

Potion yang diinginkan: **1**

Setelah meminum ramuan ini, aura kekuatan terlihat mengelilingi Sanyaz dan gerakannya menjadi lebih cepat dan mematikan.

===== TURN 1 (Pikachu) =====

Damage berkurang -1098.8% karena DEF Power: -219.6

Sisa HP Sanyaz adalah 1418.0

Klik apapun untuk lanjut: **w**

=====

AYOO LAWAN MONSTER Pikachu !!!

Name : Pikachu

ATK Power : 20.0

DEF Power : 20.0

HP : 110.0

Level : 1

=====

SEMANGAT MONSTER Sanyaz KAMU PASTI BISA !!!

Name : Sanyaz

ATK Power : 1198.8

DEF Power : 1198.8

HP : 1418.0

Level : 3

===== TURN 2 (Sanyaz) =====

1. Attack
2. Use Potion
3. Quit

Pilih perintah: **2**

===== POTION LIST =====

1. Strength Potion (Qty : 9) - Increases ATK Power
2. Healing Potion (Qty : 15) - Restores Health
3. Resilience Potion (Qty : 20) - Increases DEF Power
4. Cancel

Potion yang diinginkan: **4**

===== TURN 2 (Sanyaz) =====

1. Attack
2. Use Potion
3. Quit

Pilih perintah: **1**

Attack berkurang 66.4% karena DEF Power: 796.0032000000001
Sisa HP Pikachu adalah -665.0

=====

Selamat, Anda berhasil mengalahkan monster Pikachu!!!

Total OC yang diperoleh: 30

=====

g. F09 - Arena

```
function arena(data: list, user_id : int)
{ I.S : Pengguna sudah login sebagai Agent }
{ F.S : OC pengguna bertambah ketika sudah menyelesaikan arena }
```

>>> Arena

=====

Selamat datang di Arena!!

===== MONSTER LIST =====

1. Sanyaz
2. Charizard

```

3. Onix
Pilih monster untuk bertarung: 1

RAWRRR, a mengeluarkan monster Sanyaz !!!

Name      : Sanyaz
ATK Power : 1198.8
DEF Power : 1198.8
HP        : 1198.8
Level     : 3

Klik apapun untuk lanjut: w
===== STAGE 1 =====

RAWRRR, Monster Jigglypuff telah muncul !!!

Name      : Jigglypuff
ATK Power : 29.0
DEF Power : 29.0
HP        : 110.0
Level     : 1

===== TURN 1 (Sanyaz) =====
1. Attack
2. Use Potion
3. Quit

Pilih perintah: 1

Attack berkurang 71.0% karena DEF Power: 851.148
Sisa HP Jigglypuff adalah 0

=====
Selamat, Anda berhasil mengalahkan monster Jigglypuff !!!
STAGED CLEARED !!! Total OC yang diperoleh pada sesi ini: 30
Memulai stage berikutnya ...

===== STATS =====
Total hadiah      : 30 OC
Jumlah stage      : 1
Damage diberikan  : 29
Damage diterima   : 1198.

```

h. F10 - Shop & Currency

```

function shop(users: dict, user_now: str, role_now: str) ->
tuple[str, str, bool]
{ I.S : Pengguna sudah login sebagai Agent }
{ F.S : Data pengguna pada inventory akan berubah setelah
dilakukannya aksi di shop }

```

```

>>> Shop & Currency
=====
Trasshaimase! Selamat datang di SHOP!!
Mau apa (lihat/beli/keluar) :lihat
Lihat apa (monster/item): item
ID | Type                  | Stok  | Harga
1  | Strength Potion         | 2     | 30

```



```

2 | Healing Potion      | 5      | 10
3 | Resilience Potion  | 3      | 20
Mau apa (lihat/beli/keluar) :keluar
Mr. Yanto bilang makasih, belanja lagi ya nanti :)

>>> Shop & Currency
=====
Trasshaimase! Selamat datang di SHOP!!
Mau apa (lihat/beli/keluar) :lihat
Lihat apa (monster/item): monster
ID | Type           | ATK Power | DEF Power | HP   | Stok | Harga
1  | Sanyaz         | 999       | 999       | 999  | 10   | 500
2  | Pikachu        | 20        | 20        | 110  | 5    | 700
3  | Charizard      | 28        | 15        | 110  | 3    | 1000
Mau apa (lihat/beli/keluar) :beli
Jumlah OWCA coinmu sekarang adalah: 1000
Mau beli apa? (monster/item): monster
Masukkan id monster: 1
Monster Sanyaz sudah ada dalam inventory-mu! Pembelian dibatalkan.
Mau apa (lihat/beli/keluar) :beli
Jumlah OWCA coinmu sekarang adalah: 1000
Mau beli apa? (monster/item): item
Masukkan id potion: 1
Masukkan jumlah: 1
Berhasil membeli item: 1 Strength Potion. Item sudah masuk ke
inventory-mu!
Mau apa (lihat/beli/keluar) :lihat
Lihat apa (monster/item): item
ID | Type           | Stok | Harga
1  | Strength Potion | 1    | 30
2  | Strength Potion | 2    | 30
3  | Healing Potion  | 5    | 10
Mau apa (lihat/beli/keluar) :beli
Jumlah OWCA coinmu sekarang adalah: 970
Mau beli apa? (monster/item): 2
Inputan salah!
Mau beli apa? (monster/item): item
Masukkan id potion: 2
Masukkan jumlah: 3
Berhasil membeli item: 3 Strength Potion. Item sudah masuk ke
inventory-mu!
Mau apa (lihat/beli/keluar) :beli
Jumlah OWCA coinmu sekarang adalah: 940
Mau beli apa? (monster/item): monster
Masukkan id monster: 3
OC-mu tidak cukup.
Mau apa (lihat/beli/keluar) :keluar
Mr. Yanto bilang makasih, belanja lagi ya nanti :)

```

i. F11 - Laboratory

```

function laboratory(monster, monster_inv, oc, id) -> tuple[oc: int,
monster_inv: list]

{ I.S : Pengguna sudah login sebagai Agent }
{ F.S : Data pengguna pada inventory akan berubah setelah
dilakukannya aksi di laboratory}

>>> laboratory
=====

```

Selamat datang di Lab Dokter Asep !!!

===== MONSTER LIST =====

1. Chaca (Level: 1)
2. Pikachow (Level: 2)
3. Zeze (Level: 5)

===== UPGRADE PRICE =====

1. Level 1 -> Level 2: 300 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1000 OC

>>> Pilih monster (atau keluar): 4

Pilihan tidak valid, silakan pilih nomor monster yang valid.

Selamat datang di Lab Dokter Asep !!!

===== MONSTER LIST =====

1. Chaca (Level: 1)
2. Pikachow (Level: 2)
3. Zeze (Level: 5)

===== UPGRADE PRICE =====

1. Level 1 -> Level 2: 300 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1000 OC

>>> Pilih monster (atau keluar): 2

Pikachow akan di-upgrade ke level 3.

Harga untuk melakukan upgrade Pikachow adalah 500 OC.

>>> Lanjutkan upgrade (Y/N): Y

Selamat, Pikachow berhasil di-upgrade ke level 3!

{berhasil mengupgrade monster}

Selamat datang di Lab Dokter Asep !!!

===== MONSTER LIST =====

1. Chaca (Level: 1)
2. Pikachow (Level: 3)
3. Zeze (Level: 5)

===== UPGRADE PRICE =====

1. Level 1 -> Level 2: 300 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1000 OC

{monster yang telah di-upgrade telah naik level. Penampilan List berbeda}

>>> Pilih monster (atau keluar): 3

Maaf, monster yang Anda pilih sudah memiliki level maksimum.

Selamat datang di Lab Dokter Asep !!!

```

===== MONSTER LIST =====
1. Chaca (Level: 1)
2. Pikachow (Level: 3)
3. Zeze (Level: 5)

===== UPGRADE PRICE =====
1. Level 1 -> Level 2: 300 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1000 OC
>>> Pilih monster (atau keluar): 2

Pikachow akan di-upgrade ke level 4.
Harga untuk melakukan upgrade Pikachow adalah 800 OC.
Maaf, OC Anda tidak mencukupi untuk melakukan upgrade.

{upgrade gagal sebab OC tidak cukup}

Selamat datang di Lab Dokter Asep !!!

===== MONSTER LIST =====
1. Chaca (Level: 1)
2. Pikachow (Level: 3)
3. Zeze (Level: 5)

===== UPGRADE PRICE =====
1. Level 1 -> Level 2: 300 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1000 OC
>>> Pilih monster (atau keluar): keluar

Terima kasih sudah datang ke Lab Dokter Asep.

```

j. F12 - Shop Management

```

function laboratory(monster, monster_inv, oc, id) -> tuple[oc: int,
monster_inv: list]

```

k. F13 - Monster Management

```

function tambahData(data: list[list[str]], delimiter: str = ";") ->
str
function showMonsterData(data: list[list[str]]) -> list[str]

{ I.S : Pengguna sudah login sebagai Admin }
{ F.S : Data monster akan berubah setelah dilakukannya aksi di
monster management }

>>> monster_management

SELAMAT DATANG DI DATABASE PARA MONSTER !!!

```

1. Tampilkan semua monster
2. Tambah monster baru
3. Keluar

Pilih Aksi : 1

ID	TYPE	ATK	DEF	HP
1	Snorlax	60	66	43
2	Pikachu	52	53	40
3	Goku	54	63	45
4	Bejita	46	47	40
5	Naruto	62	50	29

1. Tampilkan semua monster
2. Tambah monster baru
3. Keluar

Pilih Aksi : 2

Memulai pembuatan monster baru...

Masukkan Type>Nama : Sasuke

Masukkan ATK Power : 50

Masukkan DEF Power: 40

Masukkan HP: 55

Name/Type	ATK Power	DEF Power	HP
Sasuke	50	40	55

Sasuke | 50 | 40 | 55

Tambahkan Monster ke Database (Y/N) : y

Monster baru telah ditambahkan!

1. Tampilkan semua monster
2. Tambah monster baru
3. Keluar

Pilih Aksi : 1

ID	TYPE	ATK	DEF	HP
1	Snorlax	60	66	43
2	Pikachu	52	53	40
3	Goku	54	63	45
4	Bejita	46	47	40
5	Naruto	62	50	29
6	Sasuke	50	40	55

1. Tampilkan semua monster
2. Tambah monster baru
3. Keluar

Pilih Aksi : 2

Memulai pembuatan monster baru...

Masukkan Type>Nama : Mbappe

Masukkan ATK Power : 90

Masukkan DEF Power: 51

DEF Power harus bernilai 0-50, coba lagi!

{gagal. input DEF diatas 50}

Masukkan DEF Power: 30

Masukkan HP: 35

Name/Type	ATK Power	DEF Power	HP
Mbappe	90	30	35

Mbappe | 90 | 30 | 35

Tambahkan Monster ke Database (Y/N) : y

Monster baru telah ditambahkan!

1. Tampilkan semua monster

```
2. Tambah monster baru
3. Keluar
```

```
Pilih Aksi : 1
```

ID	TYPE	ATK	DEF	HP
1	Snorlax	60	66	43
2	Pikachu	52	53	40
3	Goku	54	63	45
4	Bejita	46	47	40
5	Naruto	62	50	29
6	Sasuke	50	40	55
7	Mbappe	90	30	35

```
1. Tampilkan semua monster
2. Tambah monster baru
3. Keluar
```

```
Pilih Aksi : 3
```

```
Selamat tinggal!
```

I. F14 - Load

```
function load(filename: str) -> tuple
function loading(directory: str) -> str or None
    { I.S : Program belum dijalankan }
    { F.S : Program dijalankan dengan memuat data dari folder yang
    spesifik }
```

```
>>> Load
```

```
=====
python main.py data/07-05-2024
```

```
Loading...
```

```
Selamat datang di program OWCA!
```

```
1. Login
2. Register
3. Help
```

```
>>> Pilih command yang akan kamu pilih :
```

```
# Jika tidak ada nama folder yang diberikan
```

```
>>> Load
```

```
=====
Tidak ada nama folder yang diberikan!
Usage: python main.py <nama_folder>
```

m. F15 - Save

```
procedure save(file_name: str)
  { I.S : Program berjalan }
  { F.S : Perubahan pada program dicatat dan dimasukkan ke dalam
  folder spesifik }
```

```
# Jika ingin save file
```

```
>>> Save
```

```
=====
```

```
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n)
```

```
y
```

```
Masukkan nama folder: ab
```

```
Membuat folder data\ab...
```

```
Berhasil menyimpan data di folder data\ab!
```

```
# Jika tidak ingin save file
```

```
>>> Save
```

```
=====
```

```
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n)
```

```
n
```

n. F16 - Exit

```
function exit()
  { I.S : Program berjalan }
  { F.S : Pengguna keluar dari game dan perubahan pada program
  dicatat dan dimasukkan ke dalam folder spesifik }
```

```
# Jika ingin save file
```

```
>>> Exit
```

```
=====
```

```
Terima kasih!
```

```
Sampai jumpa.
```

```
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n)
```

```
y
```

```
Masukkan nama folder: ab
```

```
Membuat folder data\ab...
```

```
Berhasil menyimpan data di folder data\ab!
```

```
# Jika tidak ingin save file
```

```
>>> Exit
```

```
=====
```

```
Terima kasih!
```

```
Sampai jumpa.
```

```
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n)
```

```
n
```

V. Desain Kamus Data

VI. Desain Dekomposisi Algoritmik dan Fungsional Program

Desain dekomposisi algoritmik dan fungsional program berdasarkan alur pengguna sebagai seorang Agent adalah sebagai berikut.

A. F14 - Load

```
function load(filename: str) -> tuple  
function loading(directory: str) -> str or None  
    { I.S : Program belum dijalankan }  
    { F.S : Program dijalankan dengan memuat data dari folder  
yang spesifik }
```

B. F01 - Register

```
function register(user: list, monster: list, monster_inv: list,  
sudah_login: bool) -> tuple[list, list, bool]  
    { I.S : Pengguna belum memiliki akun untuk melakukan login }  
    { F.S : Data akun pengguna akan tersimpan di database yang  
dapat digunakan oleh pengguna untuk login }
```

C. F02 - Login

```
function login(sudah_login: bool, user: list) -> tuple[bool, int,  
str, str, int]  
    { I.S : Pengguna belum login ke dalam program }  
    { F.S : Pengguna sudah login ke dalam program }
```

D. F04 - Menu & Help

```
procedure help_sebelumlogin()  
{ I.S : Pengguna berada di halaman utama (sebelum login) }  
{ F.S : Menu berisikan command (perintah) yang dapat dipilih oleh  
pengguna sebelum login untuk membantu pengguna }
```


E. F07 - Inventory

```
function inventory(userinventory : list, yourmonsterdata: list,
monsterdata: list, coin: int, user_idx : int)
{ I.S : Pengguna berada di halaman utama setelah login sebagai
agent }
{ F.S : Pengguna berada dalam menu inventory yang dapat
digunakan pengguna untuk melihat potion dan monster yang
dimilikinya }
```

F. F08 - Battle

```
function battle(data : list, user_id : int)
{ I.S : Pengguna sudah login sebagai Agent }
{ F.S : OC pengguna bertambah ketika sudah menyelesaikan battle
}
```

G. F09 - Arena

```
function arena(data: list, user_id : int)
{ I.S : Pengguna sudah login sebagai Agent }
{ F.S : OC pengguna bertambah ketika sudah menyelesaikan arena
}
```

H. F10 - Shop & Currency

```
function shop(users: dict, user_now: str, role_now: str) ->
tuple[str, str, bool]
{ I.S : Pengguna sudah login sebagai Agent }
{ F.S : Data pengguna pada inventory akan berubah setelah
```

```
dilakukannya aksi di shop }
```

I. F15 - Save

```
procedure save(file_name: str)
  { I.S : Program berjalan }
  { F.S : Perubahan pada program dicatat dan dimasukkan ke dalam
  folder spesifik }
```

J. F03 - Logout

```
function logout(statusLogin: bool) -> bool
  { I.S : Pengguna sudah login ke dalam program }
  { F.S : Pengguna keluar dari program (logout) }
```

K. F16 - Exit

```
function exit()
  { I.S : Program berjalan }
  { F.S : Pengguna keluar dari game dan perubahan pada program
  dicatat dan dimasukkan ke dalam folder spesifik }
```

Desain dekomposisi aloritmik dan fungsional program sebagai seorang Admin adalah sebagai berikut.

L. F12 - Shop Management

```
function shop_management ()
  { I.S : Pengguna sudah login sebagai Admin }
  { F.S : Data shop akan berubah setelah dilakukannya aksi di
  shop management }
```

M. F12 - Monster Management

```
function tambahData(data: list[list[str]], delimiter: str = ";")  
-> str  
function showMonsterData(data: list[list[str]]) -> list[str]  
  
{ I.S : Pengguna sudah login sebagai Admin }  
{ F.S : Data monster akan berubah setelah dilakukannya aksi di  
monster management }
```

VII. Spesifikasi Program

A. F00 - Random Number Generator

```
import datetime
waktu_saat_ini ← call datetime.now()
detik ← floor((waktu_saat_ini - datetime(2024, 5, 17)).total_seconds())
// 1

function lcg(x, a, c, m)
  while True do
    x ← (a * x + c) mod m
    yield x

function random_uniform_sample(jml_num, batas, seed=detik)
  interval ← [0, batas]
  a, c, m ← 1103515245, 12345, 2 ** 31
  bsdrand ← call lcg(seed, a, c, m)

  lower, upper ← interval[0], interval[1]
  sample ← []

  for i ← 0 TO jml_num do
    observation ← (upper - lower) * (CALL bsdrand() / (2 ** 31 -
1)) + lower
    call sample.append(ROUND(observation))

  return sample
```

B. F01 - Register

```
if validUsername then
    dataUsername ← list of usernames from user starting from index 1 to the
length of user
    if username is in dataUsername then
        output "Username username sudah terpakai, silakan gunakan username
lain!"
        call register(user)
    else
        output ("Silakan pilih salah satu monster sebagai monster
awalmu.")
        output ("1. Charizard")
        output ("2. Bulbasaur")
        output ("3. Aspal")
        numMos ← integer value of user input "Monster pilihanmu : "
        Mos ← ""
        if numMos is equal to 1 then
            Mos ← "Charizard"
        elif numMos is equal to 2 then
            Mos ← "Bulbasaur"
        elif numMos is equal to 3 then
            Mos ← "Aspal"
        output ("Selamat datang Agent username. Mari kita mengalahkan Dr.
Asep Spakbor dengan Mos!")
        append [string value of (length of dataUsername + 1), username, pwd,
"agent", "0"] to user
    end if
else
    output ("Username hanya boleh berisi alfabet, angka, underscore, dan
strip!")
    call register(user)
end if
```

C. F02 - Login

```
login(user)

user_input ← input("Username: ")
pass_input ← input("Password: ")
user_idx ← 0
user_valid ← False

for i in range(len(user)) do
    if user[i][1] == user_input then
        user_valid ← True
        user_idx ← i
        break

if not user_valid then
    output("Username tidak terdaftar!")
    return False
elif user[user_idx][2] ≠ pass_input then
    output("Password salah!")
    return False
else
    output("Selamat datang, Agent", user_input + "!")
    output("Masukkan command 'help' untuk daftar command yang dapat kamu panggil.")
    return True
```

D. F03 - Logout

```
Algorithm Logout(log_status)
    if log_status then
        output("Logout gagal!")
        output("Anda belum login, silahkan login terlebih dahulu sebelum
melakukan logout")
        return ← True
    else
        output("Logout berhasil!")
        output("Silahkan login lagi jika ingin mengakses kembali
program")
        return ← False
    end if
End Algorithm
```

E. F04 - Menu

```
menu_help(u, ui, un, ls):
    output("===== HELP =====")
    if ls == False then
        output("Kamu belum login sebagai role apapun. Silahkan login
terlebih dahulu.")
        output()
        output("    1. Login: Masuk ke dalam akun yang sudah terdaftar")
        output("    2. Register: Membuat akun baru")
        return
    end if

    if u[ui][3] == "agent" then
        output(f"Halo Agent {un}. Kamu memanggil command HELP. Kamu
memilih jalan yang benar, semoga kamu tidak sesat kemudian. Berikut adalah
hal-hal yang dapat kamu lakukan sekarang:")
        output("    1. Logout: Keluar dari akun yang sedang digunakan")
        output("    2. Inventory: Melihat owca-dex yang dimiliki oleh
Agent")
    else
        output("Selamat datang, Admin. Berikut adalah hal-hal yang dapat
kamu lakukan:")
        output("    1. Logout: Keluar dari akun yang sedang digunakan")
```

```

        output("    2. Shop: Melakukan manajemen pada SHOP sebagai tempat
jual beli peralatan Agent")
    end if

    output()
    output("Footnote:")
    output("    1. Untuk menggunakan aplikasi, silahkan masukkan nama
fungsi yang terdaftar")
    output("    2. Jangan lupa untuk memasukkan input yang valid")
end menu_help

```

F. F07 - Inventory

```

Algorithm cek(a, user_idx)
    if a[0] = user_idx then
        return true
    else
        return false
    end if
end Algorithm

Algorithm pot_inventory(userinventory, user_idx)
    jumlahstr ← 0
    for i in userinventory do
        if i[1] = 'strengthpotion' and cek(i, user_idx) then
            jumlahstr ← i[2]
            break
        else
            jumlahstr ← 0
        end if
    end for

    jumlahres ← 0
    for i in userinventory do
        if i[1] = 'respotion' and cek(i, user_idx) then
            jumlahres ← i[2]
            break

```



```

        else
            jumlahres ← 0
        end if
    end for

    jumlahheal ← 0
    for i in userinventory do
        if i[1] = 'healingpotion' and cek(i, user_idx) then
            jumlahheal ← i[2]
            break
        else
            jumlahheal ← 0
        end if
    end for

    return [jumlahstr, jumlahres, jumlahheal]
end Algorithm

Algorithm pot_list(qty)
    pot_type ← ['atk', 'def', 'heal']
    pot_type_2 ← []
    pot_qty ← []

    for i from 0 to 2 do
        if qty[i] ≠ 0 then
            pot_type_2.append(pot_type[i])
            pot_qty.append(qty[i])
        end if
    end for

    return (pot_type_2, pot_qty)
end Algorithm

Algorithm monster_id(yourmonsterdata, user_idx)
    mons_id_array ← []
    for i in yourmonsterdata do
        if cek(i, user_idx) then

```

```

        mons_id_array.append(i[1])
    end if
end for
return mons_id_array
end Algorithm

Algorithm monster_lv(yourmonsterdata, user_idx)
    mons_lv_array ← []
    for i in yourmonsterdata do
        if cek(i, user_idx) then
            mons_lv_array.append(i[2])
        end if
    end for
    return mons_lv_array
end Algorithm

Algorithm inventory(userinventory, yourmonsterdata, monsterdata, coin,
user_idx)
    output ("jumlah owca coin anda saat ini : " + coin)
    while true do
        check ← input ("silakan pilih jenis item yang ingin diketahui
anda (monster/item/back): ")
        if check.lower() = 'item' then
            x ← 1
            types ← []
            for i in userinventory do
                if cek(i, user_idx) then
                    if i[1] = 'strength potion' then
                        type ← 'atk potion'
                    elif i[1] = 'resilience potion' then
                        type ← 'def potion'
                    elif i[1] = 'healing potion' then
                        type ← 'heal potion'
                    else
                        type ← "type"
                    end if
                    if type ≠ "type" then
                        types.append(type)

```

```

        output(x + ". type: " + type)
        x ← x + 1
    end if
end if
end for
while true do
    pot_number ← input("masukkan nomor potion untuk
menampilkan detail item (1/2/3/back): ")
    if pot_number.lower() ≠ 'back' and integer(pot_number) ≤
3 then
        pot_number ← integer(pot_number)
        output("type: " + types[pot_number - 1])
        output("quantity: " + userinventory[pot_number][2])
    else
        break
    end if
end while
else if check.lower() = 'monster' then
    y ← 0
    for i in yourmonsterdata do
        if cek(i, user_idx) then
            if i[1] ≠ "monster_id" then
                output ((y + 1) + ". " +
monsterdata[integer(i[1])][1])
                y ← y + 1
            end if
        end if
    end for
    while true do
        mons_number ← input("masukkan nomor monster untuk
menampilkan detail monster (1/2/dst/back): ")
        if mons_number.lower() ≠ 'back' then
            mons_number ← integer(mons_number)
            monsterlvl ←
integer(yourmonsterdata[mons_number][2])
            for i in monsterdata do
                if i[0]=yourmonsterdata[mons_number][1] then
                    output ("nama      : " + i[1])

```

G. F08 - Battle dan F06 - Potion

```
Algorithm battle(data, user_id)

    monster ← data["monster"]

    for i from 1 to length(monster) - 1 do
        monster[i][0] ← to_integer(monster[i][0])
        monster[i][2] ← to_integer(monster[i][2])
        monster[i][3] ← to_integer(monster[i][3])
        monster[i][4] ← to_integer(monster[i][4])
    end for

    monster_inventory ← data["monster_inventory"]
    for i from 1 to length(monster_inventory) - 1 do
        monster_inventory[i][0] ← to_integer(monster_inventory[i][0])
        monster_inventory[i][1] ← to_integer(monster_inventory[i][1])
        monster_inventory[i][2] ← to_integer(monster_inventory[i][2])
    end for

    item_inventory ← data["item_inventory"]
    for i from 1 to length(item_inventory) - 1 do
        item_inventory[i][0] ← to_integer(item_inventory[i][0])
        item_inventory[i][2] ← to_integer(item_inventory[i][2])
    end for

    os.system("cls")

    popped_monster_idx ← []
    user_monster ← []

    for i from 1 to length(monster_inventory) - 1 do
        if monster_inventory[i][0] = user_id then
            append(popped_monster_idx, i)
            monster_id ← monster_inventory[i][1]
            level ← monster_inventory[i][2]
            type_monster ← monster[monster_id + 1][1]
            atk_power ← monster[monster_id + 1][2]
            def_power ← monster[monster_id + 1][3]
            hp ← monster[monster_id + 1][4]
            user_monster_i ← {
```

```

        "monster_id" : monster_id,
        "level" : level,
        "type_monster" : type_monster,
        "atk_power" : atk_power,
        "def_power" : def_power,
        "hp" : hp,
        "hp_normal" : hp
    }
    append(user_monster, user_monster_i)
end if
end for

for i from length(popped_monster_idx) - 1 to 0 do
    pop(monster_inventory, i)
end for

popped_item_idx ← []
user_item ← []
for i from 1 to length(item_inventory) - 1 do
    if item_inventory[i][0] = user_id then
        append(popped_item_idx, i)
        user_item_i ← {
            "type": item_inventory[i][1],
            "quantity": item_inventory[i][2]
        }
        if user_item_i["type"] = "Strength Potion" then
            user_item_i["efect"] ← "Increases ATK Power"
            user_item_i["message1"] ← "Setelah meminum ramuan ini,
aura kekuatan terlihat mengelilingi "
            user_item_i["message2"] ← " dan gerakannya menjadi lebih
cepat dan mematikan."
        elif user_item_i["type"] = "Resilience Potion" then
            user_item_i["efect"] ← "Increases DEF Power"
            user_item_i["message1"] ← "Setelah meminum ramuan ini,
muncul sebuah energi pelindung di sekitar "
            user_item_i["message2"] ← " yang membuatnya terlihat
semakin tangguh dan sulit dilukai."
        elif user_item_i["type"] = "Healing Potion" then
            user_item_i["efect"] ← "Restores Health"

```

```

        user_item_i["message1"] ← "Setelah meminum ramuan ini,
luka-luka yang ada di dalam tubuh "

        user_item_i["message2"] ← " sembuh dengan cepat. Dalam
sekejap, dia terlihat kembali prima dan siap melanjutkan pertempuran."

    end if

    append(user_item, user_item_i)

end if

end for

for i from length(popped_item_idx) - 1 to 0 do
    pop(item_inventory, i)
end for

n ← random_uniform_sample(15) + 1
level_lawan ← random_uniform_sample(10) + 1
monster_lawan ← monster[n]
data_lawan ← {
    "monster_id" : monster_lawan[0],
    "level" : level_lawan,
    "type_monster" : monster_lawan[1],
    "atk_power" : monster_lawan[2],
    "def_power" : monster_lawan[3],
    "hp" : monster_lawan[4],
    "hp_normal" : monster_lawan[4]
}

output(_f"""
RAWRRR, Monster {data_lawan["type_monster"]} telah muncul !!!

Name      : {data_lawan["type_monster"]}
ATK Power : {data_lawan["atk_power"]}
DEF Power : {data_lawan["def_power"]}
HP        : {data_lawan["hp"]}
Level     : {data_lawan["level"]}
""")

output( "===== MONSTER LIST =====")

for i from 0 to length(user_monster) - 1 do
    output(_f"{i + 1}. {user_monster[i]['type_monster']}")

```

```

end for

choosen_monster_id ← 0
while true do
    choosen_monster_id ← to_integer(input("Pilih monster untuk
bertarung: "))
    if (0 < choosen_monster_id ≤ length(user_monster)) then
        break
    else
        output ("Pilihan nomor tidak tersedia!")
    end if
end while

using_monster ← user_monster[choosen_monster_id - 1]

output (f"""
RAWRRR, {data["user"][user_id][1]} mengeluarkan monster
{using_monster['type_monster']} !!!

Name      : {using_monster['type_monster']}
ATK Power : {using_monster['atk_power']}
DEF Power : {using_monster['def_power']}
HP        : {using_monster['hp']}
Level     : {using_monster['level']}
""")

lanjut ← input("Klik apapun untuk lanjut: ")

num ← 1
used_potion ← {
    "Strength Potion" : 0 ,
    "Resilience Potion" : 0,
    "Healing Potion" : 0
}

while true do
    os.system("cls")
    is_did_something ← 0
    output (f"""

```

```

=====
AYOO LAWAN MONSTER {data_lawan["type_monster"]} !!!

Name      : {data_lawan["type_monster"]}
ATK Power : {data_lawan["atk_power"]}
DEF Power : {data_lawan["def_power"]}
HP        : {data_lawan["hp"]}
Level     : {data_lawan["level"]}
""")

    output( f"""
=====

SEMANGAT MONSTER {using_monster['type_monster']} KAMU PASTI BISA!!!

Name      : {using_monster['type_monster']}
ATK Power : {using_monster['atk_power']}
DEF Power : {using_monster['def_power']}
HP        : {using_monster['hp']}
Level     : {using_monster['level']}
""")

    output (f"""
===== TURN {num} ({using_monster['type_monster']}) =====
1. Attack
2. Use Potion
3. Quit
""")

    turn ← 0
    while true do
        turn ← to_integer(input("Pilih perintah: "))
        If turn < 1 or turn > 3 then
            output( "Perintah tidak valid!")
        else
            break
        end if
    end while

    output ("\n")

```



```

    if turn = 3 then
        output( "Anda berhasil kabur dari BATTLE!")
        break
    elif turn = 2 then
        if length(user_item) = 0 then
            output( "Anda tidak memiliki potion dalam inventory!")
        else
            potion ← 0
            while true do
                output( "===== POTION LIST =====")
                for i from 0 to length(user_item) - 1 do
                    output( f"{i + 1}. {user_item[i]['type']} (Qty : {user_item[i]['quantity']}) - {user_item[i]['effect']}")
                end for
                output( f"{length(user_item) + 1}. Cancel")
                output( "\n")
                potion ← to_integer(input("Potion yang diinginkan:
"))

                if potion < 1 or potion > length(user_item) + 1 then
                    output( "Potion tidak valid!")
                elif potion = length(user_item) + 1 then
                    output( "Batal menggunakan potion.")
                    break

                elif used_potion[user_item[potion - 1]["type"]] = 1
and user_item[potion - 1]['type'] ≠ "Healing Potion" then
                    output( f"Kamu mencoba memberikan ramuan ini
kepada {using_monster['type_monster']}, namun dia menolaknya seolah-olah
dia memahami ramuan tersebut sudah tidak bermanfaat lagi.")
                elif user_item[potion - 1]["quantity"] = 0 then
                    output( "Wah, kamu sedang tidak memiliki ramuan
ini, silahkan pilih ramuan lain!")
                else
                    if user_item[potion - 1]["type"] = "Strength
Potion" then
                        user_item[potion - 1]["quantity"] ←
user_item[potion - 1]["quantity"] - 1

```

H. F09 - Arena

```
import os
from rng import random_uniform_sample

function toInteger(matrix)
  for i in range (len(matrix))do
    for j in range (len(matrix[i]))do
      if is_digit(str(matrix[i][j]))then
        matrix[i][j] = int(matrix[i][j])
    return matrix

function is_digit(word)
  result ← True
  for letter in word do
    if not (47 < ord(letter) < 58)then
      result ← False
      break
  return result

function arena(data, user_id)
  monster ← toInteger(data['monster'])
  monster_inventory ← toInteger(data['monster_inventory'])
  item_inventory ← toInteger(data['item_inventory'])

  popped_monster_idx ← []
  user_monster ← []
  for i in range (1, len(monster_inventory))do
    if monster_inventory[i][0] == user_id then
      popped_monster_idx.append(i)
      monster_id ← monster_inventory[i][1]
      level ← monster_inventory[i][2]
      type_monster ← monster[monster_id][1]
      atk_power ← monster[monster_id][2]
      def_power ← monster[monster_id][3]
      hp ← monster[monster_id][4]
      user_monster_i ← {
        "monster_id" : monster_id,
        "level" : level,
        "type_monster" : type_monster,
        "atk_power" : atk_power * ((level - 1) * 10) / 100,
        "def_power" : def_power * ((level - 1) * 10) / 100,
        "hp" : hp * ((level - 1) * 10) / 100,
        "hp_normal" : hp * ((level - 1) * 10) / 100
```

```

    }
    user_monster.append(user_monster_i)

popped_item_idx ← []
user_item ← []
for i in range (1, len(item_inventory))do
    if item_inventory[i][0] == user_id then
        popped_item_idx.append(i)
        user_item_i ← {
            "type": item_inventory[i][1],
            "quantity": item_inventory[i][2]
        }
        if user_item_i["type"] == "Strength Potion" then
            user_item_i["effect"] ← "Increases ATK Power"
            user_item_i["message1"] ← "Setelah meminum ramuan ini,
aura kekuatan terlihat mengelilingi "
            user_item_i["message2"] ← " dan gerakannya menjadi lebih
cepat dan mematikan."
            elif user_item_i["type"] == "Resilience Potion"then
                user_item_i["effect"] ← "Increases DEF Power"
                user_item_i["message1"] ← "Setelah meminum ramuan ini,
muncul sebuah energi pelindung di sekitar "
                user_item_i["message2"] ← " yang membuatnya terlihat
semakin tangguh dan sulit dilukai."
                elif user_item_i["type"] == "Healing Potion" then
                    user_item_i["effect"] ← "Restores Health"
                    user_item_i["message1"] ← "Setelah meminum ramuan ini,
luka-luka yang ada di dalam tubuh "
                    user_item_i["message2"] ← " sembuh dengan cepat. Dalam
sekejap, dia terlihat kembali prima dan siap melanjutkan pertempuran."
                    user_item.append(user_item_i)

os.system("cls")

output "Selamat datang di Arena!!"

output "===== MONSTER LIST ====="
for i in range (len(user_monster))do
    output f"{i+1}. {user_monster[i]['type_monster']}"

choosen_monster_id ← 0
while True do
    choosen_monster_id ← int(input("Pilih monster untuk bertarung: "))
    if (0 < choosen_monster_id ≤ len(user_monster))then

```

```

        break
    else
        output "Pilihan nomor tidak tersedia!"
        using_monster ← user_monster[choosen_monster_id - 1]

        output f"""
RAWRRR, {data["user"][user_id][1]} mengeluarkan monster
{using_monster['type_monster']} !!!

Name      : {using_monster['type_monster']}
ATK Power : {using_monster['atk_power']}
DEF Power : {using_monster['def_power']}
HP        : {using_monster['hp']}
Level     : {using_monster['level']}
"""

        lanjut ← input("Klik apapun untuk lanjut: ")

        stage ← 1
        num ← 1
        jumlah_hadiah_array ← [30, 50, 100, 170, 260]
        jumlah_hadiah ← 0
        while True do
            output f"===== STAGE {stage} ====="

            apakah_menang ← False
            used_potion ← {
                "Strength Potion" : 0,
                "Resilience Potion" : 0,
                'Healing Potion' : 0
            }

            n ← random_uniform_sample(15) + 1
            level_lawan ← stage
            monster_lawan ← monster[n]
            data_lawan ← {
                "monster_id" : monster_lawan[0],
                "level" : level_lawan,
                "type_monster" : monster_lawan[1],
                "atk_power" : monster_lawan[2] * (100 + (level_lawan - 1) *
10) / 100,
                "def_power" : monster_lawan[2] * (100 + (level_lawan - 1) *
10) / 100,
                "hp" : monster_lawan[4] * (100 + (level_lawan - 1) * 10) /

```

```

100,
    "hp_normal" : monster_lawan[4] * (100 + (level_lawan - 1) *
10) / 100
    }

    output f"""
RAWRRR, Monster {data_lawan["type_monster"]} telah muncul !!!

Name      : {data_lawan["type_monster"]}
ATK Power : {data_lawan["atk_power"]}
DEF Power : {data_lawan["def_power"]}
HP        : {data_lawan["hp"]}
Level     : {data_lawan["level"]}
"""

    output f"""
===== TURN {num} ({using_monster['type_monster']}) =====
1. Attack
2. Use Potion
3. Quit
"""

    turn ← 0
    while True do
        turn ← int(input("Pilih perintah: "))
        if turn < 1 or turn > 3 then
            output "Perintah tidak valid!"
            turn ← int(input("Pilih perintah: "))
        else
            break
    output "\n"
    if turn == 3 then
        output "Anda berhasil kabur dari BATTLE!"
        break
    elif turn == 2 then
        if len(user_item) == 0 then
            output "Anda tidak memiliki potion dalam inventory!"
        else
            potion ← 0
            while True do
                output "===== POTION LIST ====="
                for i in range(len(user_item))
                    output f"{i+1}. {user_item[i]['type']} (Qty :
{user_item[i]['quantity']}) - {user_item[i]['efect']}
                output f"{len(user_item) + 1}. Cancel"

```

```

        output "\n"
        potion ← int(input("Potion yang diinginkan: "))
        if potion < 1 or potion > len(user_item) + 1 then
            output "Potion tidak valid!"
        elif potion == len(user_item) + 1 then
            output "Batal menggunakan potion."
            break
        elif used_potion[user_item[potion - 1]["type"]] == 1
and user_item[potion - 1]['type'] ≠ "Healing Potion" then
            output f"Kamu mencoba memberikan ramuan ini kepada
{using_monster['type_monster']}, namun dia menolaknya seolah-olah dia
memahami ramuan tersebut sudah tidak bermanfaat lagi."
        elif user_item[potion - 1]["quantity"] == 0 then
            output "Wah, kamu sedang tidak memiliki ramuan
ini, silahkan pilih ramuan lain!"
        else
            if user_item[potion - 1]["type"] == "Strength
Potion" then
                user_item[potion - 1]["quantity"] -= 1
                used_potion["Strength Potion"] = 1
                output f"{user_item[potion -
1]['message1']}{using_monster['type_monster']}{user_item[potion -
1]['message2']}\n"
            elif user_item[potion - 1]["type"] == "Resilience
Potion" then
                user_item[potion - 1]["quantity"] -= 1
                used_potion["Resilience Potion"] = 1
                output f"{user_item[potion -
1]['message1']}{using_monster['type_monster']}{user_item[potion -
1]['message2']}\n"
            elif user_item[potion - 1]["type"] == "Healing
Potion" then
                user_item[potion - 1]["quantity"] -= 1
                using_monster['hp'] =
using_monster['hp_normal']
                output f"{user_item[potion -
1]['message1']}{using_monster['type_monster']}{user_item[potion -
1]['message2']}\n"
                is_did_something = 1
                break

        elif turn == 1 then
            if used_potion["Strength Potion"] == 1 then
                random_attack = random_uniform_sample(60

```

I. F10 - Shop & Currency

```
function toInteger(matrix)
  for i from 0 to length(matrix) - 1 do
    for j from 0 to length(matrix[i]) - 1 do
      if is_digit(string(matrix[i][j])) then
        matrix[i][j] ← int(matrix[i][j])
  return matrix

function is_digit(word)
  result ← true
  for letter in word do
    if not (47 < ord(letter) < 58) then
      result ← false
      break
  return result

function isIn(x, A)
  res ← false
  for _ in A do
    if _ == x then
      res ← true
      break
  return res

function itsIndex(x, A)
  for i from 0 to length(A) - 1 do
    if A[i] == x then
      return i

function shop_currency(data, user_id)
  user ← toInteger(data['user'])
  monster ← toInteger(data['monster'])
  monster_inventory ← toInteger(data['monster_inventory'])
  monster_shop ← toInteger(data['monster_shop'])
  item_inventory ← toInteger(data['item_inventory'])
  item_shop ← toInteger(data['item_shop'])

  // {USER INVENTORY THINGS}
```

```

popped_monster_inventory ← []
user_monster ← []
for i from 1 to length(monster_inventory) - 1 do
    if monster_inventory[i][0] == user_id then
        append popped_monster_inventory with i
        id ← monster_inventory[i][1]
        type_ ← monster[id][1]
        level ← monster_inventory[i][2]
        user_monster_i ← {
            "type" : type_,
            "level" : level
        }
        append user_monster with user_monster_i

popped_user_item ← []
user_item ← []
for i from 1 to length(item_inventory) - 1 do
    if item_inventory[i][0] == user_id then
        append popped_user_item with i
        type_ ← item_inventory[i][1]
        quantity ← item_inventory[i][2]
        user_item_i ← {
            "type" : type_,
            "quantity" : quantity
        }
        append user_item with user_item_i

username ← user[user_id][1]
oc ← user[user_id][4]

user_inventory ← {
    "id" : user_id,
    "username" : username,
    "oc" : oc,
    "user_monster" : user_monster,
    "user_item" : user_item
}
// {END USER INVENTORY THINGS}

// {DISPLAY MONSTER THINGS}
popped_monster_shop ← []
display_monster ← []
for i from 1 to length(monster_shop) - 1 do
    append popped_monster_shop with i

```



```

        id ← monster_shop[i][0]
        stock ← monster_shop[i][1]
        price ← monster_shop[i][2]
        type_ ← monster[id][1]
        atk_power ← monster[id][2]
        def_power ← monster[id][3]
        hp ← monster[id][4]
        display_monster_i ← {
            "id" : id,
            "stock" : stock,
            "price" : price,
            "type" : type_,
            "atk_power" : atk_power,
            "def_power" : def_power,
            "hp" : hp
        }
        append display_monster with display_monster_i
    // {END DISPLAY MONSTER THINGS}

// {DISPLAY ITEM THINGS}
popped_item_shop ← []
display_item ← []
for i from 1 to length(item_shop) - 1 do
    append popped_item_shop with i
    id ← i
    type_ ← item_shop[i][0]
    stock ← item_shop[i][1]
    price ← item_shop[i][2]
    display_item_i ← {
        "id" : id,
        "type" : type_,
        "stock" : stock,
        "price" : price
    }
    append display_item with display_item_i
// {END DISPLAY ITEM THINGS}

// {START PROGRAM}
os.system("cls")
output "Irasshaimase! Selamat datang di SHOP!!"
while true do
    aksi ← input("Mau apa (lihat/beli/keluar) :")
    if aksi == "lihat" then
        lihat_apa ← ""

```

```

while true do
    lihat_apa ← input("Lihat apa (monster/item): ")
    if lihat_apa == "monster" or lihat_apa == "item" then
        break
    else
        output "Inputan salah!"

    if lihat_apa == "monster" then
        output "ID | Type | ATK Power | DEF Power | HP
| Stok | Harga"
        for item in display_monster do
            output item['id'] + " | " + item['type'] + " | " +
item['atk_power'] + " | " + item['def_power'] + " | " + item['hp'] + " |
" + item['stock'] + " | " + item['price']
        else if lihat_apa == "item" then
            output "ID | Type | Stok | Harga"
            for item in display_item do
                output item['id'] + " | " + item['type'] + " | " +
item['stock'] + " | " + item['price']
            else if aksi == "beli" then
                output "Jumlah OWCA coinmu sekarang adalah: " +
user_inventory['oc']
                beli_apa ← ""
                while true do
                    beli_apa ← input("Mau beli apa? (monster/item): ")
                    if beli_apa == "monster" or beli_apa == "item" then
                        break
                    else
                        output "Inputan salah!"
                if beli_apa == "monster" then
                    id_monster_dibeli ← int(input("Masukkan id monster: "))
                    monster_dibeli ← display_monster[id_monster_dibeli - 1]
                    user_monster_type ← []
                    for i from 0 to length(user_inventory["user_monster"]) -
1 do
                        append user_monster_type with
user_inventory["user_monster"][i]["type"]
                    if monster_dibeli['price'] > user_inventory['oc'] then
                        output "OC-mu tidak cukup."
                    else if isIn(monster_dibeli['type'], user_monster_type)
then
                        output "Monster " + monster_dibeli['type'] + " sudah
ada dalam inventory-mu! Pembelian dibatalkan."
                    else

```

```

        output "Berhasil membeli monster: " +
monster_dibeli['type'] + ". Item sudah masuk ke inventory-mu!"
        display_monster[id_monster_dibeli - 1]["stock"] ←
display_monster[id_monster_dibeli - 1]["stock"] - 1
        type_monster_dibeli ← {
            "type" : monster_dibeli['type'],
            "level" : 1
        }
        append user_inventory['user_monster'] with
type_monster_dibeli
        user_inventory['oc'] ← user_inventory['oc'] -
monster_dibeli['price']

    else if beli_apa == "item" then
        id_item_dibeli ← int(input("Masukkan id potion: "))
        jumlah_potion ← int(input("Masukkan jumlah: "))
        item_dibeli ← display_item[id_item_dibeli - 1]
        if item_dibeli['price'] * jumlah_potion >
user_inventory['oc'] then
            output "OC-mu tidak cukup."
        else
            output "Berhasil membeli item: " + jumlah_potion + "
" + item_dibeli['type'] + ". Item sudah masuk ke inventory-mu!"
            display_item[id_item_dibeli - 1]["stock"] ←
display_item[id_item_dibeli - 1]["stock"] - 1
            belum_tersimpan ← true
            for i from 0 to length(user_inventory['user_item']) -
1 do
                if user_inventory['user_item'][i]['type'] ==
item_dibeli['type'] then
                    user_inventory['user_item'][i]['quantity'] ←
user_inventory['user_item'][i]['quantity'] + 1
                    output user_inventory['user_item']
                    belum_tersimpan ← false
                if belum_tersimpan then
                    type_item_dibeli ← {
                        'type' : item_dibeli['type'],
                        'quantity' : 1
                    }
                    append user_inventory['user_item'] with
type_item_dibeli
                    user_inventory['oc'] ← user_inventory['oc'] -
item_dibeli['price']

```

```

        else if aksi == "keluar" then
            output "Mr. Yanto bilang makasih, belanja lagi ya nanti :)"
            break
        else
            output "Inputan salah!"

for i from length(popped_user_item) - 1 to 0 do
    item_inventory.pop(i)

for i from length(popped_monster_inventory) - 1 to 0 do
    monster_inventory.pop(i)

for i from length(popped_monster_shop) - 1 to 0 do
    monster_shop.pop(i)

for i from length(popped_item_shop) - 1 to 0 do
    item_shop.pop(i)

// {ITEM INVENTORY}
output user_inventory['user_item']
for i from 0 to length(user_inventory['user

```

J. F11 - Laboratory

```
function monster_upgrade(monsters, idx, owca_coin)

    pilihan_monster ← monsters[idx]

    sudah_upgrade ← false

    while not sudah_upgrade do

        if pilihan_monster['level'] ≥ 5 then

            output( "Maaf, monster yang Anda pilih sudah memiliki level maksimum.")

            break

        else

            upgrade_harga_arr ← [300, 500, 800, 1000]

            level_now ← pilihan_monster['level']

            if level_now ≤ 4 then

                upgrade_harga ← upgrade_harga_arr[level_now - 1]

                output "\n" + pilihan_monster['name'] + " akan di-upgrade ke level " + (pilihan_monster['level'] + 1) + "."

                output "Harga untuk melakukan upgrade " + pilihan_monster['name'] + " adalah " + upgrade_harga + " OC."

                if owca_coin ≥ upgrade_harga then

                    confirm ← input(">>> Lanjutkan upgrade (Y/N): ")

                    if confirm in ['Y', 'y'] then

                        pilihan_monster['level'] ← pilihan_monster['level'] + 1

                        owca_coin ← owca_coin - upgrade_harga

                        output ("Selamat, " + pilihan_monster['name'] + " berhasil di-upgrade ke level " + pilihan_monster['level'] + "!")

                        sudah_upgrade ← true

                    elif confirm in ['N', 'n'] then

                        output ("Upgrade dibatalkan.")

                        break

                    else

                        output( "Pilihan tidak valid. Silakan masukkan Y atau N.")

                else

                    output ("Maaf, OC Anda tidak mencukupi untuk melakukan upgrade.")

                    break

            else

                output ("Maaf, monster sudah mencapai level maksimum.")

                break
```

```

    return owca_coin

function laboratory(monsters, owca_coins)

    while true do
        output("\\n"
                "Selamat datang di Lab Dokter Asep !!!\\n"
                "\\n"
                "===== MONSTER LIST =====")
        for ix, monster in enumerate(monsters, 1) do
            output(ix + ". " + monster['name'] + " (Level: " +
monster['level'] + ")")

        output("\\n===== UPGRADE PRICE =====")
            "1. Level 1 -> Level 2: 300 OC"
            "2. Level 2 -> Level 3: 500 OC"
            "3. Level 3 -> Level 4: 800 OC"
            "4. Level 4 -> Level 5: 1000 OC")

        pilih ← input(">>> Pilih monster (atau keluar): ")
        if pilih == 'keluar' then
            output( "Terima kasih sudah datang ke Lab Dokter Asep.")
            break
        elif pilih.isdigit() then
            pilih ← integer value of pilih
            if 1 ≤ pilih ≤ length of monsters then
                owca_coins ← monster_upgrade(monsters, pilih - 1,
owca_coins)
            else
                output( "Pilihan tidak valid, silakan pilih nomor monster
yang valid.")
            else
                output( "Pilihan tidak valid, silakan pilih nomor monster
yang valid.")

```

K. F13 - Monster Management

```
monster ← [  
  {"ID": 1, "TYPE": "Snorlax", "ATK": 60, "DEF": 66, "HP": 43},  
  {"ID": 2, "TYPE": "Pikachu", "ATK": 52, "DEF": 53, "HP": 40},  
  {"ID": 3, "TYPE": "Goku", "ATK": 54, "DEF": 63, "HP": 45},  
  {"ID": 4, "TYPE": "Bejita", "ATK": 46, "DEF": 47, "HP": 40},  
  {"ID": 5, "TYPE": "Naruto", "ATK": 62, "DEF": 50, "HP": 29},  
]  
  
// Function to remove spaces from a string  
function mstrip(str)  
  nstring ← ""  
  for char in str do  
    if char ≠ " " then  
      nstring ← nstring + char  
  return nstring  
  
// Function to split a string by a separator  
function split(str, separator)  
  hasil ← []  
  kata ← ''  
  for char in str do  
    if char == separator then  
      append hasil with kata  
      kata ← ''  
    else  
      kata ← kata + char  
  append hasil with kata  
  return hasil  
  
// Function to check if a string represents an integer  
function is_integer(s)  
  digits ← "0123456789"  
  for char in s do  
    if char not in digits then  
      return false  
  return true  
  
// Function to generate a new ID  
function id_baru(monster)  
  max_id ← -1  
  for creature in monster do  
    if creature["ID"] > max_id then  
      max_id ← creature["ID"]
```

```

    return max_id + 1

// Function to check if a monster type/name already exists
function cek_sama(nama, dict)
    for row in dict do
        if lower(row["TYPE"]) == lower(nama) then
            return false
    return true

// Function for monster management
function monster_management()
    output ""
    output "SELAMAT DATANG DI DATABASE PARA MONSTER !!!"
    output ""
    running ← true
    while running == true do
        output ""
        output "1. Tampilkan semua monster"
        output "2. Tambah monster baru"
        output "3. Keluar"
        output ""
        input_1 ← int(input("Pilih Aksi : "))
        output ""
        if input_1 == 3 then
            output "Selamat tinggal!"
            break
        else if input_1 == 1 then
            categories ← ["ID", "TYPE", "ATK", "DEF", "HP"]
            header ← ""
            for category in categories do
                header ← header + category + " " * (10 - len(category))
            output header
            output "-" * (len(header) + 5)
            for creature in monster do
                data_row ← ""
                for category in categories do
                    data_row ← data_row + str(creature[category]) + " " *
(10 - len(str(creature[category])))
                output data_row
        else if input_1 == 2 then
            output "Memulai pembuatan monster baru..."
            valid ← false
            while valid == false do
                new_monster ← input("Masukkan Type>Nama : ")

```



```

        if cek_sama(new_monster, monster) == false then
            output "Monster sudah terdaftar di database!"
        else if new_monster == "" then
            output "Masukkan input yang valid"
        else
            valid ← true
valid_2 ← false
while valid_2 == false do
    atk_power ← str(input("Masukkan ATK Power : "))
    if atk_power == "" then
        output "input tidak valid, coba lagi!"
    else if is_integer(atk_power) == false then
        output "Masukkan input bertipe Integer, coba lagi!"
    else if is_integer(atk_power) == true then
        valid_2 ← true
valid_3 ← false
while valid_3 == false do
    def_power ← str(input("Masukkan DEF Power: "))
    if def_power == "" then
        output "input tidak valid, coba lagi!"
    else if is_integer(def_power) == false then
        output "Masukkan input bertipe Integer, coba lagi!"
    else if is_integer(def_power) == true then
        if int(def_power) < 0 or int(def_power) > 50 then
            output "DEF Power harus bernilai 0-50, coba
lagi!"
        else
            valid_3 ← true
valid_4 ← false
while valid_4 == false do
    hp ← str(input("Masukkan HP: "))
    if hp == "" then
        output "input tidak valid, coba lagi!"
    else if is_integer(hp) == false then
        output "Masukkan input bertipe integer, coba lagi!"
    else if is_integer(hp) == true then
        valid_4 ← true
output "Name/Type    | ATK Power | DEF Power | HP"
output new_monster + " " * (12 - len(new_monster)) + " | " +
atk_power + " " * (10 - len(atk_power)) + " | " + def_power + " " * (10 -
len(def_power)) + " | " + hp
inp2 ← input("Tambahkan Monster ke Database (Y/N) : ")
if lower(inp2) == 'y' then
    new_id ← id_baru(monster)

```

```

        append monster with {"ID": new_id, "TYPE": new_monster,
"ATK": int(atk_power), "DEF": int(def_power), "HP": int(hp)}
        output "Monster baru telah ditambahkan!"
    else if lower(inp2) == 'n' then
        output "Monster gagal ditambahkan!"

output monster_management()

```

L. F14 - Load

```

Algorithm load()

Function parse(file)
    matrix ← empty list
    for each line in file do
        if line = "99999" then
            continue
        else
            array ← empty list
            data ← empty string
            for each character _ in line do
                if _ ≠ ";" then
                    data ← data concatenated with _
                else
                    append data to array
                    data ← empty string
            data ← data without last character
            append data to array
            append array to matrix
    output (matrix)
    return matrix
End Function parse

dict ← empty dictionary

```

```

    parser ← create.ArgumentParser object with description "Load data
from external files."

    add argument "folder_name" to parser with type string and nargs "?"
and help "Nama folder yang berisi file penyimpanan."

    args ← parse arguments using parser

    folder_name ← folder_name from args

    if folder_name is None then
        output ("Tidak ada nama folder yang diberikan!")
        output ("Usage: python main.py <nama_folder>")
        return None

    elif not exists folder_name then
        output ("Folder 'folder_name' tidak ditemukan.")
        return None

    else
        output ("Loading...")

        dict['folder_name'] ← folder_name

        with open(join(folder_name, "user.csv"), "r") as file do
            dict['user'] ← parse(file)

        with open(join(folder_name, "monster.csv"), "r") as file do
            dict['monster'] ← parse(file)

        with open(join(folder_name, "item_inventory.csv"), "r") as file
do
            dict['item_inventory'] ← parse(file)

        with open(join(folder_name, "monster_inventory.csv"), "r") as
file do
            dict['monster_inventory'] ← parse(file)

        with open(join(folder_name, "monster_shop.csv"), "r") as file do
            dict['monster_shop'] ← parse(file)

        with open(join(folder_name, "item_shop.csv"), "r") as file do
            dict['item_shop'] ← parse(file)

        output ("Selamat datang di program OWCA!")

        return dict

End Algorithm load

```

M. F15 - Save

```
import os

function save(data)

    function save_csv(folder_path, type_csv, data)
        file_path ← os.path.join(folder_path, type_csv)
        file ← open(file_path, "w")
        len_data ← len(data)
        for j in range(len_data)
            line ← ""
            len_row ← len(data[j])
            for i in range(len_row)
                line += str(data[j][i])
                if i ≠ len_row - 1
                    line += ";"
            else
                line += "\n"
            if j == len_data - 1
                line += "99999"
            file.write(line)
        file.close()

    folder_name ← input("Masukkan nama folder: ")

    if not os.path.exists("data")
        output "Membuat folder data..."
        os.makedirs("data")
    folder_path ← os.path.join("data", folder_name)
    if not os.path.exists(folder_path)
        os.makedirs(folder_path)
        output f"Membuat folder {folder_path}..."

    save_csv(folder_path, "user.csv", data["user"])
    save_csv(folder_path, "monster.csv", data["monster"])
    save_csv(folder_path, "item_inventory.csv", data["item_inventory"])
    save_csv(folder_path, "monster_inventory.csv",
data["monster_inventory"])
    save_csv(folder_path, "monster_shop.csv", data["monster_shop"])
    save_csv(folder_path, "item_shop.csv", data["item_shop"])

    output f"Berhasil menyimpan data di folder {folder_path}!"
```

N. F16 - Exit

```
Algorithm exit(data)
    output ("Apakah Anda mau melakukan penyimpanan file yang sudah
diubah? (y/n) ")
    yesOrNo ← input()
    if yesOrNo = "y" then
        save(data)
    elif yesOrNo ≠ "n" then
        exit(data)
    end if
End Algorithm
```

VIII. HASIL PENGUJIAN

```
Selamat datang di program OWCA!  
1. Login  
2. Register  
3. Help  
>>> Pilih command yang akan kamu pilih : █
```

Gambar 7.1 Tampilan Awal

```
Selamat datang di program OWCA!  
1. Login  
2. Register  
3. Help  
>>> Pilih command yang akan kamu pilih : 1  
[['id', 'username', 'password', 'role', 'oc'], ['0', 'NA', 'NA', 'NA', '0'], ['1', 'bill_2231', 'fsgdgrgh', 'agent', '0'], ['2', 'bill_2232', 'fsgxdgb', 'agent', '0'], ['3', 'bill_2233', 'cvhg', 'agent', '0'], ['4', 'bill_2234', 'dfhg', 'agent', '0'], ['5', 'bill_2235', 'fsgdgh', 'agent', '0'], ['6', 'bill_2236', 'dfhgyf', 'agent', '0'], ['7', 'bill_2237', 'dhfghg', 'agent', '0'], ['8', 'bill_2238', 'dgxfcgb', 'agent', '0'], ['9', 'bill_2239', 'fhjchcvb', 'agent', '0'], ['10', 'yaz_2324', 'abcd', 'agent', '100'], ['11', 'admin1', 'admin1', 'admin', '0'], ['12', 'admin2', 'admin2', 'admin', '0'], ['13', 'admin3', 'admin3', 'admin', '0'], ['14', 'a', 'a', 'agent', '1000']]  
Username: █
```

Gambar 7.2 Saat pilih login

```
Selamat datang di program OWCA!  
1. Login  
2. Register  
3. Help  
>>> Pilih command yang akan kamu pilih : 1  
[['id', 'username', 'password', 'role', 'oc'], ['0', 'NA', 'NA', 'NA', '0'], ['1', 'bill_2231', 'fsgdgrgh', 'agent', '0'], ['2', 'bill_2232', 'fsgxdgb', 'agent', '0'], ['3', 'bill_2233', 'cvhg', 'agent', '0'], ['4', 'bill_2234', 'dfhg', 'agent', '0'], ['5', 'bill_2235', 'fsgdgh', 'agent', '0'], ['6', 'bill_2236', 'dfhgyf', 'agent', '0'], ['7', 'bill_2237', 'dhfghg', 'agent', '0'], ['8', 'bill_2238', 'dgxfcgb', 'agent', '0'], ['9', 'bill_2239', 'fhjchcvb', 'agent', '0'], ['10', 'yaz_2324', 'abcd', 'agent', '100'], ['11', 'admin1', 'admin1', 'admin', '0'], ['12', 'admin2', 'admin2', 'admin', '0'], ['13', 'admin3', 'admin3', 'admin', '0'], ['14', 'a', 'a', 'agent', '1000']]  
Username: a  
Password: a  
Selamat datang, Agent a!  
  
Loading...  
  
Masukkan command "Help" untuk daftar command yang dapat kamu panggil.  
Masukkan command "Exit" untuk keluar dari program.  
  
>>> Masukkan command : █
```

Gambar 7.3 Saat pilih login lalu berhasil login

```

Selamat datang di program OWCA!
1. Login
2. Register
3. Help
>>> Pilih command yang akan kamu pilih : 2
Masukkan username: sss
Masukkan password: qwerty123
Silakan pilih salah satu monster sebagai monster awalmu.
1. Charizard
2. Bulbasaur
3. Aspal
Monster pilihanmu : █

```

Gambar 7.4 Saat pilih register pada lalu berhasil register

```

Selamat datang, Agent a!

Loading...

Masukkan command "Help" untuk daftar command yang dapat kamu panggil.
Masukkan command "Exit" untuk keluar dari program.

>>> Masukkan command : help
===== HELP =====
Halo Agent a. Kamu memanggil command HELP. Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian. Berikut a
adalah hal-hal yang dapat kamu lakukan sekarang:
1. Logout: Keluar dari akun yang sedang digunakan
2. Inventory: Lihat OWCA-dex dan peralatanmu, Agent a
3. Laboratory: Upgrade monster yang kamu miliki, Agent a!
4. Battle : Ayo bertarung di Arena, Agent a dan menangkan hadiahnya!
5. Arena : Ayo masuk ke Arena, Agent a dan latih monstermu!
6. Shop and Currency : Ayo belanjaan OWCA-mu dengan potion dan monster di sini, Agent a!

Footnote:
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
2. Jangan lupa untuk memasukkan input yang valid

>>> Pilih fungsi : █

```

Gambar 7.5 Saat sudah ada user lalu masukkan command help

```

Selamat datang di program OWCA!
1. Login
2. Register
3. Help
>>> Pilih command yang akan kamu pilih : 2
Masukkan username: sss
Masukkan password: qwerty123
Silakan pilih salah satu monster sebagai monster awalmu.
1. Charizard
2. Bulbasaur
3. Aspal
Monster pilihanmu : 1
Selamat datang Agent sss. Mari kita mengalahkan Dr. Asep Spakbor dengan Charizard!
Masukkan nama folder: 356
Membuat folder data\356...
Berhasil menyimpan data di folder data\356!

Loading...

Masukkan command "Help" untuk daftar command yang dapat kamu panggil.
Masukkan command "Exit" untuk keluar dari program.

>>> Masukkan command : █

```

Gambar 7.6 Saat save folder

```

Masukkan command "Help" untuk daftar command yang dapat kamu panggil.
Masukkan command "Exit" untuk keluar dari program.

>>> Masukkan command : help
===== HELP =====
Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu.

1. Login: Masuk ke dalam akun yang sudah terdaftar
2. Register: Membuat akun baru

>>> Pilih fungsi : █

```

Gambar 7.7 Tampilan dasar menu help sebelum register/login

```

Selamat datang di program OWCA!
1. Login
2. Register
3. Help
>>> Pilih command yang akan kamu pilih : 3
===== HELP =====
Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu.

1. Login: Masuk ke dalam akun yang sudah terdaftar
2. Register: Membuat akun baru

>>> Pilih fungsi : █

```


Gambar 7.8 Tampilan dasar menu help saat masuk dari main menu

```
===== HELP =====
Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu.

1. Login: Masuk ke dalam akun yang sudah terdaftar
2. Register: Membuat akun baru

>>> Pilih fungsi : 1
[[['id', 'username', 'password', 'role', 'oc'], ['0', 'NA', 'NA', 'NA', '0'], ['1', 'bill_2231', 'fsdgdgrgh', 'agent',
'0'], ['2', 'bill_2232', 'fsgxdgb', 'agent', '0'], ['3', 'bill_2233', 'cvhg', 'agent', '0'], ['4', 'bill_2234', 'dfhg',
'agent', '0'], ['5', 'bill_2235', 'fsdgdgh', 'agent', '0'], ['6', 'bill_2236', 'dfhgyf', 'agent', '0'], ['7', 'bil
_2237', 'dhfghg', 'agent', '0'], ['8', 'bill_2238', 'dgxfcgb', 'agent', '0'], ['9', 'bill_2239', 'fhjchcvb', 'agent',
'0'], ['10', 'yaz_2324', 'abcd', 'agent', '100'], ['11', 'admin1', 'admin1', 'admin', '0'], ['12', 'admin2', 'admi
n2', 'admin', '0'], ['13', 'admin3', 'admin3', 'admin', '0'], ['14', 'a', 'a', 'agent', '1000']]
Username: █
```

Gambar 7.9 Tampilan awal login

```
>>> Masukkan command : help
===== HELP =====
Halo Agent a. Kamu memanggil command HELP. Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian. Berikut a
dahlah hal-hal yang dapat kamu lakukan sekarang:
1. Logout: Keluar dari akun yang sedang digunakan
2. Inventory: Lihat OWCA-dex dan peralatanmu, Agent a
3. Laboratory: Upgrade monster yang kamu miliki, Agent a!
4. Battle : Ayo bertarung di Arena, Agent a dan menangkan hadiahnya!
5. Arena : Ayo masuk ke Arena, Agent a dan latih monstermu!
6. Shop and Currency : Ayo belanjaan OWCA-mu dengan potion dan monster di sini, Agent a!

Footnote:
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
2. Jangan lupa untuk memasukkan input yang valid

>>> Pilih fungsi : 1
Logout berhasil!

Silahkan login lagi jika ingin mengakses kembali program
1. Login
2. Register
3. Help
>>> Pilih command yang akan kamu pilih : █
```

Gambar 7.10 Saat logout berhasil

```
>>> Masukkan command : help
===== HELP =====
Halo Agent a. Kamu memanggil command HELP. Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian. Berikut a
dahlah hal-hal yang dapat kamu lakukan sekarang:
1. Logout: Keluar dari akun yang sedang digunakan
2. Inventory: Lihat OWCA-dex dan peralatanmu, Agent a
3. Laboratory: Upgrade monster yang kamu miliki, Agent a!
4. Battle : Ayo bertarung di Arena, Agent a dan menangkan hadiahnya!
5. Arena : Ayo masuk ke Arena, Agent a dan latih monstermu!
6. Shop and Currency : Ayo belanjaan OWCA-mu dengan potion dan monster di sini, Agent a!

Footnote:
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
2. Jangan lupa untuk memasukkan input yang valid

>>> Pilih fungsi : 2
jumlah OWCA coin anda saat ini : 1000
Silakan pilih jenis item yang ingin diketahui Anda (Monster/Item/Back): █
```

Gambar 7.11 Saat pilih inventory pada help

```
>>> Pilih fungsi : 2
jumlah OWCA coin anda saat ini : 1000
Silakan pilih jenis item yang ingin diketahui Anda (Monster/Item/Back): Monster
1. Sanyaz
2. Charizard
3. Onix
Masukkan nomor monster untuk menampilkan detail monster (1/2/dst/Back):
```

Gambar 7.12 Monster inventory

```
Masukkan nomor monster untuk menampilkan detail monster (1/2/dst/Back): 1
Nama      : Sanyaz
ATK Power : 1098.9
DEF Power : 1098.9
HP        : 1098.9
Level     : 3
Masukkan nomor monster untuk menampilkan detail monster (1/2/dst/Back):
```

Gambar 7.13 Detail monster

```
Masukkan nomor monster untuk menampilkan detail monster (1/2/dst/Back): Back
Silakan pilih jenis item yang ingin diketahui Anda (Monster/Item/Back):
```

Gambar 7.14 Saat back kembali ke menu awal inventory

```
Silakan pilih jenis item yang ingin diketahui Anda (Monster/Item/Back): Item
1. Type: ATK potion
2. Type: HEAL potion
3. Type: DEF potion
Masukkan nomor potion untuk menampilkan detail item (1/2/3/Back):
```

Gambar 7.15 Inventory item

```
Silakan pilih jenis item yang ingin diketahui Anda (Monster/Item/Back): Item
1. Type: ATK potion
2. Type: HEAL potion
3. Type: DEF potion
Masukkan nomor potion untuk menampilkan detail item (1/2/3/Back): 1
Type: ATK potion
Quantity: 10
Masukkan nomor potion untuk menampilkan detail item (1/2/3/Back):
```

Gambar 7.16 Detail potion

```

Masukkan nomor potion untuk menampilkan detail item (1/2/3/Back): Back
Silakan pilih jenis item yang ingin diketahui Anda (Monster/Item/Back): Back

Masukkan command "Help" untuk daftar command yang dapat kamu panggil.
Masukkan command "Exit" untuk keluar dari program.

>>> Masukkan command : 

```

Gambar 7.17 Keluar dari fungsi inventory

```

>>> Pilih fungsi : 3

Selamat datang di Lab Dokter Asep !!!

===== MONSTER LIST =====
1. Sanyaz : level 3
2. Charizard : level 2
3. Onix : level 1

===== UPGRADE PRICE =back=====
1. Level 1 -> Level 2: 300 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1000 OC
>>> Pilih monster (atau keluar): 

```

Gambar 7.18 Tampilan laboratory

```

RAWRRR, Monster Sanyaz telah muncul !!!

Name      : Sanyaz
ATK Power : 999.0
DEF Power : 999.0
HP        : 999.0
Level     : 1

===== MONSTER LIST =====
1. Sanyaz
2. Charizard
3. Onix
Pilih monster untuk bertarung: 

```

Gambar 7.19 Tampilan battle awal dengan monster

```
=====
SEMANGAT MONSTER Sanyaz KAMU PASTI BISA !!!

Name      : Sanyaz
ATK Power : 1198.8
DEF Power : 1198.8
HP        : 1198.8
Level     : 3

===== TURN 1 (Sanyaz) =====
1. Attack
2. Use Potion
3. Quit

Pilih perintah: 1

Attack berkurang -899.0% karena DEF Power: -10777.212
Sisa HP Sanyaz adalah 11776.0
===== TURN 1 (Sanyaz) =====
Damage berkurang -1098.8% karena DEF Power: -10969.02
Sisa HP Sanyaz adalah 12167.0
Klik apapun untuk lanjut: █
```

Gambar 7.20 Ketika saat battle fungsi attack

```
Pilih perintah: 2

===== POTION LIST =====
1. Strength Potion (Qty : 10) - Increases ATK Power
2. Healing Potion (Qty : 15) - Restores Health
3. Resilience Potion (Qty : 20) - Increases DEF Power
4. Cancel

Potion yang diinginkan: 1
Setelah meminum ramuan ini, aura kekuatan terlihat mengelilingi Sanyaz dan gerakannya menjadi lebih cepat dan mematikan.

===== TURN 2 (Sanyaz) =====
Damage berkurang -1098.8% karena DEF Power: -10969.02
Sisa HP Sanyaz adalah 23136.0
Klik apapun untuk lanjut: █
```

Gambar 7.21 Ketika menggunakan potiom

```

===== TURN 3 (Sanyaz) =====
1. Attack
2. Use Potion
3. Quit

Pilih perintah: 3

Anda berhasil kabur dari BATTLE!

Masukkan command "Help" untuk daftar command yang dapat kamu panggil.
Masukkan command "Exit" untuk keluar dari program.

>>> Masukkan command : █

```

Gambar 7.22 Quit saat battle

```

Selamat datang di Arena!!
===== MONSTER LIST =====
1. Sanyaz
2. Charizard
3. Onix
Pilih monster untuk bertarung: 1

RAWRRR, a mengeluarkan monster Sanyaz !!!

Name      : Sanyaz
ATK Power : 1198.8
DEF Power : 1198.8
HP        : 1198.8
Level     : 3

Klik apapun untuk lanjut: d
===== STAGE 1 =====

RAWRRR, Monster Gengar telah muncul !!!

Name      : Gengar
ATK Power : 33.0
DEF Power : 33.0
HP        : 110.0
Level     : 1

===== TURN 1 (Sanyaz) =====
1. Attack
2. Use Potion
3. Quit

Pilih perintah: █

```

7.23 Tampilan Arena

```
Trasshaimase! Selamat datang di SHOP!!
Mau apa (lihat/beli/keluar) :lihat
Lihat apa (monster/item): monster
ID | Type | ATK Power | DEF Power | HP | Stok | Harga
1 | Sanyaz | 999 | 999 | 999 | 10 | 500
2 | Pikachu | 20 | 20 | 110 | 5 | 700
3 | Charizard | 28 | 15 | 110 | 3 | 1000
Mau apa (lihat/beli/keluar) :█
```

Gambar 7.24 tampilan shop

```
Trasshaimase! Selamat datang di SHOP!!
Mau apa (lihat/beli/keluar) :lihat
Lihat apa (monster/item): monster
ID | Type | ATK Power | DEF Power | HP | Stok | Harga
1 | Sanyaz | 999 | 999 | 999 | 10 | 500
2 | Pikachu | 20 | 20 | 110 | 5 | 700
3 | Charizard | 28 | 15 | 110 | 3 | 1000
Mau apa (lihat/beli/keluar) :beli
Jumlah OWCA coinmu sekarang adalah: 1000
Mau beli apa? (monster/item): 1
Inputan salah!
Mau beli apa? (monster/item): monster
Masukkan id monster: 1
Monster Sanyaz sudah ada dalam inventory-mu! Pembelian dibatalkan.
Mau apa (lihat/beli/keluar) :█
```

Gambar 7.25 Tampilan pembelian shop gagal

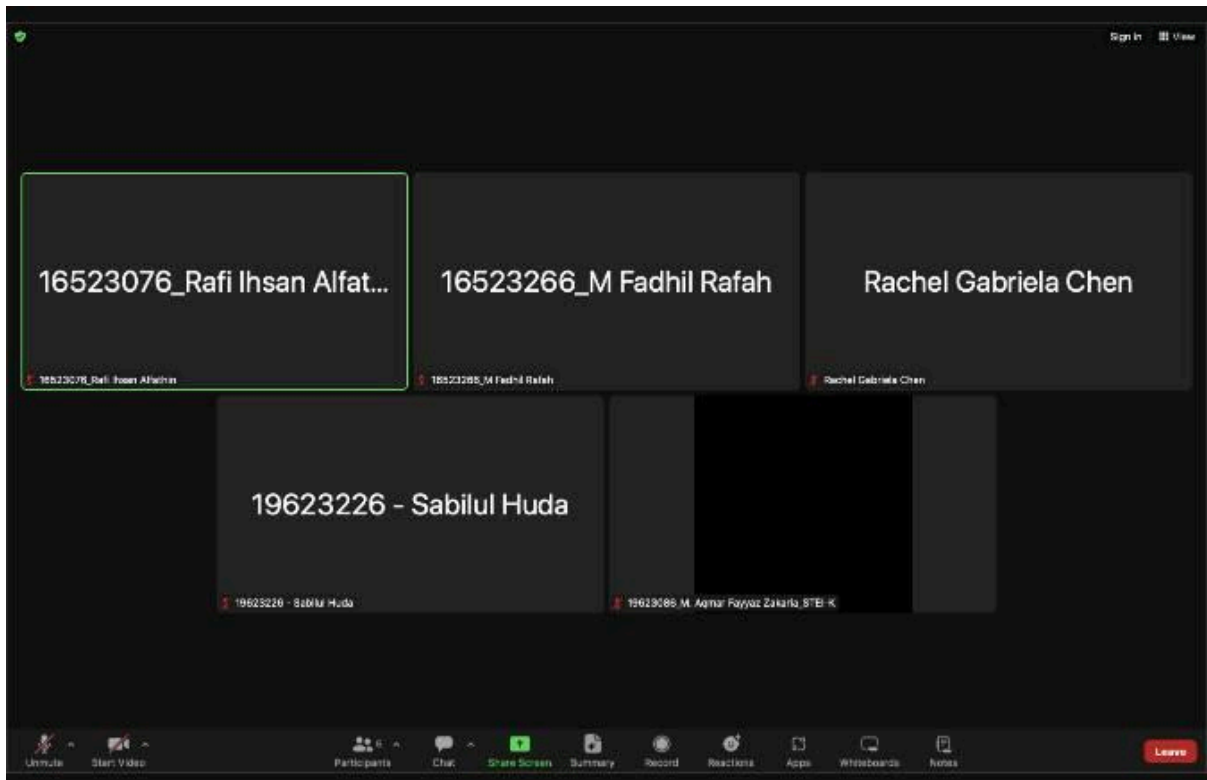
```
Mau apa (lihat/beli/keluar) :lihat
Lihat apa (monster/item): monster
ID | Type | ATK Power | DEF Power | HP | Stok | Harga
1 | Sanyaz | 999 | 999 | 999 | 10 | 500
2 | Pikachu | 20 | 20 | 110 | 5 | 700
3 | Charizard | 28 | 15 | 110 | 3 | 1000
Mau apa (lihat/beli/keluar) :keluar
Mr. Yanto bilang makasih, belanja lagi ya nanti :)

Masukkan command "Help" untuk daftar command yang dapat kamu panggil.
Masukkan command "Exit" untuk keluar dari program.

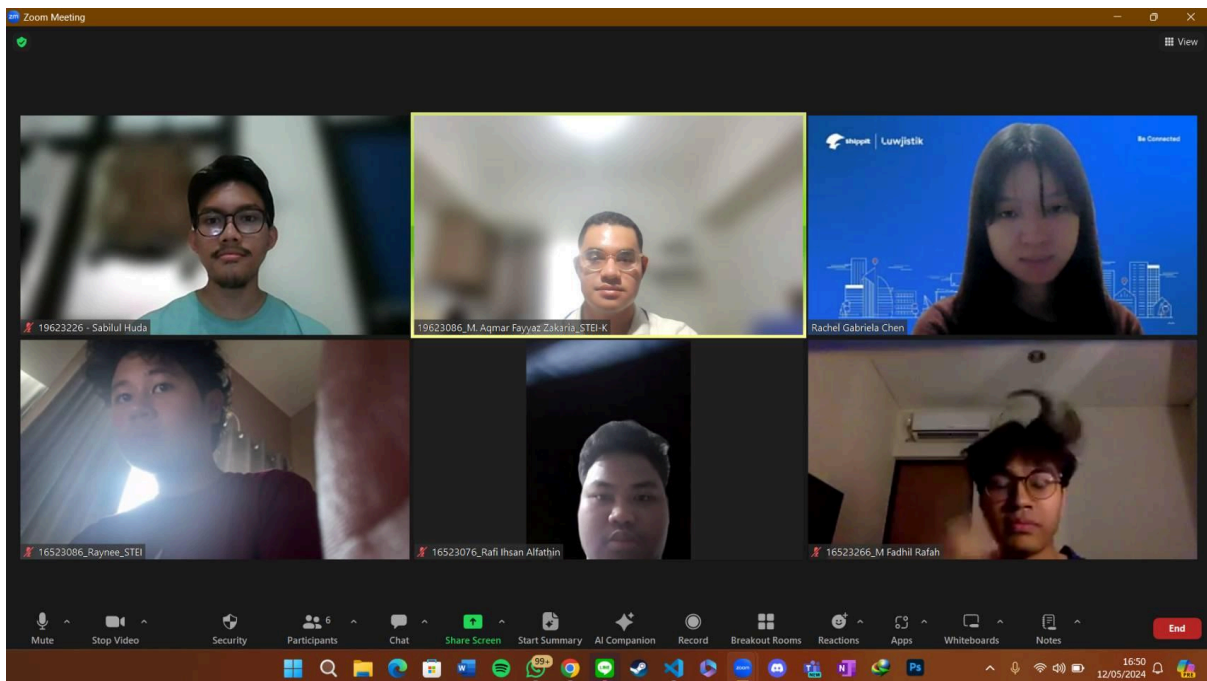
>>> Masukkan command : █
```

Gambar 7.26 Keluar dari shop

Catatan dan Lampiran



Gambar 8.1 Dokumentasi Asistensi 1



Gambar 8.2 Dokumentasi Asistensi 2



Gambar 8.3 Dokumentasi Kerja Kelompok

Catatan dan MoM asistensi 1 dan 2:

https://drive.google.com/drive/folders/1UA0b75_KuszjsBHtlOy1pVnmss5EsEN?usp=sharing

