

**TUGAS BESAR  
IF1210 DASAR PEMROGRAMAN  
KELAS 07**



**Oleh  
Kelompok A**

<b>Adrian Sami Pratama</b>	<b>16523037</b>
<b>Felicia Kannitha Ruth</b>	<b>16523167</b>
<b>Naomi Risaka Sitorus</b>	<b>19623087</b>
<b>Bertha Soliany Frandi</b>	<b>19623107</b>
<b>Michael Alexander Angkawijaya</b>	<b>19623157</b>

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
BANDUNG  
2024**

## **Halaman 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.”

Anggota kelompok A:

- Adrian Sami Pratama (NIM: 16523037)
- Felicia Kannitha Ruth (NIM: 16523167)
- Naomi Risaka Sitorus (NIM: 19623087)
- Bertha Soliany Frandi (NIM: 19623107)
- Michael Alexander Angkawijaya (NIM: 19623157)

## **Daftar Isi**

Halaman Pernyataan Kelompok .....	ii
Daftar Tabel.....	vii
Daftar Gambar .....	viii
I. Deskripsi Persoalan.....	1
II. Daftar Pembagian Kerja Anggota Kelompok .....	4
III. Checklist Hasil Rancangan, Implementasi, dan Testing .....	6
IV. Desain Command.....	7
a. Program Utama .....	7
b. F00 - Random Number Generator .....	7
c. F01 – Register .....	7
d. F02 – Login.....	8
e. F03 – Logout.....	9
f. F04 - Menu & Help.....	9
g. F05 – Monster.....	11
h. F06 – Potion.....	11
i. F07 – Inventory.....	12
j. F08 – Battle.....	13
k. F09 – Arena.....	17
l. F10 - Shop & Currency.....	19
m.F11 – Laboratory.....	20
n. F12 - Shop Management.....	22
o. F13 - Monster Management.....	23
p. F14 – Load .....	24
q. F15 – Save.....	25
r. F16 – Exit.....	26
V. Desain Kamus Data.....	27
a. Program Utama .....	27
b. F00 - Random Number Generator .....	27
c. F01 – Register .....	28
d. F02 – Login.....	28
e. F03 – Logout .....	28
f. F04 - Menu & Help.....	29
g. F05 – Monster .....	29

h.	F06 – Potion.....	30
i.	F07 – Inventory.....	30
j.	F08 – Battle.....	30
k.	F09 – Arena.....	32
l.	F10 - Shop & Currency.....	33
m.	F11 – Laboratory.....	33
n.	F12 - Shop Management.....	34
o.	F13 - Monster Management.....	35
p.	F14 – Load .....	35
q.	F15 – Save.....	35
r.	F16 – Exit.....	35
VI.	Desain Dekomposisi Algoritmik dan Fungsional Program .....	37
a.	Program Utama .....	37
b.	F00 - Random Number Generator .....	37
c.	F01 - Register.....	37
d.	F02 – Login.....	37
e.	F03 – Logout.....	37
f.	F04 - Menu & Help.....	37
g.	F05 – Monster .....	37
h.	F06 – Potion.....	38
i.	F07 – Inventory.....	38
j.	F08 – Battle.....	38
k.	F09 - Arena .....	39
l.	F10 - Shop & Currency.....	39
m.	F11 - Laboratory .....	40
n.	F12 - Shop Management.....	40
o.	F13 - Monster Management.....	41
p.	F14 - Load.....	41
q.	F15 - Save .....	41
r.	F16 - Exit .....	41
VII.	Spesifikasi Program .....	42
a.	Program Utama .....	42
b.	F00 - Random Number Generator .....	45
c.	F01 - Register.....	46

d.	F02 – Login.....	50
e.	F03 – Logout.....	51
f.	F04 - Menu & Help.....	52
g.	F05 – Monster.....	53
h.	F06 – Potion.....	62
i.	F07 – Inventory.....	66
j.	F08 – Battle.....	68
k.	F09 – Arena.....	77
l.	F10 - Shop & Currency.....	79
m.	F11 – Laboratory.....	86
n.	F12 - Shop Management.....	88
o.	F13 - Monster Management.....	100
p.	F14 - Load.....	103
q.	F15 - Save .....	104
r.	F16 - Exit .....	106
VIII.	Hasil Pengujian Program .....	107
a.	Program Utama .....	90
b.	F00 - Random Number Generator .....	107
c.	F01 – Register .....	107
d.	F02 - Login .....	108
e.	F03 - Logout .....	110
f.	F04 - Menu & Help.....	111
g.	F05 - Monster.....	112
h.	F06 - Potion.....	115
i.	F07 - Inventory .....	117
j.	F08 - Battle .....	120
k.	F09 - Arena .....	126
l.	F10 - Shop & Currency.....	129
m.	F11 - Laboratory .....	133
n.	F12 - Shop Management.....	135
o.	F13 - Monster Management.....	137
p.	F14 - Load.....	140
q.	F15 - Save .....	140
r.	F16 - Exit .....	140

IX.	Lampiran .....	142
-----	----------------	-----

## **Daftar Tabel**

Tabel 2. 1 Pembagian kerja anggota kelompok .....	5
Tabel 3. 1 Checklist Hasil Rancangan, Implementasi, dan Testing .....	6

## Daftar Gambar

Gambar 8. 1 Register berhasil.....	107
Gambar 8. 2 Username tidak valid .....	107
Gambar 8. 3 Username sudah terdaftar.....	108
Gambar 8. 4 Monster awal yang dipilih tidak ditemukan.....	108
Gambar 8. 5 Pemanggilan help berhasil .....	111
Gambar 8. 6 Pemanggilan help berhasil .....	111
Gambar 8. 7 Pemanggilan command dalam menu berhasil.....	111
Gambar 8. 8 Pemanggilan command dalam menu berhasil 2.....	112
Gambar 8. 9 Pemanggilan command dalam menu berhasil 3 .....	112
Gambar 8. 10 Pemanggilan command dalam menu tidak valid .....	112
Gambar 8. 11 tampilan monster pada battle_1 .....	113
Gambar 8. 12 tampilan monster pada battle_2 .....	114
Gambar 8. 13 tampilan monster pada battle_3 .....	115
Gambar 8.14 Pemanggilan potion di battle.....	116
Gambar 8.15 Contoh input valid_1.....	116
Gambar 8.17 Contoh input tidak valid_1 .....	116
Gambar 8.18 Contoh input tidak valid_2 .....	117
Gambar 8.19 Input tidak valid kasus kosong.....	117
Gambar 8.20 Pemanggilan inventory dalam menu HELP .....	117
Gambar 8.21 Penggunaan input valid .....	117
Gambar 8.22 Penggunaan input yang valid pada inventory .....	118
Gambar 8.23 Penggunaan input yang valid pada inventory .....	118
Gambar 8.24 Penggunaan input valid pada inventory .....	118
Gambar 8.25 Perintah keluar dari lab .....	119
Gambar 8.26 Input tidak valid dengan angka lain .....	119
Gambar 8.27 Input tidak valid dengan angka lain .....	119
Gambar 8.28. Input tidak valid dengan input kosong.....	120
Gambar 8.29 Input tidak valid dengan input string bebas .....	120
Gambar 8.42 Pemanggilan laboratory pada HELP .....	133
Gambar 8.43 Contoh input valid.....	134
Gambar 8.44 Contoh input valid.....	134
Gambar 8.45 input tidak valid dengan angka lain .....	134
Gambar 8.46 Input tidak valid dengan string bebas .....	135
Gambar 8.47 Input tidak valid dengan input kosong.....	135
Gambar 8. 35.....	140

## I. Deskripsi Persoalan

Pada tugas besar ini, program yang dibuat adalah program untuk membantu Agent Purry mengalahkan monster monster yang dibuat oleh Dr. Asep Spakbor. Cara mengalahkan monsternya adalah dengan mencari dan melatih monster sendiri untuk dipakai melawan monster Dr. Asep Spakbor dengan cara turn-based combat. Program ini terdiri dari beberapa bagian spesifikasi.

Bagian yang pertama adalah random number generator. Random number generator (RNG) adalah suatu fungsi untuk menghasilkan satu angka acak dari range tertentu. RNG ini menggunakan algoritma Linear Congruential Generator (LCG).

Bagian yang kedua adalah register. Sebelum user bisa menggunakan fitur yang ada pada program, user harus membuat akun terlebih dahulu. Pada saat register, user akan membuat username dan password yang nantinya akan disimpan di file “user.csv” pada saat men-save program. Pada saat register, user akan otomatis mendapatkan role agent dan tidak memiliki O.W.C.A coin. Pada saat ini juga, user bisa memilih monster starter untuk dipakai battle nantinya.

Setelah user register dan mendapatkan akun, user bisa login untuk mulai menggunakan fitur pada program. Pada bagian ketiga, login, user bisa masuk ke akun yang dimilikinya. Selain role agent, ada juga 1 role admin untuk mengatur fitur pada game tersebut.

Jika ada login, pastinya ada juga logout agar user bisa keluar dari program. Jika user sudah login dengan suatu akun, user bisa logout. User akan kehilangan akses terhadap akunnya setelah logout.

Pada saat setelah login, user akan ditampilkan menu fitur apa saja yang bisa digunakan. Pada bagian ini, user bisa memilih menu help untuk program menampilkan pilihan pilihan yang bisa dilakukan untuk menjalankan program.

Pada program ini, ada juga bagian monster, yaitu bagian program yang mengurus atribut yang dimiliki monster. Setiap monster memiliki 5 atribut, yaitu id, type atau nama, ATK power, DEF power, dan HP. Data semua monster akan disimpan di dalam file “monster.csv”. Id adalah kode unik yang dimiliki setiap monster, sedangkan type adalah nama monster tersebut. ATK power adalah kekuatan serangan monster tersebut. Pada saat battle, monster akan menyerang dengan rentang damage sebesar  $\pm 30\%$  dari ATK power monster tersebut. DEF power adalah pertahanan yang dimiliki suatu monster. DEF power bernilai di rentang 0 sampai 50. DEF power akan mengurangi damage yang diterima sebesar DEF power%. Terakhir, HP adalah nyawa yang dimiliki monster. Selain atribut tersebut, setiap monster juga memiliki level tertentu. Data level ini akan disimpan di file yang berbeda, yaitu “monster\_inventory.csv”. Nilai level adalah rentang 1 sampai 5. Jika monster level 1, maka monster akan menggunakan base atribut, sedangkan jika level lebih dari 1, maka pada saat battle, atribut monster akan menjadi base atribut ditambah dengan base atribut dikali dengan (level x 10%).

Selain monster, program ini juga memiliki potion. Potion ini bisa digunakan sekali untuk 1 macam potionnya pada saat battle berlangsung dan hanya bisa digunakan jika user

memiliki potion tersebut. Data potion yang dimiliki user akan disimpan di file “item\_inventory.csv”. Pada program ini, ada 3 macam potion, yaitu strength potion, resilience potion, dan healing potion. Strength potion memberikan efek meningkatkan ATK power sebesar 5% dari ATK power awal. Resilience potion meningkatkan DEF power sebesar 5% dari DEF power awal. Kemudian healing potion mengisi darah sebanyak 25% dari base HP monster. ATK, DEF, dan HP yang digunakan adalah sesuai monster level tersebut.

Pada program ini, setiap akun memiliki inventory tersendiri. Inventory ini diisi dengan monster dan juga potion yang dimiliki oleh user. Setiap item yang ada di inventory juga dapat dimunculkan detailnya. Inventory juga dapat menunjukkan jumlah O.W.C.A coin yang dimiliki oleh user. Inventory ini akan disimpan di dalam file “item\_inventory.csv” dan “monster\_inventory.csv”.

Program ini juga memiliki subprogram untuk melakukan battle. Mekanisme battle dalam program ini adalah battle yang bersifat turn based 1v1 game. Setiap battle akan memiliki beberapa putaran di mana monster user dan musuh akan saling bergantian menyerang. Pada saat pertama kali mulai battle, RNG akan men-generate nomor random yang akan menghasilkan id monster yang muncul. Setelah monster musuh muncul, user akan memilih monster mana yang ingin dipakai untuk bertempur. User akan memiliki 3 pilihan pada setiap turn, yaitu attack, use potion, dan quit. Attack berarti monster user akan menyerang monster musuh dengan ketentuan ATK dan DEF yang akan mengubah besar nilai damage yang diterima monster musuh. Setelah itu, monster musuh akan menyerang user kembali. Jika user memilih use potion, user akan diberikan pilihan potion apa saja yang dimilikinya. User bisa memilih menggunakan potion yang dimiliki atau cancel use potion dan kembali ke pilihan awal. Jika user memilih menggunakan potion, user akan mendapatkan efek dari potion tersebut kemudian monster musuh akan langsung menyerang tanpa user menyerang terlebih dahulu. Kemudian terakhir, jika user memilih quit, user akan kabur dan keluar dari battle tersebut. Battle akan terus berlangsung sampai salah satu HP dari monster user atau monster musuh habis. Jika user menang, user akan diberi hadiah berupa O.W.C.A coin yang jumlahnya di-generate oleh RNG dengan range 5-30. Setelah selesai battle, menang atau kalah, HP monster user akan di-reset setelah selesai battle.

Selain battle, program ini juga memiliki fitur arena, yaitu fitur di mana user akan melakukan battle sebanyak 5 stage atau 5 kali. Monster musuh yang dilawannya akan memiliki level sesuai stage nya. Mekanisme pertarungan pada setiap stage sama dengan battle seperti biasa dan HP monster user akan di-reset setiap selesai 1 stage. Setiap stage akan memberikan hadiah O.W.C.A coin tertentu. Arena akan selesai jika user memilih quit, atau user kalah dalam stage tertentu, atau user menyelesaikan semua 5 stage tersebut. Hadiah akan diberikan sesuai stage yang diselesaikan.

Pada program ini juga terdapat shop & currency, di mana user bisa membeli monster atau item yang dijual di shop ini. Item item yang ada di shop disimpan di dalam database. User hanya bisa melihat item yang dijual di shop. Khusus untuk monster, jika user sudah memiliki suatu monster, maka user tidak akan bisa membeli monster tersebut.

Monster yang dimiliki user juga bisa di-upgrade ke level yang lebih tinggi. Tempat untuk meng-upgrade nya adalah laboratory. Pada saat memilih command laboratory, user akan

ditampilkan monster yang dimilikinya di inventory. Setelah itu, user bisa memilih monster mana yang ingin di-upgrade. Jika monster tersebut di bawah level 5 (masih bisa diupgrade), maka akan ditampilkan harga untuk meng-upgrade 1 level lebih tinggi. Selama user memiliki O.W.C.A coin yang cukup, user akan terus bisa meng-upgrade monster nya, maksimal sampai level 5. Level monster ini akan disimpan di dalam file “monster\_inventory.csv”.

Terdapat dua fitur khusus yang hanya bisa diakses oleh akun admin, yang pertama adalah shop management. Database untuk menyimpan item yang dijual di shop akan berbeda dengan database untuk semua item yang ada. Admin akan bisa mengatur apa yang dijual di shop, sehingga bisa dilakukan penambahan, pengurangan, atau perubahan item yang dijual di shop. Admin juga bisa mengatur harga item item nya. Database tersebut akan disimpan di dalam file “item\_shop.csv” dan “monster\_shop.csv”.

Fitur khusus admin yang kedua adalah monster management. Admin bisa melakukan pengaturan monster yang ada pada database “monster.csv”. Admin juga bisa menambahkan monster yang diinginkan dengan fitur ini.

Untuk membaca save-an akun, program ini memiliki prosedur load. Load akan membaca folder yang berisi file save untuk dijalankan di program. Load hanya akan digunakan sekali pada saat awal memulai program. Load juga memiliki beberapa asumsi, yaitu file penyimpanan dalam folder pasti ada, memiliki nama yang fixed, dan memiliki format yang sesuai dengan struktur data eksternal. Namun, prosedur akan tetap melakukan validasi apakah folder tersebut ada atau tidak.

Tidak hanya load, program ini juga memiliki prosedur save. Prosedur ini akan menuliskan semua perubahan pada file csv masing masing. Mekanismenya, pertama, program akan meminta nama folder yang akan dijadikan tempat penyimpanan. Jika folder belum ada, maka prosedur akan membuat folder baru dan menyimpan di folder tersebut. Jika folder sudah ada namun di dalamnya belum ada file, maka prosedur akan membuat file baru. Kemudian jika folder sudah ada dan sudah terisi, maka prosedur akan me-replace file yang ada di dalam tersebut. Program hanya akan menulis ke csv pada saat prosedur ini dipanggil.

Terakhir, program ini juga memiliki prosedur exit. Seperti namanya, prosedur ini digunakan untuk mengakhiri program atau keluar dari program. Sebelum keluar, user akan diberikan pilihan apakah ingin men-save terlebih dahulu. Jika sudah memilih, maka program pun akan selesai.

## II. Daftar Pembagian Kerja Anggota Kelompok

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F00 - Random Number Generator	function lcg, generate_number	16523037	16523037	19623157
F01- Register	procedure register, procedure cekValid, dan function cekPassword	19623107	19623107	19623087
F02- Login	function login	19623087	19623087	19623107
F03- Logout		16523167	16523167	19623107
F04- Menu & Help	procedure help	19623157	19623157	19623087
F05- Monster	function attack_user, attack_bot, defense, level_buff, procedure gambar_monster	16523037	16523037	16523167
F06- Potion	function pilih_potion, use_potion	19623157	19623157	16523037
F07- Inventory	procedure inventory	19623157	19623157	16523037
F08- Battle	function battle	16523037	16523037	16523167
F09- Arena	function arena	16523037	16523037, 19623157	
F10- Shop & Currency	procedure display, function id_shop, read_filemonst	16523167	16523167, 19623157	19623087

	er, read_filepotio n, has_monsterin ven, buy_monster, buy_potion, shopmain			
F11- Laboratory	function laboratory dan function is_integer	19623087	19623087	16523037
F12- Shop Management	procedure display, function id_shop, read_filemonst er, read_filepotio n, add, update, delete, shop_manage ment	16523167	16523167, 19623157	19623157
F13- Monster Management	function monster_ma nagement dan function is_integer	19623087	19623087	19623107
F14- Load	Argparse dan function csv_to_array	19623107	19623107	19623157
F15- Save	procedure save dan procedure write_csv	19623107	19623107	16523167
F16- Exit	function exit	19623107	19623107	16523167

Tabel 2. 1 Pembagian kerja anggota kelompok

### III. Checklist Hasil Rancangan, Implementasi, dan Testing

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	V	V
F13- Monster Management	V	V	V
F14- Load	V	V	V
F15- Save	V	V	V
F16- Exit	V	V	V

Tabel 3. 1 Checklist Hasil Rancangan, Implementasi, dan Testing

## IV. Desain Command

### a. Program Utama

Nama program: program\_OWCA

Masukan:

> **“Masukkan perintah!\nKetik "HELP" untuk melihat menu”**

Keluaran:

Bila user memasukkan perintah yang tidak valid, atau perintah yang tidak dapat dilakukan user sesuai dengan role dan/atau status nya, dicetak:

> **“Perintah tidak valid! Pastikan input sudah benar.”**

Bila user memasukkan perintah yang valid, perintah tersebut akan dijalankan sesuai dengan desain dan spesifikasinya masing-masing.

Bila user memasukkan perintah “EXIT”, dicetak:

> **“Sampai jumpa kembali Agent!”** (bila role user adalah Agent)

> **“Sampai jumpa kembali admin!”** (bila role user adalah admin)

### b. F00 - Random Number Generator

Nama command: generate\_number(x\_prev, num\_range=None)

Masukan:

> Menerima input argumen x\_prev (dari fungsi lcg) dan num\_range

Keluaran:

> Mengeluarkan angka acak dengan range yang dimasukkan pada input

### c. F01 – Register

Nama command: register

Procedure: register(logged\_in, array\_user, array\_monster, array\_monster\_inventory)

Masukan:

> Masukkan username:

> Masukkan password:

> Monster pilihanmu:

Keluaran:

Melakukan tiga kali validasi yang berbeda terlebih dahulu sebelum username yang diberikan dapat terdaftar sebagai user baru. Pengecekan yang pertama adalah mengenai status pengguna sedang login menggunakan username lain atau tidak. Jika ya, pengguna tidak bisa melakukan register dan akan mencetak

"Anda masih login dengan username {username}. Silahkan logout terlebih dahulu sebelum melakukan register." Jika tidak, akan lanjut ke pengecekan kedua.

Validasi kedua adalah mengenai karakter yang digunakan oleh pengguna untuk username. Karakter yang boleh digunakan hanya alfabet, angka, underscore, dan stip. Jika didapati karakter di luar yang diperbolehkan, program akan mencetak "Username hanya boleh berisi alfabet, angka, underscore, dan strip." Setelah itu akan meminta ulang masukkan username. Jika username yang diberikan sudah memenuhi syarat pada validasi ini, username akan menjalani validasi terakhir.

Validasi selanjutnya adalah mengenai apakah username yang diberikan oleh pengguna sudah digunakan oleh pengguna lain atau tidak (sesuai dengan file csv yang di load). Jika username sudah pernah digunakan, pengguna tidak bisa menggunakan username itu dan akan diminta untuk melakukan input ulang untuk username. Jika username yang diberikan merupakan username yang baru maka akan dilakukan pengecekan pada password yang diberikan oleh pengguna dimana password tidak diperbolehkan kosong atau berupa hanya spasi.

Setelah username dan password telah divalidasi, pengguna akan diminta untuk memilih monster untuk dijadikan monster awal. Masukkan untuk pemilihan monster juga divalidasi untuk memastikan pengguna memilih monster yang ada pada database. Barulah setelah tahap ini, username dan password yang diberikan oleh pengguna terdaftar menjadi user baru pada program.

#### d. F02 – Login

Nama command: login

Function: login(logged\_in, user\_id, array\_user)

Masukan:

- > Masukkan username
- > Masukkan password

Keluaran:

Jika username ditemukan dalam array\_user dan password sesuai, dicetak:

- > **'Login berhasil!'**
- > **'Selamat datang', array\_user[i][3], username + '!'**
- > **'Ketik "HELP" untuk melihat list command yang dapat kamu gunakan.'**

Jika username ditemukan dalam array\_user tetapi password tidak sesuai, dicetak:

- > **'Password kamu salah!'**

Jika username tidak ditemukan dalam array\_user, dicetak:

- > **'Username kamu belum terdaftar!'**
- > **'Lakukan registrasi terlebih dahulu sebelum melakukan login.'**

#### e. F03 – Logout

Nama command: logout

Function: logout (check login, logout)

Masukan:

- > User akan dipastikan terlebih dahulu bahwa ia sudah melakukan login dan memberikan masukkan apabila user inginmelakukan logout atau tidak.

Keluaran:

- > Program akan mengecek bila user sudah melakukan login apa belum

Bila user belum melakukan login, keluaran:

- > Program tidak akan bisa melakukan logout

Bila user sudah melakukan login, keluaran:

- > Program akan meminta input dari user bila ingin dilakukan logout atau tidak

Bila user memilih iya:

- > Program akan melakukan aksi untuk mengeluarkan user dari permainan

Bila user memilih tidak:

- > Program akan tetap berada di dalam permainan

#### f. F04 - Menu & Help

Nama command: help(login,role,username)

Masukan :

- > Menerima argumen login (status login user), role (role user) dan username (username user)

Keluaran :

- > Command / hal yang dapat dilakukan oleh user,
- > "...."

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

- 1. Logout: Keluar dari akun yang sedang digunakan**
  - 2. Shop Management: Melakukan manajemen pada SHOP sebagai tempat jual beli peralatan Agent**
  - 3. Monster Management: Melakukan manajemen pada database monster**
  - 4. Save: Menyimpan data ke dalam folder**
  - 5. Exit: Keluar dari program**
- """"

Bila pengguna sudah login, dan role pengguna adalah Agent, dicetak:

> """"

Halo Agent {username}! 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: Melihat detail monster, potion, dan OC yang dimiliki oleh Agent**
  - 3. Battle: Bertarung 1v1 melawan monster**
  - 4. Arena: Melatih monster yang dimiliki oleh Agent**
  - 5. Shop: Membeli item berupa monster atau potion**
  - 6. Laboratory: Meningkatkan level monster yang dimiliki oleh Agent**
  - 7. Save: Menyimpan data ke dalam folder**
  - 8. Exit: Keluar dari program**
- """"

Bila pengguna belum login, dicetak:

> """"

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

- 1. Login: Masuk ke dalam akun yang sudah terdaftar**
  - 2. Register: Membuat akun baru**
  - 3. Save: Menyimpan data ke dalam folder**
  - 4. Exit: Keluar dari program**
- """"

**Sebagai panduan bagi user, dicetak:**

> """"

**Footnote:**

**1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar**

**2. Jangan lupa untuk memasukkan input yang valid**

""""

#### **g. F05 – Monster**

Nama command: attack(attackpower)

Masukan :

> Menerima argumen attackpower yang merupakan base attack power monster

Keluaran:

> Menghasilkan attack power dengan rentang  $\pm 30\%$  dari base attack power

#### **h. F06 – Potion**

Nama command: use\_potion(user\_id, item\_inventory, status\_potion, nama\_monster)

Masukan :

> Menerima argumen user\_id, item\_inventory (data item\_inventory.csv), status\_potion (status jenis potion yang sudah terpakai dan belum terpakai), dan nama\_monster

Keluaran :

> Program akan meminta user untuk input pilihan potion yang ingin dipakai.

Bila user tidak memiliki potion, dicetak:

**> "Wah, kamu sedang tidak memiliki ramuan ini, silahkan pilih ramuan lain!"**

Bila user sudah menggunakan jenis potion tersebut, dicetak:

**> "Kamu mencoba memberikan ramuan ini kepada {nama\_monster}, namun dia menolaknya seolah-olah dia memahami ramuan tersebut sudah tidak bermanfaat lagi."**

Bila user menggunakan potion ATK, dicetak:

> "Setelah meminum ramuan ini, aura kekuatan terlihat mengelilingi {nama\_monster} dan gerakannya menjadi lebih cepat dan mematikan."

Bila user menggunakan potion DEF, dicetak:

> "Setelah meminum ramuan ini, muncul sebuah energi pelindung di sekitar {nama\_monster} yang membuatnya terlihat semakin tangguh dan sulit dilukai."

Bila user menggunakan potion HEAL, dicetak:

> "Setelah meminum ramuan ini, luka-luka yang ada di dalam tubuh {nama\_monster} sembuh dengan cepat. Dalam sekejap, {nama\_monster} terlihat Kembali prima dan siap melanjutkan pertempuran."

### i. F07 – Inventory

Nama command: inventory(user: list, item\_inventory: list, monster\_inventory: list, monster: list)

Masukan:

> Menerima argumen user (list data user), item\_inventory (data item\_inventory.csv), monster\_inventory (data monster\_inventory.csv) , monster (data monster.csv)

Keluaran:

> "Selamat datang di Inventory!\nAnda dapat melihat detail monster dan item yang dimiliki pada Inventory\n"

> ===== INVENTORY LIST (User ID: {user[0]})  
=====

> "Jumlah O.W.C.A Coin-mu sekarang {user[4]}."

Untuk setiap monster yang dimiliki user, ditampilkan dengan format:

> "{count}. Monster \t(Name: {monster[int(monster\_inventory[i][1])][1]},  
Lvl. {monster\_inventory[i][2]}, HP:  
{monster[int(monster\_inventory[i][1])][4]})"

Untuk setiap potion yang dimiliki user, ditampilkan dengan format:

> "{count}. Potion \t(Type: {tipe}, Qty: {item\_inventory[i][2]})"

Setelah itu, user dapat melihat detail dari item.

Untuk detail dari monster, ditampilkan dengan format:

> """"

**Monster**

```

Name    : {user_inventory[idx-1][1]}

ATK Power : {int(user_inventory[idx-1][2]) + int((int(user_inventory[idx-1][2])*(((int(user_inventory[idx-1][5])-1)*10)/100)))}

DEF Power : {int(user_inventory[idx-1][3]) + int((int(user_inventory[idx-1][3])*(((int(user_inventory[idx-1][5])-1)*10)/100)))}

HP      : {int(user_inventory[idx-1][4]) + int((int(user_inventory[idx-1][4])*(((int(user_inventory[idx-1][5])-1)*10)/100)))}

Level   : {user_inventory[idx-1][5]}

"""

```

Untuk detail dari potion, ditampilkan dengan format:

> """"

### **Potion**

```

Type    : {user_inventory[idx-1][1]}

Quantity : {user_inventory[idx-1][2]}

"""

```

## j. F08 – Battle

Nama command: battle(monsterarray, monster\_inventory, item\_inventory, user\_id)

Masukan:

```

> Menerima list monsterarray, yaitu list data semua monster

> Menerima list monster_inventory, yaitu list yang diambil dari
monster_inventory.csv

> Menerima list item_inventory, yaitu list yang diambil dari item_inventory.csv

> Menerima user_id

```

Keluaran:

> Program akan melakukan mekanisme battle. Pada awal battle, program akan menggunakan RNG untuk menghasilkan id monster yang muncul. Setelah itu, program akan menampilkan data monster yang muncul dengan tampilan berikut.

"""

**RAWRRR, Monster {monsterarray[monster\_muncul\_id][1]} telah muncul !!!**

**gambarmonster(monsterarray[monster\_muncul\_id][0])**

**Name** : {monsterarray[monster\_muncul\_id][1]}

**ATK Power : {monsterarray[monster\_muncul\_id][2]}**

**DEF Power : {monsterarray[monster\_muncul\_id][3]}**

**HP : {monsterarray[monster\_muncul\_id][4]}**

**Level : 1**

"""

Dengan gambar monster adalah gambar monster yang muncul (setiap monster memiliki gambar yang unik). Setelah data monster musuh ditampilkan, program akan meminta input monster yang mana yang ingin dipilih oleh user untuk bertarung, dengan tampilan sebagai berikut.

"===== MONSTER LIST ====="

"{count}. {monsterarray[monster\_id][1]}"

"{count}. {monsterarray[monster\_id][1]}"

**"Pilih monster untuk bertarung: "**

Dengan count adalah urutan angka yang bisa dipilih dan monsterarray[monster\_id][1] adalah monster yang dimiliki oleh user. Jika user memilih monster yang ada pada list tersebut, program akan lanjut ke bagian selanjutnya, sedangkan jika input tidak valid, maka program akan mengeluarkan pesan error.

**"Pilihan nomor tidak tersedia!"**

Setelah user memilih monster, maka akan ditampilkan data monster user dan pilihan aksi yang bisa dilakukan

**gambar\_monster(user\_battle\_monster[0])**

"""

**RAWR, Agent X mengeluarkan monster {user\_battle\_monster[1]} !!!**

**Name : {user\_battle\_monster[1]}**

**ATK Power : {user\_battle\_monster[2]}**

**DEF Power : {user\_battle\_monster[3]}**

**HP : {user\_battle\_monster[4]}**

**Level : {level}**

"""

**===== TURN {turn\_counter} ({monster\_name}) =====**

**1. Attack**

## 2. Use Potion

### 3. Quit

"""

**"Pilih perintah: "**

Dengan turn counter adalah jumlah turn yang sudah dilakukan dan monster\_name adalah nama monster yang sedang memiliki turn.

Jika user memilih 1, yaitu attack, maka akan muncul output hasil serangan yang dilakukan oleh user, kemudian musuh akan menyerang user dan mengeluarkan hasilnya juga.

"""

**SCHWINKKK, {user\_monster\_name} menyerang {bot\_monster\_name} !!!**

Name : {enemy\_battle\_monster[1]}

ATK Power : {enemy\_battle\_monster[2]}

DEF Power : {enemy\_battle\_monster[3]}

HP : {base\_HP\_enemy\_monster}

Level : {enemy\_battle\_monster\_level}

**Penjelasan:** ATT: {user\_damage}  
 ({tanda}{math.floor((user\_battle\_monster[2]-user\_damage)/user\_battle\_monster[2]\*100)}%), Reduced by: {damage\_reduction} ({enemy\_battle\_monster[3]}%), ATT Results: {damage\_after\_reduction}

"""

Dengan user\_monster\_name adalah monster yang sedang menyerang dan bot\_monster\_name adalah monster yang sedang diserang, enemy\_battle\_monster adalah monster yang sedang diserang. User damage adalah damage yang dikeluarkan oleh penyerang (dengan menggunakan RNG, sebelum dikurangi dengan defense musuh), damage reduction merupakan pengurangan damage karena defense musuh, dan damage\_after\_reduction adalah damage yang diterima oleh musuh setelah dikurangi oleh defense musuh.

Jika user memilih 2, yaitu use potion, maka user akan bisa menggunakan potion, tetapi tidak akan menyerang di turn tersebut.

**"===== POTION LIST ====="**

**"{count}. {txt1} Potion (Qty: {item\_inventory[i][2]}) - {txt2}"**

**"Pilih potion untuk diminum: "**

Dengan txt1 adalah jenis potion dan txt2 adalah keterangan potion tersebut. Jika user memilih potion yang tidak dimilikinya, maka akan muncul pesan error.

**"Wah, kamu sedang tidak memiliki ramuan ini, silahkan pilih ramuan lain!"**

Jika user memilih angka yang tidak valid, maka akan muncul pesan error juga.

**"Pilihan nomor tidak tersedia!"**

Jika user memilih potion yang sudah digunakan, maka akan muncul pesan error juga.

**"Kamu mencoba memberikan ramuan ini kepada {nama\_monster}, namun dia menolaknya seolah-olah dia memahami ramuan tersebut sudah tidak bermanfaat lagi."**

Jika user memilih potion yang dimilikinya dan belum diminum, maka akan ada output tertentu untuk setiap jenis potion.

Untuk strength potion:

**"Setelah meminum ramuan ini, aura kekuatan terlihat mengelilingi {nama\_monster} dan gerakannya menjadi lebih cepat dan mematikan."**

Untuk resilience potion:

**"Setelah meminum ramuan ini, muncul sebuah energi pelindung di sekitar {nama\_monster} yang membuatnya terlihat semakin tangguh dan sulit dilukai."**

Untuk healing potion:

**"Setelah meminum ramuan ini, luka-luka yang ada di dalam tubuh {nama\_monster} sembuh dengan cepat. Dalam sekejap, {nama\_monster} terlihat kembali prima dan siap melanjutkan pertempuran."**

Jika user memilih 3, yaitu quit, maka user akan kabur dari battle dan dikeluarkan output ini.

**"Anda berhasil kabur dari BATTLE!"**

Setelah bagian, user, maka monster pun akan menyerang user dan memberikan output hasil serangannya. Proses ini akan berulang sampai user menang atau kalah.

Jika user kalah, maka akan muncul output seperti ini.

**"Yahhh, Anda dikalahkan monster {bot\_battle\_monster[1]}. Jangan menyerah, coba lagi !!!"**

Jika user menang, maka user akan mendapatkan O.W.C.A coin dengan output seperti ini.

**"Selamat, Anda berhasil mengalahkan monster {bot\_battle\_monster[1]} !!!"**

**"OC yang diperoleh: {OC}"**

Dengan OC adalah O.W.C.A coin.

#### k. F09 – Arena

Nama command: arena

Masukan:

> Sama dengan battle, hanya saja user memilih monster sebelum monster musuh muncul.

Keluaran:

> Command ini mirip dengan battle. Pada awal command, program akan meminta input monster apa yang ingin digunakan.

**"Selamat datang di Arena!!"**

**"===== MONSTER LIST ====="**

**"{count}. {monsterarray[monster\_id][1]}"**

**"{count}. {monsterarray[monster\_id][1]}"**

**"Pilih monster untuk bertarung: "**

Monster yang dipilih user akan digunakan sampai stage terakhir. Setiap berganti stage, HP monster user akan kembali full seperti semula. Setelah memilih monster, arena akan mulai dari stage 1, setiap stage akan memunculkan monster yang berbeda (monster yang sama bisa muncul lebih dari sekali). Level monster musuh menyesuaikan dengan nomor stage.

**"===== STAGE {bot\_battle\_monster\_level} ====="**

**gambar\_monster(monsterarray[monster\_muncul][0])**

"""

**RAWRRR, Monster {monsterarray[monster\_muncul][1]} telah muncul !!!**

**Name : {monsterarray[monster\_muncul][1]}**

**ATK Power : {monsterarray[monster\_muncul][2]}**

**DEF Power : {monsterarray[monster\_muncul][3]}**

**HP : {monsterarray[monster\_muncul][4]}**

**Level : {level}**

"""

Kemudian program akan battle seperti biasa. Arena akan selesai jika user kalah pada stage tertentu atau menyelesaikan seluruh stage. User akan mendapatkan hadiah sebanyak stage yang diselesaikan.

Kasus menyelesaikan stage selain stage 5

**"Selamat, Anda berhasil mengalahkan monster {bot\_battle\_monster[1]} !!!"**

**"STAGE CLEARED! Anda akan mendapatkan {OC} OC pada sesi ini!"**

**"Memulai stage berikutnya..."**

Dengan OC adalah hadiah yang didapatkan pada stage tersebut dan bot\_battle\_monster[1] adalah nama monster musuh.

Kasus menyelesaikan semua stage

**"Selamat, Anda berhasil mengalahkan monster {bot\_battle\_monster[1]} !!!"**

**"STAGE CLEARED! Anda akan mendapatkan {OC} OC pada sesi ini!"**

**"Selamat, Anda berhasil menyelesaikan seluruh stage Arena !!!"**

"""

**===== STATS =====**

**Total hadiah : {total\_OC} OC**

**Jumlah stage : 5**

**Damage diberikan : {total\_damage\_dealt}**

**Damage diterima : {total\_damage\_taken}**

"""

Dengan total\_OC adalah jumlah OC yang didapatkan, total\_damage\_dealt adalah jumlah damage yang dikeluarkan selama melakukan battle, dan total\_damage\_taken adalah jumlah damage yang diterima selama melakukan battle.

Untuk kasus kalah pada stage tertentu, user akan diberikan OC sebanyak stage yang diselesaikan

**"Yahhh, Anda dikalahkan monster {bot\_battle\_monster[1]}. Jangan menyerah, coba lagi !!!"**

**"GAME OVER! Sesi latihan berakhir pada stage {bot\_battle\_monster\_level}!"**

"""

===== STATS =====

**Total hadiah : {total\_OC} OC**

**Jumlah stage : {bot\_battle\_monster\_level}**

**Damage diberikan : {total\_damage\_dealt}**

**Damage diterima : {total\_damage\_taken}**

"""

Kemudian jika user memilih kabur, akan muncul output sebagai berikut.

**"GAME OVER! Anda mengakhiri sesi latihan!"**

"""

===== STATS =====

**Total hadiah : {total\_OC} OC**

**Jumlah stage : {bot\_battle\_monster\_level}**

**Damage diberikan : {total\_damage\_dealt}**

**Damage diterima : {total\_damage\_taken}**

"""

## I. F10 - Shop & Currency

Nama command: main

Function: main (table, display, id\_shop, read\_filemonster, read\_filepotion, has\_monsterinven, buy\_monster, buy\_potion)

Masukan:

> Memberikan pilihan aksi yang memenuhi parameter untuk display komponen monster dan potion, buy\_monster, buy\_potion, dan keluar dari program.

Keluaran:

> Program akan mengeluarkan aksi sesuai dengan input yang diberikan oleh user dan memproses pilihan dari user.

Apabila pilihan adalah untuk melihat, maka akan dijalankan **fungsi display** yang akan mengeluarkan daftar yang ada pada monster dan potion.

Apabila pilihan dari user adalah untuk membeli, akan dijalankan **fungsi buy\_monster atau buy\_potion**

Apabila masukan user salah, dikeluarkan

> "**Masukkan anda tidak valid.**"

Apabila user sudah memiliki monster, dikeluarkan

> "**Anda sudah memiliki monster ini, masukkan ID lain!**"

Apabila user berhasil membeli monster, dikeluarkan

> "**Kamu berhasil membeli {type}. Item sudah masuk ke inventory-mu! "**

> "**Sisa O.W.C.A. Coin-mu {owca-price}.**

Apabila user berhasil membeli potion, dikeluarkan

> "**kamu berhasil membeli {quantity} {type\_1} Potion. Item sudah masuk ke inventory-mu!"**

> "**Sisa O.W.C.A. Coin-mu {owca-price}.**

Apabila user tidak berhasil membeli monster/potion, dikeluarkan sesuai dengan keadaan user

> "**O.W.C.A Coin-mu kurang**"

> "**Stok monster tidak mencukupi.**"

> "**Stok potion tidak mencukupi.**"

## m. F11 – Laboratory

Nama command: laboratory

Function: laboratory(array\_user, array\_monster\_inventory, array\_monster, user\_id)

Masukan:

> Masukkan id monster yang ingin di-*upgrade*

> Masukkan konfirmasi *upgrade* (jika monster masih dapat di-*upgrade*)

Keluaran:

> Menampilkan seluruh monster yang dimiliki

> Menampilkan daftar harga untuk *upgrade* monster

> Menampilkan jumlah koin yang dimiliki user

Jika monster yang dipilih untuk di-*upgrade* tidak ditemukan, dicetak:

> '**Monster yang ingin diupgrade tidak ditemukan.**'

Jika monster yang dipilih untuk di-*upgrade* sudah memiliki level maksimum (level 5), dicetak:

> '**Monster yang dipilih sudah memiliki level maksimum.**'

Jika monster yang dipilih untuk di-*upgrade* ditemukan dan belum memiliki level maksimum (level 5), dicetak:

> **f'{array\_monster[int(array\_monster\_inventory[idx[upgrade\_id - 1]][1])][1]} akan diupdate ke level {level + 1}.'**

> '**Harga upgrade ini adalah', str(price) + '.**'

Jika masukan konfirmasi *upgrade* tidak valid, dicetak:

> '**Input tidak valid!**'

Jika *upgrade* jadi dilakukan:

Jika jumlah koin untuk *upgrade* tidak cukup, dicetak:

> '**Upgrade gagal! Anda tidak memiliki cukup koin.**'

Jika jumlah koin untuk *upgrade* cukup, dicetak:

> '**Upgrade berhasil!**'

> **f'{array\_monster[int(array\_monster\_inventory[idx[upgrade\_id - 1]] [1])][1]} sekarang ada di level {level}.'**

Jika *upgrade* tidak jadi dilakukan:

> '**Upgrade dibatalkan.**'

Function: is\_integer (user\_input)

Masukan:

> Menerima user\_input yang berupa string

Keluaran:

> Mengembalikan True jika user\_input adalah sebuah integer dan False jika user\_input bukan merupakan sebuah integer

#### n. F12 - Shop Management

Nama command: shop management

Function: shop management (table, id\_shop, display, read\_filemonster, read\_filepotions, add, update, delete)

Masukan:

> Memberikan pilihan aksi yang memenuhi parameter untuk display komponen monster dan potion, add, update, delete, dan keluar dari program.

Keluaran:

> Program akan mengeluarkan aksi sesuai dengan input yang diberikan oleh user dan memproses pilihan dari user.

Apabila pilihan adalah untuk melihat, maka akan dijalankan **fungsi display** yang akan mengeluarkan daftar yang ada pada monster dan potion.

Apabila pilihan adalah untuk menambah, maka akan dijalankan **fungsi add**.

Saat tidak ada monster yang dapat ditambahkan lagi, dikeluarkan

> "**Tidak ada monster yang dapat ditambahkan!**"

Saat tidak ada potion yang dapat ditambahkan, dikeluarkan

> "**Tidak ada potion yang dapat ditambahkan!**"

Saat masukkan ID salah, dikeluarkan

> "**ID salah! Masukkan lagi**"

Saat masukkan stok tidak valid, dikeluarkan

> "**Stok awal harus lebih besar dari nol!**"

Saat masukkan harga tidak valid, dikeluarkan

> "**Harga harus lebih besar dari nol!**"

Saat monster atau potion berhasil ditambahkan pada data, dikeluarkan

> "**Monster / Potion baru berhasil ditambahkan ke dalam toko!**"

Apabila pilihan adalah untuk update, maka akan dijalankan **fungsi update**.

Saat masukkan ID salah, dikeluarkan

> "**ID salah! Masukkan lagi!**"

Saat masukkan stok tidak valid, dikeluarkan

> "**Stok awal harus lebih besar dari nol!**"

Saat masukkan harga tidak valid, dikeluarkan

> "**Harga harus lebih besar dari nol!**"

Apabila pilihan adalah untuk delete, maka akan dijalankan **fungsi delete**.

Saat masukkan ID salah, dikeluarkan

> "**ID salah! Masukkan ID kembali!**"

Saat masukkan selain iya/tidak, dikeluarkan

> "**Masukkan salah!**"

Apabila pilihan untuk keluar maka akan diberikan keluaran

> "**Anda sudah keluar dari toko. Terima kasih!**"

> "**Terima kasih! Progress terakhir telah disimpan.**"

## o. F13 - Monster Management

Nama command: monster\_management

Function: monster\_management(array\_monster)

Masukan:

> Masukkan aksi yang ingin dilakukan

Jika user ingin membuat monster baru:

> Masukkan tipe atau nama monster

> Masukkan ATK power monster

> Masukkan DEF power monster

> Masukkan HP monster

> Masukkan konfirmasi untuk menambahkan monster baru ke database

Keluaran:

> Menampilkan aksi yang dapat dilakukan di monster\_management

Jika aksi yang dipilih user tidak ditemukan, dicetak:

> '**Aksi yang dipilih tidak ditemukan.**'

Jika user memilih untuk menampilkan semua monster:

> Menampilkan semua monster dalam database

Jika user memilih untuk menambahkan monster baru:

> '**Saatnya membuat monster baru!**'

Jika ada masukan user yang tidak valid, dicetak:

> '**Input tidak valid!**'

> '**Monster baru berhasil dibuat!**'

> '**Type:**', monster\_type

> '**ATK Power:**', monster\_atk\_power

> '**DEF Power:**', monster\_def\_power

> '**HP:**', monster\_hp

Jika konfirmasi user untuk menambahkan monster baru ke database tidak valid, dicetak:

> '**Input tidak valid!**'

Jika user memilih untuk menambahkan monster baru ke database, dicetak:

> '**Monster berhasil ditambahkan ke database!**'

Jika user memilih untuk tidak menambahkan monster baru ke database,

dicetak:

> '**Monster gagal ditambahkan ke database!**'

Function: is\_integer (user\_input)

Masukan:

> Menerima user\_input yang berupa string

Keluaran:

> Mengembalikan True jika user\_input adalah sebuah integer dan False jika user\_input bukan merupakan sebuah integer

#### p. F14 – Load

Nama command: load (bukan merupakan command yang dapat dipanggil saat program berjalan melainkan hanya dapat diakses pada pertama kali program dijalankan).

Function: csv\_to\_array(filepath)

Masukan:

> memberikan nama folder yang ingin di load dengan cara mengetik python main.py <nama folder>

Keluaran:

Melakukan validasi terhadap masukkan pengguna yang berupa nama folder yang ingin di load. Apabila pengguna memberikan nama folder yang valid (ada pada parent folder ./data), program akan melakukan akses pada file csv yang ada pada folder tersebut dengan menggunakan fungsi csv\_to\_array dan program akan berlanjut sesuai dengan kode program utama. Apabila nama folder yang diberikan oleh pengguna tidak ditemukan (tidak ada pada parent folder ./data) maka program akan mencetak 'Folder "'+args.folder+'" tidak ditemukan' dan pengguna akan keluar dari program. Apabila pengguna tidak memberikan nama folder, maka program akan mencetak sebagai berikut:

> "**Tidak ada nama folder yang diberikan!**"

> "**Usage : python main.py <nama\_folder>**"

Dengan: args.folder adalah masukkan nama folder dari pengguna berupa string

#### **q. F15 – Save**

Nama command: save

Procedure: save(array\_user, array\_monster, array\_monster\_inventory, array\_monster\_shop, array\_item\_inventory, array\_item\_shop)

Masukan:

> Memberikan parameter **array\_user**, **array\_monster**, **array\_monster\_inventory**, **array\_monster\_shop**, **array\_item\_inventory**, dan **array\_item\_shop** array\_monster\_inventory, array\_monster\_shop, array\_item\_inventory, array\_item\_shop

> Masukkan nama folder

Keluaran:

Melakukan pengecekan mengenai keberadaan parent folder berupa ./data. Apabila belum ada, akan dibuat folder dengan nama data dan akan mencetak "**Membuat folder data...**".

Melakukan pengecekan mengenai nama folder yang diberikan oleh pengguna. Apabila belum pernah dilakukan penyimpanan pada nama folder yang diberikan oleh pengguna, akan dibuat folder dengan nama sesuai masukkan pengguna dan

mencetak "**Membuat folder data/{folder}**". Apabila sebelumnya telah melakukan penyimpanan data pada folder tersebut atau folder dengan nama yang sesuai dengan masukkan pengguna sudah ada, akan dilakukan *overwrite* data dan mencetak "**Berhasil menyimpan data di folder data/{folder}!**".

Dengan: folder adalah inputan nama folder berupa string.

#### r. F16 – Exit

Nama command: exit

Function: exit()

Masukan:

> ‘Apakah Anda mau melakukan penyimpanan file yang sudah diubah?  
(Y/N)’

Keluaran:

Program akan memvalidasi masukan pilihan dari pengguna, yaitu pilihan ingin menyimpan file (melakukan save) atau tidak. Masukkan akan valid jika berupa “Y”, “y”, “N”, atau “n”. Program akan terus menanyakan masukan pilihan kepada pengguna sampai masukan valid.

Jika masukkan dari pengguna adalah “Y” atau “y”, maka program akan mengembalikan boolean bernilai True. Boolean ini akan digunakan pada program utama untuk melakukan pemanggilan program save(array\_user, array\_monster, array\_monster\_inventory, array\_monster\_shop, array\_item\_inventory, array\_item\_shop) dan barulah kemudian keluar dari program,

Jika masukkan dari pengguna adalah “N” atau “n”, maka program akan mengembalikan boolean bernilai False. Dengan boolean bernilai False ini, pengguna akan langsung keluar dari program berdasarkan kode pada program utama.

## V. Desain Kamus Data

### a. Program Utama

#### KAMUS LOKAL

```

logged_in           : boolean
username, role, perintah : string
oc_received         : int

procedure register (logged_in: bool, array_user: list, array_monster: list,
array_monster_inventory: list)
function login (logged_in: boolean, user_id: integer, array_user: array of (array of
string) → boolean, string

procedure logout (input log_out: integer, check_login: boolean, output: string)

procedure help(login: bool, role: str, username: str)

procedure inventory (user_id: int, item_inventory: list, status_potion: list,
nama_monster: str)

function battle (monsterarray: array of data_monster, monster_inventory: array of
data_monster_inventory, item_inventory: array of data_item_inventory, user_id: integer,
bot_battle_monster_level: integer, arena: boolean) → boolean

function arena (monsterarray: array of data_monster, monster_inventory: array of
data_monster_inventory, item_inventory: array of data_item_inventory, user_id: integer,
stage: integer) → integer

function shopmain(data_monstershop: list, data_monster: list, data_potionshop: list,
data_monster_inventory: list, data_potion_inventory: list, data_owca: list, user_id: int)

function laboratory(array_user: list, array_monster_inventory: list, array_monster: list,
user_id: int) → (list, list)

shop_management(data_monstershop: list, data_monster: list, data_potionshop: list)

function monster_management(array_monster: list) → list

procedure save (input array_user: list, input array_monster: list, input
array_monster_inventory: list, input array_monster_shop: list, input
array_item_inventory: list, input array_item_shop: list)

procedure exit() → boolean

```

### b. F00 - Random Number Generator

#### KAMUS LOKAL

```

a, c, m           : integer
seed              : None
x0, x1, x2, x_prev : int

```

num\_range : array [x1, x2] of integer

function lcg () → integer  
 {I.S tidak terdefinisi; F.S menghasilkan sebuah nilai integer acak}

function generate\_number (x\_prev: integer, num\_range: array [x1, x2] of integer) → integer

{I.S terdefinisi x\_prev (x\_prev diambil dari function lcg()) dan num\_range; F.S menghasilkan nilai integer acak dari range tersebut}

### c. F01 – Register

KAMUS LOKAL

valid, ada, pw: boolean

username, password, nama\_monster\_pilihan, monster\_pilihan: string

maks, batas\_awal, batas\_akhir, count: integer

list\_usn, list\_pw: array of character

list\_ascii, list\_pw\_ascii: array ist of integer

data\_user, data\_monster\_inventory: matrix

function is\_integer (user\_input: string) → boolean  
 {I.S user\_input terdefinisi; F.S menghasilkan boolean}

procedure register (logged\_in: bool, array\_user: list, array\_monster: list, array\_monster\_inventory: list)

{I.S input logged\_in: bool, array\_user: list, array\_monster: list, array\_monster\_inventory: list}

function cekPassword (password: str) → str

{I.S password terdefinisi; F.S mengembalikan password}

### d. F02 – Login

function login(logged\_in: boolean, user\_id: integer, array\_user: array of (array of string)) → boolean, string

KAMUS LOKAL

username, password: string

username\_found: boolean

user\_id: integer

### e. F03 – Logout

KAMUS LOKAL

log\_out: integer

check\_login: boolean

Procedure logout (input log\_out: integer, check\_login: boolean, output: string)

{I.S check\_login, log\_out terdefinisi, F.S terminasi program}

#### f. F04 - Menu & Help

KAMUS LOKAL

login : boolean

role, username : string

procedure help(login: bool, role: str, username: str)

#### g. F05 – Monster

KAMUS LOKAL

type data\_monster: < id, type, atk\_power, def\_power, hp: string >

type data\_monster\_battle: < id: string, type: string, atk\_power, def\_power, hp: float >

attackpower : float

a, c, m : integer

seed : None

x0, x1, x2, x\_prev, x : integer

num\_range : array [x1, x2] of integer

tanda, pengali, level : integer

damage, defpower : float

enemydamage : float

damage\_reduction : float

user\_battle\_monster : data\_monster

id : string

function attack\_user (attackpower: float) → float

{I.S attackpower terdefinisi; F.S mengembalikan nilai float}

function attack\_bot (attackpower: float) → float

{I.S attackpower terdefinisi; F.S mengembalikan nilai float}

function defense (defpower: float, enemydamage: float) → float

{I.S defpower terdefinisi, enemydamage terdefinisi; F.S mengembalikan nilai float}

level\_buff(level: int, user\_battle\_monster: data\_monster) → data\_monster\_battle

{I.S level, user\_battle\_monster terdefinisi; F.S mengembalikan bentukan type data\_monster\_battle}

procedure gambar\_monster (input id: string, output: string)

{I.S id terdefinisi; F.S output string terdefinisi}

### **h. F06 – Potion**

#### KAMUS LOKAL

```

type data_item_inventory: < user_id, type, quantity: string >
type modif_item_inventory: < user_id, type, quantity: string, index: string >

user_input, txt1, txt2 : string
char : character
user_id, count, potion_dipilih, code, index : integer
item_inventory : array of data_item_inventory
pilihan_potion : array of modif_item_inventory
status_potion : dictionary {key:str, value:int}
function pilih_potion (user_id: int, item_inventory: list) → list:
function use_potion (user_id: int, item_inventory: list, status_potion: list,
nama_monster: str) → (list, list, int)

```

### **i. F07 – Inventory**

#### KAMUS LOKAL

```

type user: < id, username, password, role, oc: string >
type detail_item_inventory: < “Potion”, type, quantity: string >
type detail_monster_inventory: < “Monster”, user_id, monster_id, level: string >
type data_monster: < id, type, atk_power, def_power, hp: string

user_input, aksi, tipe : string
char : character
count, idx : integer
detail_item : detail_item_inventory
detail_item : detail_monster_inventory

user_inventory : array of detail_item_inventory and
array of detail_monster_inventory

procedure inventory (user_id: int, item_inventory: list, status_potion: list,
nama_monster: str)

```

### **j. F08 – Battle**

#### KAMUS LOKAL

```

type data_monster: < id, type, atk_power, def_power, hp: string >
type data_user: < id, username, password, role, oc: string >
type data_monster_inventory: < user_id, monster_id, level: string >

```

```

type data_item_inventory: < user_id, type, quantity: string >

user_input : string
char : character
user_id, count, level : integer
monsterarray : array of
data_monster_inventory
monster_inventory : array of
data_monster_inventory
pilihan_monster : data_monster
user_battle_monster, enemy_battle_monster : data_monster
turn_counter : int
monster_name : string
bot_monster_name, user_monster_name : string
user_damage, damage_reduction, damage_after_reduction : integer
base_HP_enemy_monster, user_battle_monster_level : integer
enemy_battle_monster_level : integer
tanda : integer
monster_muncul_id : integer
user : array of data_user
arena, win, kabur : boolean
monster_dipilih : string
bot_battle_monster_level : integer
base_atk_user_monster, base_def_user_monster, base_HP_user_monster : integer
base_atk_bot_monster, base_def_bot_monster, base_HP_bot_monster : integer
total_damage_dealt, total_damage_taken : integer
user_action : string
damage, damage_reduction, damage_after_reduction : float
code : integer
battle_lagi : character

function is_integer (user_input: string) → boolean
{I.S user_input terdefinisi; F.S menghasilkan boolean}

function pilih_monster(user_id: integer, monster_inventory: array of data_monster,
monsterarray: array of data_monster) → data_monster
{I.S terdefinisi user_id, monster_inventory, monsterarray; F.S menghasilkan array baru}

procedure gambar_monster_user (output: string)
{I.S tidak terdefinisi; F.S menghasilkan output string}

procedure monster_dipilih_user (input level: integer, user_battle_monster: array of
data_monster, output: string)
{I.S terdefinisi level, user_battle_monster; F.S terdefinisi output string}

procedure output_user_turn (input turn_counter: integer, monster_name: string, output:
string)
{I.S terdefinisi turn_counter, monster_name; F.S terdefinisi output string}

```

```

procedure output_bot_turn (input turn_counter: integer, bot_monster_name: string,
user_monster_name: string, output: string)
{I.S terdefinisi turn_counter, bot_monster_name, user_monster_name; F.S terdefinisi
output string}

procedure attack_result (input enemy_battle_monster: data_monster,
user_battle_monster: data_monster, user_damage: integer, damage_reduction: integer,
damage_after_reduction: integer, base_HP_enemy_monster: integer,
user_battle_monster_level: integer, enemy_battle_monster_level: integer, output: string)
{I.S terdefinisi enemy_battle_monster, user_battle_monster, user_damage,
damage_reduction, damage_after_reduction, base_HP_enemy_monster,
user_battle_monster_level, enemy_battle_monster_level; F.S terdefinisi output string}

procedure user_attack_turn (input user_monster_name: string, bot_monster_name: string,
output: string)
{I.S terdefinisi user_monster_name, bot_monster_name; F.S terdefinisi output string}

procedure gambar_monster_musuh (output: string)
{I.S tidak terdefinisi; F.S terdefinisi output string}

procedure munculmonster (input monsterarray: array of data_monster,
monster_muncul_id: integer, output: string)
{I.S terdefinisi monsterarray, monster_muncul_id; F.S terdefinisi output string}

function dapet_duit (user: array of data_user) → array of data_user
{I.S user terdefinisi, F.S terdefinisi mengembalikan array of data_user}

function battle (monsterarray: array of data_monster, monster_inventory: array of
data_monster_inventory, item_inventory: array of data_item_inventory, user_id: integer,
bot_battle_monster_level: integer, arena: boolean) → boolean
{I.S terdefinisi monsterarray, monster_inventory, item_inventory, user_id,
bot_battle_monster_level, arena; F.S mengembalikan nilai boolean}

```

#### k. F09 – Arena

##### KAMUS LOKAL

```

type data_monster: < id, type, atk_power, def_power, hp: string >
type data_monster_inventory: < user_id, monster_id, level: string >
type data_item_inventory: < user_id, type, quantity: string >

monsterarray : array of data_monster
monster_inventory : array of data_monster_inventory
item_inventory : array of data_item_inventory
user_id, stage : integer
total_OC, total_damage_dealt, total_damage_taken, damage_dealt, damage_taken : integer

```

```

win          :
boolean
OC          : array [0,..,4] of integer

function arena (monsterarray: array of data_monster, monster_inventory: array of
data_monster_inventory, item_inventory: array of data_item_inventory, user_id:
integer, stage: integer) → integer
{I.S terdefinisi monsterarray, monster_inventory, item_inventory, user_id, stage; F.S
mengembalikan nilai integer}

```

## I. F10 - Shop & Currency

### KAMUS LOKAL

```

choice, item_type, num_spaces, item, item_type: string
user_id, item_id, quantity, jumlah, item_length: integer
line, id, data_monsternew, data_potion, id, type, atk_power, def_power, hp, stok, price,
owca, type_1, stok_1, harga_1: array

procedure display(input display: array of data_monstershop, array of data_monster,
array of data_potionshop, read_filemonster: array of data_monstershop, array of
data_monster, read_filepotion: array of data_potionshop, output: list of strings)

{I.S terdefinisi def_display, read_filemonster, read_filepotion; F.S terdefinisi output list
of strings}

procedure buy_monster(input has_monsterinven: array of data_inventory, array of
user_id, buy_monster: array of data_monster inventory, array of data_monster, array of
data_monstershop, array of data_shop, array of user_id, output: string)

{I.S terdefinisi has_monsterinven, buy_monster; F.S terdefinisi output string}

procedure buy_potion(input buy_potion: array of data_potion_inventory, array of
data_potionshop, array of data_owca, array of user_id, output: string)

{I.S terdefinisi buy_potion; F.S terdefinisi output string}

function read_filemonster(data_monstershop: array of data, data_monster: array of
data) → 2D Matrix of Strings

{I.S terdefinisi data_monstershop, data_monster; F.S terdefinisi mengembalikan nilai
list of strings}

function read_filepotion(data_potionshop: array of data) → 2D Matrix of Strings

{I.S terdefinisi data_potionshop; F.S terdefinisi mengembalikan nilai list of strings}

function has_monsterinven(id: String) → Boolean

{I.S terdefinisi id; F.S terdefinisi mengembalikan nilai boolean}

```

## m. F11 – Laboratory

function laboratory(array\_user: list, array\_monster\_inventory: list, array\_monster: list, user\_id: int) → (list, list)

KAMUS LOKAL

monster\_amt, coin\_amt, price, monster\_id, upgrade\_id, level: integer

confirm: string

idx: array of integer

function is\_integer(user\_input: str) → bool

KAMUS LOKAL

-

#### n. F12 - Shop Management

KAMUS LOKAL

Choice, item type, add\_things, id, stok, harga, update, delete, check, id\_delete: string

item\_length: integer

line, data\_monsternew, data\_potion, data\_monsterid, id\_add, monster\_shop, data\_potionid, id\_tambah, potion\_baru: array

id: boolean

procedure display(input display: array of data\_monstershop, array of data\_monster, array of data\_potionshop, read\_filemonster: array of data\_monstershop, array of data\_monster, read\_filepotion: array of data\_potionshop, output: list of strings)

{I.S terdefinisi def\_display, read\_filemonster, read\_filepotion; F.S terdefinisi output list of strings}

procedure add(input add: data\_monstershop array of data\_monster, array of data\_monster, array of data\_potionshop, output: string)

{I.S terdefinisi add; F.S terdefinisi output string}

procedure update(input update: data\_monstershop array of data\_monster, array of data\_potionshop; output: string)

{I.S terdefinisi update; F.S terdefinisi output string}

procedure delete(input delete: data\_monstershop array of data\_monster, array of data\_potionshop; output: string)

{I.S terdefinisi delete; F.S terdefinisi output string}

### F13 - Monster Management

function monster\_management(array\_monster: list) → list

KAMUS LOKAL

i: integer

new\_monster: array of string

unique\_type: boolean

action, monster\_type, monster\_def\_power, monster\_atk\_power, monster\_hp, confirm:  
string

function is\_integer(user\_input: str) → bool

KAMUS LOKAL

-

### o. F14 – Load

KAMUS LOKAL

folder\_path, kata: string

check\_folder: boolean

### p. F15 – Save

KAMUS LOKAL

folder: string

parent\_check, folder\_check: boolean

list\_folder : array of character

list\_ascii : array of integer

procedure save (input array\_user: list, input array\_monster: list, input  
array\_monster\_inventory: list, input array\_monster\_shop: list, input  
array\_item\_inventory: list, input array\_item\_shop: list)

{I.S array\_user, array\_monster, array\_monster\_inventory, array\_monster\_shop,  
array\_item\_inventory, array\_item\_shop terdefinisi; F.S data tersimpan pada parent  
folder}

### q. F16 – Exit

KAMUS LOKAL

reassure: string

procedure exit() → boolean

{I.S reassure tidak terdefinisi; F.S boolean terdefinisi}

## VI. Desain Dekomposisi Algoritmik dan Fungsional Program

### a. Program Utama

#### b. F00 - Random Number Generator

```
function lcg () → integer
{I.S tidak terdefinisi; F.S menghasilkan sebuah nilai integer acak}

function generate_number (x_prev: integer, num_range: array [x1, x2] of integer) →
integer
{I.S terdefinisi x_prev (x_prev diambil dari function lcg()) dan num_range; F.S
menghasilkan nilai integer acak dari range tersebut}
```

#### c. F01 – Register

```
function is_integer (user_input: string) → boolean
{I.S user_input terdefinisi; F.S menghasilkan boolean}

procedure register (logged_in: bool, array_user: list, array_monster: list,
array_monster_inventory: list)
{I.S input logged_in: bool, array_user: list, array_monster: list,
array_monster_inventory: list; F.S array_user dan array_monster_inventory terjadi
perubahan}

function cekPassword (password: str) → str
{I.S password terdefinisi; F.S mengembalikan password}
```

#### d. F02 – Login

```
function login(logged_in: boolean, user_id: integer, array_user: array of (array of
string) → boolean, string
{ Memasukkan user ke dalam game dengan menerima masukan berupa username dan
password serta mengembalikan variabel logged_in dan user_id }
```

#### e. F03 – Logout

Procedure logout (input log\_out: integer, output: string)

{I.S login, log\_out terdefinisi, F.S terminasi program}

#### f. F04 - Menu & Help

```
procedure help (login: bool, role: str, username: str)
{ Memberikan output semua command yang dapat dilakukan user, dan catatan
panduan bagi user}
```

#### g. F05 – Monster

```
function attack_user (attackpower: float) → float
{I.S attackpower terdefinisi; F.S mengembalikan nilai float}
```

```

function attack_bot (attackpower: float) → float
{I.S attackpower terdefinisi; F.S mengembalikan nilai float}

function defense (defpower: float, enemydamage: float) → float
{I.S defpower terdefinisi, enemydamage terdefinisi; F.S mengembalikan nilai float}

level_buff(level: int, user_battle_monster: data_monster) → data_monster
{I.S level, user_battle_monster terdefinisi; F.S mengembalikan bentukan type
data_monster}

procedure gambar_monster (input id: string, output: string)
{I.S id terdefinisi; F.S output string terdefinisi}

```

#### **h. F06 – Potion**

```

function pilih_potion (user_id: int, item_inventory: list) → list:
{ Memberikan output seluruh potion yang dimiliki user. Mengembalikan list
potion yang dimiliki user}

function use_potion (user_id: int, item_inventory: list, status_potion: list,
nama_monster: str) → (list, list, int)

{ Menerima masukan dari user mengenai potion yang ingin dipakai serta
mengembalikan inventory yang sudah berkurang, status potion yang sudah terpakai,
dan sebuah kode yang akan digunakan di battle}

```

#### **i. F07 – Inventory**

```

procedure user_id: int, item_inventory: list, status_potion: list, nama_monster: str)
{Memberikan output seluruh item yang dimiliki oleh user (OC coin, monster, dan
potion)}

```

#### **j. F08 – Battle**

```

function is_integer (user_input: string) → boolean
{I.S user_input terdefinisi; F.S menghasilkan boolean}

function pilih_monster(user_id: integer, monster_inventory: array of data_monster,
monsterarray: array of data_monster) → data_monster
{I.S terdefinisi user_id, monster_inventory, monsterarray; F.S mengasilkan array baru}

procedure gambar_monster_user (output: string)
{I.S tidak terdefinisi; F.S menghasilkan output string}

procedure monster_dipilih_user (input level: integer, user_battle_monster: array of
data_monster, output: string)
{I.S terdefinisi level, user_battle_monster; F.S terdefinisi output string}

procedure output_user_turn (input turn_counter: integer, monster_name: string, output:
string)
{I.S terdefinisi turn_counter, monster_name; F.S terdefinisi output string}

```

```

procedure output_bot_turn (input turn_counter: integer, bot_monster_name: string,
user_monster_name: string, output: string)
{I.S terdefinisi turn_counter, bot_monster_name, user_monster_name; F.S terdefinisi
output string}

procedure attack_result (input enemy_battle_monster: data_monster,
user_battle_monster: data_monster, user_damage: integer, damage_reduction: integer,
damage_after_reduction: integer, base_HP_enemy_monster: integer,
user_battle_monster_level: integer, enemy_battle_monster_level: integer, output: string)
{I.S terdefinisi enemy_battle_monster, user_battle_monster, user_damage,
damage_reduction, damage_after_reduction, base_HP_enemy_monster,
user_battle_monster_level, enemy_battle_monster_level; F.S terdefinisi output string}

procedure user_attack_turn (input user_monster_name: string, bot_monster_name: string,
output: string)
{I.S terdefinisi user_monster_name, bot_monster_name; F.S terdefinisi output string}

procedure gambar_monster_musuh (output: string)
{I.S tidak terdefinisi; F.S terdefinisi output string}

procedure munculmonster (input monsterarray: array of data_monster,
monster_muncul_id: integer, output: string)
{I.S terdefinisi monsterarray, monster_muncul_id; F.S terdefinisi output string}

function dapet_duit (user: array of data_user) → array of data_user
{I.S user terdefinisi, F.S terdefinisi mengembalikan array of data_user}

function battle (monsterarray: array of data_monster, monster_inventory: array of
data_monster_inventory, item_inventory: array of data_item_inventory, user_id: integer,
bot_battle_monster_level: integer, arena: boolean) → boolean
{I.S terdefinisi monsterarray, monster_inventory, item_inventory, user_id,
bot_battle_monster_level, arena; F.S mengembalikan nilai boolean}

```

#### k. F09 - Arena

```

function arena (monsterarray: array of data_monster, monster_inventory: array of
data_monster_inventory, item_inventory: array of data_item_inventory, user_id: integer,
stage: integer) → integer
{I.S terdefinisi monsterarray, monster_inventory, item_inventory, user_id, stage; F.S
mengembalikan nilai integer}

```

#### l. F10 - Shop & Currency

```

procedure display(input display: array of data_monstershop, array of data_monster,
array of data_potionshop, read_filemonster: array of data_monstershop, array of
data_monster, read_filepotion: array of data_potionshop, output: list of strings)
{I.S terdefinisi def_display, read_filemonster, read_filepotion; F.S terdefinisi output list
of strings}

```

```

procedure buy_monster(input has_monsterinven: array of data_inventory, array of
user_id, buy_monster: array of data_monster inventory, array of data_monster, array of
data_monstershop, array of data_shop, array of user_id, output: string)

{I.S terdefinisi has_monsterinven, buy_monster; F.S terdefinisi output string}

procedure buy_potion(input buy_potion: array of data_potion_inventory, array of
data_potionshop, array of data_owca, array of user_id, output: string)

{I.S terdefinisi buy_potion; F.S terdefinisi output string}

function read_filemonster(data_monstershop: array of data, data_monster: array of
data) → 2D Matrix of Strings

{I.S terdefinisi data_monstershop, data_monster; F.S terdefinisi mengembalikan nilai
list of strings}

function read_filepotion(data_potionshop: array of data) → 2D Matrix of Strings

{I.S terdefinisi data_potionshop; F.S terdefinisi mengembalikan nilai list of strings}

function has_monsterinven(id: String) → Boolean

{I.S terdefinisi id; F.S terdefinisi mengembalikan nilai boolean}

```

#### **m. F11 - Laboratory**

```

function laboratory (array_user: list, array_monster_inventory: list, array_monster: list,
user_id: integer) → (list, list)

{ Menerima masukan user mengenai monster yang ingin di-upgrade serta meng-
upgrade monster jika user memiliki koin yang cukup serta monster belum berada pada
level maksimum }

function is_integer(user_input: string) → boolean

{ Menerima sebuah masukan berupa string dan mengembalikan True jika input tersebut
merupakan sebuah integer serta False jika tidak }

```

#### **n. F12 - Shop Management**

```

procedure display(input display: array of data_monstershop, array of data_monster,
array of data_potionshop, read_filemonster: array of data_monstershop, array of
data_monster, read_filepotion: array of data_potionshop, output: list of strings)

{I.S terdefinisi def_display, read_filemonster, read_filepotion; F.S terdefinisi output list
of strings}

procedure add(input add: data_monstershop array of data_monster, array of
data_monster, array of data_potionshop, output: string)

{I.S terdefinisi add; F.S terdefinisi output string}

procedure update(input update: data_monstershop array of data_monster, array of
data_potionshop; output: string)

```

{I.S terdefinisi update; F.S terdefinisi output string}

procedure delete(input delete: data\_monstershop array of data\_monster, array of data\_potionshop; output: string)

{I.S terdefinisi delete; F.S terdefinisi output string}

**o. F13 - Monster Management**

function monster\_management(array\_monster: list) → list

{ Menampilkan semua monster yang ada dalam game serta menambahkan monster baru ke dalam game berdasarkan masukan user }

function is\_integer(user\_input: string) → boolean

{ Menerima sebuah masukan berupa string dan mengembalikan True jika input tersebut merupakan sebuah integer serta False jika tidak }

**p. F14 – Load**

{I.S isi folder (file csv) terdefinisi; F.S matriks data sesuai file}

**q. F15 – Save**

procedure save (input array\_user: list, input array\_monster: list, input array\_monster\_inventory: list, input array\_monster\_shop: list, input array\_item\_inventory: list, input array\_item\_shop: list)  
{I.S array\_user, array\_monster, array\_monster\_inventory, array\_monster\_shop, array\_item\_inventory, array\_item\_shop terdefinisi; F.S data tersimpan pada parent folder}

**r. F16 – Exit**

function exit() → boolean

{I.S reassure tidak terdefinisi; F.S boolean terdefinisi}

## VII. Spesifikasi Program

### a. Program Utama

```

{ Inisialisasi }

logged_in ← False
username ← ""
role ← ""
user_id ← 0

{ Meminta perintah berikutnya }

output('Masukkan perintah!\nKetik "HELP" untuk melihat menu')
perintah ← input("">>>> ").upper()
output()

while True do

    if logged_in == False then

        if perintah == "HELP" or perintah == "MENU" then

            help(logged_in, role, username)

        elif perintah == "REGISTER" then

            register(logged_in, array_user, array_monster, array_monster_inventory)

        elif perintah == "LOGIN" then

            logged_in, user_id ← login(logged_in, user_id, array_user)

        elif perintah == "LOGOUT" then

            logged_in ← logout(logged_in)

        elif perintah == "SAVE" then

            save(array_user, array_monster, array_monster_inventory,
array_monster_shop, array_item_inventory, array_item_shop)

        elif perintah == "EXIT" then

            keluar ← exit()

        if keluar == True then

            save(array_user, array_monster, array_monster_inventory,
array_monster_shop, array_item_inventory, array_item_shop)

```

```

output("Sampai jumpa kembali!")

elif keluar == False then

    output("Sampai jumpa kembali!")

    break

else

    output("Perintah tidak valid! Pastikan input sudah benar.")

else { logged_in == True }

    { Mendapatkan role user yang sedang login }

    username ← array_user[user_id][1]

    role ← array_user[user_id][3]

    if role == "admin" then

        if perintah == "HELP" or perintah == "MENU" then

            help(logged_in, role, username)

        elif perintah == "REGISTER" then

            register(logged_in, array_user, array_monster, array_monster_inventory)

        elif perintah == "LOGIN" then

            logged_in, user_id ← login(logged_in, user_id, array_user)

        elif perintah == "LOGOUT" then

            logged_in ← logout(logged_in)

        elif perintah == "SHOP MANAGEMENT" then

            array_monster_shop, array_item_shop ←
shop_management(array_monster_shop, array_monster, array_item_shop)

        elif perintah == "MONSTER MANAGEMENT" then

            array_monster ← monster_management(array_monster)

        elif perintah == "SAVE" then

            save(array_user, array_monster, array_monster_inventory,
array_monster_shop, array_item_inventory, array_item_shop)

        elif perintah == "EXIT" then

            keluar ← exit()

            if keluar == True then

```

```

    save(array_user, array_monster, array_monster_inventory,
array_monster_shop, array_item_inventory, array_item_shop)

    output("Sampai jumpa kembali admin!")

elif keluar == False then

    output("Sampai jumpa kembali admin!")

    break

else

    output("Perintah tidak valid!")

else { role == "agent" }

    if perintah == "REGISTER" then

        register(logged_in, array_user, array_monster, array_monster_inventory)

    elif perintah == "LOGIN" then

        logged_in, user_id ← login(logged_in, user_id, array_user)

    elif perintah == "LOGOUT" then

        logged_in ← logout(logged_in)

    elif perintah == "HELP" or perintah == "MENU" then

        help(logged_in, role, username)

    elif perintah == "INVENTORY" then

        inventory(array_user[user_id], array_item_inventory,
array_monster_inventory, array_monster)

    elif perintah == "BATTLE" then

        array_user[user_id] ← battlemain(array_monster, array_monster_inventory,
array_item_inventory, user_id, 1, False, 0, [], array_user)

    elif perintah == "ARENA" then

        oc_received ← arena(array_monster, array_monster_inventory,
array_item_inventory, user_id, 1)

        oc_received += int(array_user[user_id][4])

        array_user[user_id][4] ← str(oc_received)

    elif perintah == "SHOP" then

        array_monster_shop, array_item_shop, array_monster_inventory,
array_item_inventory, array_user ← shopmain(array_monster_shop, array_monster,
array_item_shop)

```

```

array_item_shop, array_monster_inventory, array_item_inventory, array_user,
user_id)

elif perintah == "LABORATORY" then

    array_monster_inventory, array_user ← laboratory(array_user,
array_monster_inventory, array_monster, user_id)

elif perintah == "SAVE" then

    save(array_user, array_monster, array_monster_inventory,
array_monster_shop, array_item_inventory, array_item_shop)

elif perintah == "EXIT" then

    keluar ← exit()

    if keluar == True then

        save(array_user, array_monster, array_monster_inventory,
array_monster_shop, array_item_inventory, array_item_shop)

        output("Sampai jumpa kembali Agent!")

    elif keluar == False then

        output("Sampai jumpa kembali Agent!")

    break

else

    output("Perintah tidak valid!")

output()

output('Masukkan perintah!\nKetik "HELP" untuk melihat menu')

perintah ← input(">>> ").upper()

```

### b. F00 - Random Number Generator

```

import os
import time

{Realisasi fungsi dan prosedur}
function lcg () → integer

    A ← 4827
    c ← 0
    m ← 2**31-1
    seed ← None

```

```

if seed = None then
    x0 ← os.getpid()*100 + time.time()*100

else
    x0 ← seed

    x_prev ← (a * x0 + c) mod m
    → x_prev

function generate_number (x_prev: integer, num_range: array [x1, x2] of integer) →
integer

    a ← 48271
    c ← 0
    m ← 2**31-1
    x_prev ← (a * x_prev + c) mod m

    if num_range = None:
        → x_prev

    else:
        → (x_prev / (m - 1)) * (num_range[1] - num_range[0]) +
            num_range[0]

```

### c. F01 – Register

```

function is_integer(user_input: str):
    i traversal [user_input]

    if(ord(char) < ord('0')) or (ord(char) > ord('9')) then
        --> False

    --> True

```

```

procedure register(logged_in: bool, array_user: list, array_monster: list,
array_monster_inventory: list):

    output("Selamat datang di register!\n")

    while True do

        output("Pilih aksi dibawah ini:")
        output("1. Register\n2. Keluar\n")
        input(aksi)

        if aksi = "1" then

```

```

input(username)

input(password)

if logged_in = True then

    output("Register gagal")

    output("Anda masih login. Silahkan logout terlebih dahulu sebelum
melakukan register.")

else

    list_usn <- []
    i traversal [username]
    list_usn.append(letter)

    list_ascii <- [0 for i in range(len(username))]
    i traversal [0..<len(username)]
    list_ascii[i] <- ord(username[i])

    valid <- False

    i traversal [list_ascii]
    if i==45 or i==95 or (i>=48 and i<=57) or (i>=65 and i<=90) or (i>=97
and i<=122) then

        valid <- True

    else

        valid <- False

    if valid = False then

        output("Username hanya boleh berisi alfabet, angka, underscore, dan
strip.")

    else

        ada <- False

        i traversal [0..<len(array_user)]
        if array_user[i][1] = username then

            output(f"Username sudah terpakai, silahkan gunakan username
lain.\n")

```

```

ada <- True
else
    ada <- False

if ada = False then
    cekPassword(password)
    data_user <- [0 for i in range(5)]
    maks <- -1
    i traversal [1..<len(array_user)]
    if int(array_user[i][0]) < maks then
        maks <- int(array_user[i][0])
    data_user <- [str(int(array_user[maks][0]) + 1), username, password,
    'agent', '0']
    array_user.append(data_user)

output("Silahkan pilih salah satu monster sebagai monster awalmu.")
batas_awal <- 9999
batas_akhir <- -1
i traversal [0..<len(array_monster)]
if i != 0 and i <= 3 then
    output(i+".",end=" ")
    output(array_monster[i][1])
    if i <= batas_awal then
        batas_awal <- i
    if i >= batas_akhir then
        batas_akhir <- i
output("Masukkan angka dari monster yang dipilih.")
input(monster_pilihan)
while True do
    if monster_pilihan = " then

```

```

output("Monster harus dipilih dan tidak boleh kosong. Masukkan
angka untuk monster yang ingin pilih.\n")

else if(is_integer(monster_pilihan)) and (int(monster_pilihan) <=
3 and int(monster_pilihan) >= 1) then

    monster_pilihan = int(monster_pilihan)

    break

else

    output("Input tidak valid. Coba pilih ulang.\n")

    input(monster_pilihan)

nama_monster_pilihan <- array_monster[monster_pilihan][1]

output("Selamat datang Agent. Mari kita mengalahkan Dr. Asep
Spakbor !")

data_monster_inventory <- [data_user[0], str(monster_pilihan), '1']

array_monster_inventory.append(data_monster_inventory)

break

else if aksi = "2" then

    output("Anda keluar dari register!")

    break

else

    output("Aksi tidak valid! Pilih ulang aksi yang ingin dilakukan!")

function cekPassword(password: str) -> str

pw <- True

while pw do

    list_pw <- []

    i traversal [password]

    list_pw.append(i)

    list_pw_ascii <- [0 for i traversal [0..<len(password)]]]

    i traversal [0..<len(password)]

```

```

list_pw_ascii[i] <- ord(password[i])
count <- 0
i traversal [list_pw_ascii]
if i = 32 then
    Count <- count + 1
if count = 0 then
    pw <- False
else
    output("Password tidak boleh menggunakan spasi. Masukkan kembali
password baru.")
    input(password)
if password = " " then
    output("Password tidak boleh kosong. Masukkan kembali password baru.")
    input(password)
--> password

```

#### d. F02 – Login

function login(logged\_in: boolean, user\_id: integer, array\_user: array of (array of string) → boolean, string

KAMUS LOKAL

username, password: string

username\_found: boolean

user\_id: integer

ALGORITMA

if not (logged\_in) then

username\_found <- False

output('>>> Username: ')

input(username)

output('>>> Password: ')

input(password)

output()

```

i traversal [1...(length(array_user) - 1)]
    if (array_user[i][1] = username) and (array_user[i][2] = password) then
        output('Login berhasil!')
        output('Selamat datang', array_user[i][3], username + '!')
        output('Ketik "HELP" untuk melihat list command yang dapat kamu
              gunakan.')
        user_id <- i
        → True, user_id { return login status dan user_id }

    else if (array_user[i][1] = username) and (array_user[i][2] /= password) then
        output('Password kamu salah!')
        username_found <- True
        break

    if not (username_found) then
        output('Username kamu belum terdaftar!')
        output('Lakukan registrasi terlebih dahulu sebelum melakukan login.')
    else { logged_in = True }

        output('Login gagal!')
        output('Kamu sudah login dengan akun', array_user[user_id][1] + ', lakukan
              logout terlebih dahulu sebelum melakukan login kembali.')
    → logged_in, user_id

```

#### e. F03 – Logout

{realisasi prosedur}

function check\_login() → Boolean

→ False

procedure logout()

if check\_login() then

log\_out <- input("Apakah anda ingin keluar dari permainan? (iya / tidak):
").lower()

if log\_out = "iya" then

```

output("User berhasil melakukan logout.")

else

    output("Logout dibatalkan.")

else

    output("User belum login.")

```

#### f. F04 - Menu & Help

```

procedure help(login: bool, role: str, username: str):

    output()

    output("===== HELP =====")

    output()

    if login then {sudah login}

        if role == "admin" then

            output("Selamat datang, Admin! Berikut adalah hal-hal yang dapat kamu
lakukan:")

            output()

            output("\\t1. Logout: Keluar dari akun yang sedang digunakan")
            output("\\t2. Shop Management: Melakukan manajemen pada SHOP sebagai
tempat jual beli peralatan Agent")
            output("\\t3. Monster Management: Melakukan manajemen pada database
monster")
            output("\\t4. Save: Menyimpan data ke dalam folder")
            output("\\t5. Exit: Keluar dari program")

        else {role == "agent"}

            output(f'Halo Agent {username}! Kamu memanggil command HELP. Kamu
memilih jalan yang benar, semoga kamu tidak sesat kemudian. Berikut adalah hal-hal
yang dapat kamu lakukan sekarang:")

            output()

            output("\\t1. Logout: Keluar dari akun yang sedang digunakan")
            output("\\t2. Inventory: Melihat detail monster, potion, dan OC yang dimiliki
oleh Agent")
            output("\\t3. Battle: Bertarung 1v1 melawan monster")
            output("\\t4. Arena: Melatih monster yang dimiliki oleh Agent")
            output("\\t5. Shop: Membeli item berupa monster atau potion")

```

```

output("\\t6. Laboratory: Meningkatkan level monster yang dimiliki oleh
Agent")

output("\\t7. Save: Menyimpan data ke dalam folder")

output("\\t8. Exit: Keluar dari program")

else: // belum login

output("Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu.")

output()

output("\\t1. Login: Masuk ke dalam akun yang sudah terdaftar")

output("\\t2. Register: Membuat akun baru")

output("\\t3. Save: Menyimpan data ke dalam folder")

output("\\t4. Exit: Keluar dari program")

output()

output("Footnote:")

output("\\t1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang
terdaftar")

output("\\t2. Jangan lupa untuk memasukkan input yang valid")

```

#### g. F05 – Monster

```

from src.lcg import *

{Realisasi fungsi dan prosedur}

function attack_user (attackpower: float) → float {Fungsi untuk generate damage dari
attack power}

x ← generate_number (lcg(a=48271, c=0, m=2**31-1, seed=None), [0,2])
tanda ← 0

if x = 0 then
    tanda = -1

else
    tanda = 1

pengali ← generate_number (lcg(a ← 48271, c ← 0, m ← 2**31-1, seed ← None),
[1, 31])
damage ← ((tanda*pengali)/100)*attackpower + attackpower
→ damage

function attack_bot (attackpower: float) → float {Fungsi untuk generate damage dari
attack power}

```

```

x ← generate_number (lcg(a ← 6364136223846793005, c ←
1442695040888963407, m ← 2**64, seed ← None), [0,2])

tanda ← 0

if x = 0 then
    tanda = -1

else
    tanda = 1

pengali ← generate_number (lcg (a ← 6364136223846793005, c ←
1442695040888963407, m ← 2**64, seed ← None), [1, 31])
damage ← ((tanda*pengali)/100)*attackpower + attackpower
→ damage

function defense (defpower: float, enemydamage: float) → float: {Fungsi untuk
menghitung damage yang diterima setelah dikurangi dengan DEF power}

    damage_reduction ← (defpower/100)*enemydamage
    → damage_reduction

function level_buff (level: integer, user_battle_monster: data_monster) → list: {Buff
atribut sesuai level. Digunakan pada saat battle, otomatis mengubah atribut attack,
defense, HP menjadi integer}

    if level > 1 then {Jika level lebih dari 1, atribut dibuff sesuai level, dan diubah
menjadi integer}

        user_battle_monster[2] ← integer (user_battle_monster[2]) + integer (((integer
        (user_battle_monster[2])*(((level-1)*10)/100))) {Buff attack}
        user_battle_monster[3] ← integer (user_battle_monster[3]) + integer (((integer
        (user_battle_monster[3])*(((level-1)*10)/100))) {Buff defense}
        user_battle_monster[4] ← integer (user_battle_monster[4]) + integer (((integer
        (user_battle_monster[4])*(((level-1)*10)/100))) {Buff HP}

    else {Jika level 1, atribut tidak dibuff, hanya diubah menjadi integer saja}

        user_battle_monster[2] ← integer (user_battle_monster[2])
        user_battle_monster[3] ← integer (user_battle_monster[3])
        user_battle_monster[4] ← integer (user_battle_monster[4])

    → user_battle_monster

procedure gambar_monster (input id: string)

if id = '1' then

    output(""""

    :=*:

    -*%%%

```

=%=%=%=%=%=

-+-:;%=%=%=%\*

-=:;+;%=%=%#

-+;:::+;%=%#

.+-;::::\*%+

-+.;::::-#-

+-.::::-+.

+###%=%=%=%=-----:.. .... .+.;::::-+-

:\*%=%=%=%=%\*;-----:----:.. -----:----=-=====+ ..;-=+.. :

-+;%=%=%#-;-----:-+ =... ..;-+;----=-+=+.. ;=;---.

.-=-\*#+-;----:.. -----:----=-\*=\*: .. ;=;....;-=.

;----=+++#=;-----:----:--\*\* .. ;=;.....;-=.

\*-----:----:----=-+=- .. ;=;-.....;-=.

\*-\*\*+;-;-----:+ ==#%\* ;-----\* .. ;=;.....;-=-

;+-%%.-\*;-;-----:-%. -%%+;-----=+ =;.....;-+ =

+--#%=%=%=-----\*%=%=%=%=-----\*-----;-=:

-=;----=-;-+;-;-----=-=====+ =-;----#-----;-=:

.#+;-;-----;-+--+ =\*;-\*-----;-;-----=-,

-\*+\*;-;-----=++=+=;-----\*+;----+\* =;-----;-=====;

.##+;-----;-----\*\*\*\*+;-\*-----=-\*=:

;\*-;-----;-;-----#-----;-+ =-.

=+;-;-----;-;-----#-----=-;-;-----=+ =-.

-+;-;-----;-;-----;-\*-----\* .. ;=;-;-----=--=

\*+-----;-;-----;-----+ .. ;=;-;-----=+ =-

.+-----;-;-----;-;-----;-=-----=+ =-;

+;-----;-;-----;-----#=#. -+-----=+ =-.

\*-----;-=;-----\*#-----\*#-----+ ==-----=+ =-.

\*-----\*;-;-----\*..-----+-----+ .. ;=;-=====+ =-;

=#;-----\* =;-----\*;-----\*-----=; .. ;=;-=====+ =\*.

\*#;-----\* =;-----\*;-----\*-----\* .. ;=;-=====+ =-

-=+;-;-----#-----\* .. ;-----#-----=\*% - =#\*\*+ =++ =-

\*;-#-----#-----\*-----#=-----\*%\*#-----\*%\*#-----\* =



```

====+---:::-----+ == =----=-----**.

.--=**+ +=====+++++++=-----=---();-----+--+
,=====-----= +-----=-:::---+ -:-;-----=
-;-----+ +=====-----+ =:::---+ =:::+ +-----;
-:::---+ + + + +=====+ :::+ =:::---*+ == + *-----+
,=====+-----=====+ == -+-----:::---+ -:+ + + + *-----+
-==-*-----+ + + + == *--+ :::---+ +-----+ + +-----+
=====+ : : # == + -:::---+ + *-----+ +
=:-----=. == + + =-:::---+ -----=.
-=(+ =(+ -: . == )-----= . ===== -===== .
,:::---; ..... .

```

"")

```
elif id = "3" then
```

```
output("")
```

-







output(""")

```

.#  #  @@+.

--+  :@ =@=:* - 

-+=  *:@ =**++- *-:= *.

+= @  *-*-*--= *@--++*+ +*:=- -... 

-:@ .--@ +#***+**++ + +*****@ +@ +# + % = - + @

% @ @ - + = @ # + = + + # + #####*****@ # @ = * @ + @ + * # = : = % #.

. @ @ :**+**@ #*#****#*****+ + * @ + + *****@ % - : # 

. + @ = + **+**@ ***#*#**# + @ ***#**%*+ + *#*#*+ *+ + + + + 

= + * @ # : - % @ ***#****+ + *#***#* = @ + + **+ = - - % #***+**##### 

* = + - @ + + ****+ ***+**@ + + *+ + @ **% = + % @ @ @ + + + = + + #***- :... + 

@ @ @ # + ****#****#* = # @ *# @ @ @ @ @ @ @ @ % @ @ @ @ @ @ *+ #***+ + = % 

....--= + ****+ = - - # % @ @ @ @ @ = - :... - = * + @ @ @ @ % # % * * # @ @ 

% : - = + **+**#*% @ @ @ @ @ * = - :.% @ : - * @ - + + + + + = @ @ = :****+ = - .. 

* @ @ % @ ****+ + *+ - # @ - + + + : @ - : - @ . @ . @ + + + + *+ + + = - - @ @ + ****% @ # = :.. 

.. = *## @ @ % * @ % = + *+ + + * : @ @ @ @ : @ % @ = - - - + *+ + = # % - = + + + + * 

@ @ #*+ + + = # @ + + + + + + : @ . : @ : : # @ ##***+ + + = @ @ + = - - - - - .. : 

.....- + **+ = * @ @ = + ***# # % * = : = + # + @ @ % @ % + - - - = @ @ @ @ @ @ @ @ @ 

@ *+ = - # = @ @ : + ***+ + - : = @ @ = + *+ + % @ # @ @ @ @ @ @ @ @ % + = @ @ @ @ @ @ @ % 

*: - = @ % @ % @ + - + + = * % @ @ @ @ @ @ @ @ @ + + = % # % @ .. @ @ ... + - : + @ @ @ @ @ 

= ... # @ @ @ % @ @ @ @ # - + + @ @ @ * : = % .. @ : - :... - : + @ * - % @ @ # - - @ @ ... 

% ## @ @ @ + @ @ @ : + = * .. @ @ % - : = @ @ @ @ @ @ @ #**+ + * = : @ = : @ . @ @ + 

@ @ @ @ @ : - = % @ @ @ @ @ *+ - + + @ @ = :... - + + + + + + + @ @ @ @ + + % @ @ + = 

.... @ @ @ # :... - = + @ + - : = + + *+ + *+ + *+ + * = - : @ @ * - % @ = : 

. # @ @ : .. @ @ @ = + + + + + + + *# @ + # ##**+ + *****+ + : @ . # % @ @ . = 

@ : @ % @ : @ : = + + + + + + *+ + + + + + + + + = + + + = : @ : @ : .. 

.: - @ @ : + % @ : + + ***+ + + + = - - - + + *% @ @ @ @ *+ : @ @ .... : 

- .. = @ @ + % @ : + + ***+ + % @ @ @ @ @ % *+ + + = - - . @ #* 

.... : % @ : - + % @ #**+ + + + = + + *% @ @ @ @ *+ + + = : @ @ - @ 

% : @ @ : .. = + + + % # + + - :... @ @ + = : 

= + * : @ @ % : .. - + - @ : @ @ @ @ * : @ @ - 

+ # % % : .. @ @ @ @ @ @ @ @ * = - - @ @ % ... # @ @ *.

```

```

+ *#%@@@#::..+@#=+=+::@@%%+ -:
#*%%%%%@@@@%+:::-@@@@@%@-=:
@#%%%@@@%%%@@@@@@@@@@@@:@%@=+:
#@*@@@%%%@@%%%@@=--@++ +:
.::=---====*+==--::.... -...
""")
```

else

output(""")

\_A---A

/ \ A

| O O | |

| .vvvv. | |

/ | | \ |

/ `^`^`^`^` \ | |

. / \ \ | |

/ / | | \\_ /

\ / | | | |

" | \_ |

\_,-,-,-,-,-,-,

\_,,-,-,-,-,-,-,

""")

## h. F06 – Potion

function pilih\_potion(user\_id: int, item\_inventory: list)  $\rightarrow$  list

count  $\leftarrow$  1

pilihan\_potion  $\leftarrow$  []

i traversal [0...len(item\_inventory) - 1]

```

if (user_id == item_inventory[i][0]) then
    if (count == 1) then:
        output("----- POTION -----")
        if (item_inventory[i][2] > 0) then
            if (item_inventory[i][1] == "strength") then
                txt1 ← "Strength"
                txt2 ← "Increases ATK Power"
            elif (item_inventory[i][1] == "resilience") then
                txt1 ← "Resilience"
                txt2 ← "Increases DEF Power"
            else { (item_inventory[i][1] == "healing") }
                txt1 ← "Healing"
                txt2 ← "Restores 25% Health"
        output(count . " . " . txt1 . " Potion (Qty: " . item_inventory[i][2] .
        ") - " . txt2)
        pilihan_potion.append(item_inventory[i].copy())
        pilihan_potion[count - 1].append(i)
        count ← count + 1
    if (count == 1) then
        output("Anda tidak memiliki Potion dalam inventory!")
    else
        output(count . " . Cancel")
        output()
return pilihan_potion

function use_potion(user_id: int, item_inventory: list, status_potion: list,
nama_monster: str) ->(list,list,int):
    output() { Cetak baris kosong }

    while True do

```

```

pilihan_potion ← pilih_potion(user_id, item_inventory)

if (len(pilihan_potion) == 0) then

    return item_inventory, status_potion, 0

while True do

    potion_dipilih ← input("Pilih potion untuk diminum: ")

    if (is_integer(potion_dipilih)) then

        potion_dipilih ← int(potion_dipilih)

        break

    output("Pilihan nomor tidak tersedia!") { Input tidak valid }

    { Ulangi }

    output() { Cetak baris kosong }

    { User memilih untuk membatalkan }

    if (potion_dipilih == len(pilihan_potion) + 1) then

        output()

        return item_inventory, status_potion, 0 { Kembali ke state awal }

    { Gunakan potion }

    elif ((potion_dipilih > 0) and (potion_dipilih <= len(pilihan_potion))) then

        index ← convert_to_integer(pilihan_potion[potion_dipilih - 1][3])

        item_inventory[index][2] ← convert_to_integer(item_inventory[index][2])

        if (item_inventory[index][2] == 0) then

            output("Wah, kamu sedang tidak memiliki ramuan ini, silahkan pilih ramuan lain!") { Ulangi }

        elif (status_potion[pilihan_potion[potion_dipilih - 1][1]]) then

            output("Kamu mencoba memberikan ramuan ini kepada " . nama_monster .

", namun dia menolaknya seolah-olah dia memahami ramuan tersebut sudah tidak bermanfaat lagi.") { Ulangi }

```

```

else

    item_inventory[index][2] ← item_inventory[index][2] - 1

    { Mengurangi qty potion yang dipilih }

    item_inventory[index][2] ← str(item_inventory[index][2])

    if (pilihan_potion[potion_dipilih - 1][1] == "strength") then

        status_potion["strength"] ← true

        code ← 1

        output("Setelah meminum ramuan ini, aura kekuatan terlihat mengelilingi " .

nama_monster . " dan gerakannya menjadi lebih cepat dan mematikan.")

        elif (pilihan_potion[potion_dipilih - 1][1] == "resilience") then

            status_potion["resilience"] ← true

            code ← 2

            output("Setelah meminum ramuan ini, muncul sebuah energi pelindung

di sekitar " . nama_monster . " yang membuatnya terlihat semakin tangguh dan sulit

dilukai.")

        else { pilihan_potion[i][1]== "healing" }

            status_potion["healing"] ← true

            code ← 3

            output("Setelah meminum ramuan ini, luka-luka yang ada di dalam tubuh

" . nama_monster . " sembuh dengan cepat. Dalam sekejap, " . nama_monster . "

terlihat kembali prima dan siap melanjutkan pertempuran.")

            output() { Cetak baris kosong }

            return item_inventory, status_potion, code

        else { Input tidak valid }

            output("Pilihan nomor tidak tersedia!") { Ulangi }

            output() { Cetak baris kosong }

```

### i. F07 – Inventory

```

procedure inventory(user: list, item_inventory: list, monster_inventory: list, monster:
list):
    output()
    output("Selamat datang di Inventory!\nAnda dapat melihat detail monster dan item
yang dimiliki pada Inventory\n")
    while True do
        output("Pilih aksi yang Anda ingin lakukan!")
        output("1.Buka Inventory\n2.Keluar\n")
        aksi ← input("Masukkan angka dari aksi yang ingin Anda lakukan: ")
        if aksi == "1" then
            output()
            user_inventory ← []
            count ← 1
            output(f'===== INVENTORY LIST (User ID: {user[0]})')
            =====")
            output(f"Jumlah O.W.C.A Coin-mu sekarang {user[4]}.")

            i traversal [0...len(monster_inventory) - 1]:
                if str(monster_inventory[i][0]) == str(user[0]) then {Jika user_id cocok}
                    output(f'{count}. Monster \t(Name:
{monster[int(monster_inventory[i][1])][1]}, Lvl. {monster_inventory[i][2]}, HP:
{monster[int(monster_inventory[i][1])][4]}}')

                    detail_item ← monster[int(monster_inventory[i][1])].copy()
                    detail_item[0] ← "Monster"
                    detail_item.append(monster_inventory[i][2])
                    user_inventory.append(detail_item)
                    count += 1

            i traversal [0...len(item_inventory) - 1]:
                if str(item_inventory[i][0]) == str(user[0]) then {Jika user_id cocok}
                    if int(item_inventory[i][2]) != 0 then {Jika potion ada}
                        if item_inventory[i][1] == "strength" then
                            tipe ← "ATK"
                        elif item_inventory[i][1] == "resilience" then

```

```

tipe ← "DEF"
else
    tipe ← "Heal"
output(f"{{count}. Potion \t(Type: {tipe}, Qty: {item_inventory[i][2]})}")
detail_item ← ["Potion", tipe, item_inventory[i][2]]
user_inventory.append(detail_item)
count += 1

output()
output("Ketikkan id untuk menampilkan detail item: ", end="")
while True do
    idx ← input()
    if idx == "" then
        output("Input tidak boleh kosong!")
    elif is_integer(idx) then
        idx ← int(idx)
        if idx >= 1 and idx < count then
            break
        else
            output("Input tidak valid! Masukkan angka yang sesuai dengan jumlah
jenis barang yang Anda punya!")
            output("\\nKetikkan id untuk menampilkan detail item: ", end="")
    output()
    if user_inventory[idx - 1][0] == "Monster" then
        output(f"""Monster
Name      : {user_inventory[idx - 1][1]}
ATK Power : {int(user_inventory[idx - 1][2]) + int((int(user_inventory[idx - 1][2]) *
* (((int(user_inventory[idx - 1][5]) - 1) * 10) / 100)))}
        DEF Power : {int(user_inventory[idx - 1][3]) + int((int(user_inventory[idx - 1][3]) *
* (((int(user_inventory[idx - 1][5]) - 1) * 10) / 100)))}
        HP       : {int(user_inventory[idx - 1][4]) + int((int(user_inventory[idx - 1][4]) *
* (((int(user_inventory[idx - 1][5]) - 1) * 10) / 100)))}
        Level    : {user_inventory[idx - 1][5]}
        """")
    else {user_inventory[idx - 1][0] == "Potion"}
        output(f"""Potion

```

```

Type    : {user_inventory[idx - 1][1]}
Quantity : {user_inventory[idx - 1][2]}
""")
elif aksi == "2" then
    output("Anda keluar dari Inventory!")
    break
else
    output("Aksi tidak valid! Masukkan ulang input!\n")

```

#### j. F08 – Battle

```

from src.monster import *
from src.potion import *
import math
import time

{Realisasi fungsi dan prosedur}

function is_integer (user_input: str) → bool
    for char in str (user_input)
        if (ord(char) < ord('0')) or (ord(char) > ord('9')) then
            → False
        → True

function pilih_monster (user_id: integer, monster_inventory: array of
data_monster_inventory, monsterarray: array of data_monster) → data_monster
    count ← 1 {Nomor list}
    {i adalah user id}
    pilihan_monster ← []
    i traversal [0 .. len(monster_inventory)]
    if string(user_id) = str(monster_inventory[i][0]) then
        monster_id ← integer(monster_inventory[i][1])
        output(count, ":", monsterarray[monster_id][1])
        count ← count + 1
        pilihan_monster.append(monsterarray[monster_id].copy())
    → pilihan_monster

```

```
procedure gambar_monster_user(output: string)
```

```
    output(r"""
        ^---^_
        /     \  /
      | | O   O | / |
      | | .vvvvv.| / /
      / | |   |   /
      / | `^^^^^  /
      | / |   /
      /|   __  |
      \| |   |
      | |   |
      \_\ \_\_\"
```

```
""")
```

```
procedure monster_dipilih_user(level: integer, user_battle_monster: data_monster)
```

```
    output(f"""
RAWRRR, Agent X mengeluarkan monster {user_battle_monster[1]} !!!
```

```
Name    : {user_battle_monster[1]}
```

```
ATK Power : {user_battle_monster[2]}
```

```
DEF Power : {user_battle_monster[3]}
```

```
HP     : {user_battle_monster[4]}
```

```
Level   : {level}
```

```
""")
```

```
procedure output_user_turn(turn_counter: integer, monster_name: string)
```

```
    output(f"""
===== TURN {turn_counter} ({monster_name}) =====
```

1. Attack

2. Use Potion

3. Quit

```

""")  

procedure output_bot_turn(turn_counter: integer, bot_monster_name: string,  

user_monster_name: string): {Untuk menampilkan turn bot (hanya attack)}  

    output(f"""  

===== TURN {turn_counter} ({bot_monster_name}) =====  

SCHWINKKK, {bot_monster_name} menyerang {user_monster_name} !!!  

""")  

    time.sleep(1)  

procedure attack_result(enemy_battle_monster: data_monster, user_battle_monster:  

data_monster, user_damage: integer, damage_reduction: integer,  

damage_after_reduction:integer, base_HP_enemy_monster: integer,  

user_battle_monster_level: integer, enemy_battle_monster_level: integer)  

    tanda ← 1  

    if user_damage < user_battle_monster[2] then  

        tanda ← 0  

    else  

        tanda ← 1  

    if tanda = 0 then  

        tanda ← '-'  

    output(f"""  

Name : {enemy_battle_monster[1]}  

ATK Power : {enemy_battle_monster[2]}  

DEF Power : {enemy_battle_monster[3]}  

HP : {base_HP_enemy_monster}  

Level : {enemy_battle_monster_level}  

Penjelasan: ATT: {user_damage} ({tanda}{math.floor((user_battle_monster[2]-  

user_damage)/user_battle_monster[2]*100)}%), Reduced by: {damage_reduction}  

({enemy_battle_monster[3]}%), ATT Results: {damage_after_reduction}  

""")  

else

```

```

tanda ← '+'
output(f""")

Name    : {enemy_battle_monster[1]}
ATK Power : {enemy_battle_monster[2]}
DEF Power : {enemy_battle_monster[3]}
HP      : {base_HP_enemy_monster}
Level   : {enemy_battle_monster_level}

```

Penjelasan: ATT: {user\_damage} ({tanda}{math.floor((user\_damage-user\_battle\_monster[2])/user\_battle\_monster[2]\*100)}%), Reduced by: {damage\_reduction} ({enemy\_battle\_monster[3]}%), ATT Results: {damage\_after\_reduction} """)

procedure user\_attack\_turn(user\_monster\_name: string, bot\_monster\_name: string)  
output(f""")

SCHWINKKK, {user\_monster\_name} menyerang {bot\_monster\_name} !!!  
""")

time.sleep(1)

procedure gambar\_monster\_musuh(): {Untuk mengeprint gambar monster yang muncul}

output(r""")

```

_ \----^
 /     \  ^
 | O   O |  |
 | .vvvvv. |  |
 / |   | \ |
 / `^^^^^` \ |
 ./ /|       \| |
 / /|       \|__  /
 \ /|       |  |
 ``| _  |
```

```

_,-'-'`-'!_
__.-'          '-._____
""")
```

procedure munculmonster(monsterarray: array of data\_monster, monster\_muncul\_id: integer) {Untuk menampilkan data monster yang muncul}

```

gambar_monster(monsterarray[monster_muncul_id][0])
output(f"""
```

RAWRRR, Monster {monsterarray[monster\_muncul\_id][1]} telah muncul !!!

```

Name      : {monsterarray[monster_muncul_id][1]}
ATK Power : {monsterarray[monster_muncul_id][2]}
DEF Power : {monsterarray[monster_muncul_id][3]}
HP       : {monsterarray[monster_muncul_id][4]}
Level    : 1
```

```

""")
```

function dapet\_duit (user: array of data\_user)-> array of data\_user:

```

OC ← generate_number(lcg(a ← 48271, c ← 0, m ← 2**31-1, seed ← None), [5,
31])
output(f"OC yang diperoleh: {OC}")
user_oc ← int(user[4])
user_oc ← user_oc + OC
user[4] ← string(user_oc)
→ user
```

function pilihmonster (monsterarray: array of data\_monster, monster\_inventory: array of data\_monster\_inventory, user\_id: integer)-> (integer, data\_monster)

```

output(r"===== MONSTER LIST =====")
pilihan_monster ← pilih_monster(user_id, monster_inventory, monsterarray)
repeat {Validasi pilihan monster}
    input(monster_dipilih)
    output()
```

```

if monster_dipilih = "" then
    output("Input tidak boleh kosong!")
else if (is_integer(monster_dipilih)) then
    monster_dipilih ← int(monster_dipilih)
    until ((monster_dipilih>=1)and(monster_dipilih<=len(pilihan_monster)))
    output("Pilihan nomor tidak tersedia!\n")
    → (monster_dipilih, pilihan_monster)

function battle(monsterarray: array of data_monster, monster_inventory: array of
data_monster_inventory, item_inventory: data of item_inventory, user_id: integer,
bot_battle_monster_level: integer, arena: boolean, monster_dipilih:integer,
pilihan_monster: data_monster) → boolean:
    time.sleep(1)
    win ← False {Boolean untuk hasil battle}
    monster_muncul_id ← generate_number(lcg(a ← 48271, c ← 0, m ← 2**31-1,
seed ← None), [1,len(monsterarray)])
    munculmonster(monsterarray, monster_muncul_id)
    if not arena then
        (monster_dipilih, pilihan_monster) ← pilihmonster(monsterarray,
monster_inventory, user_id)
        {Sudah memilih monster yang akan bertarung, memasukkan semua variabel untuk
dipake kalkulasi battle}
        user_battle_monster ← pilihan_monster[monster_dipilih-1].copy() {Memasukkan
data array monster user yang dipilih}
        bot_battle_monster ← monsterarray[monster_muncul_id].copy() {Memasukkan
data array monster bot yang akan melawan user}
        user_battle_monster_level ← 1
        i traversal [1..len(monster_inventory)] {Memasukkan data level monster user}
        if user_id = integer(monster_inventory[i][0]) and
integer(monster_inventory[i][1]) = int(user_battle_monster[0]) then
            user_battle_monster_level ← int(monster_inventory[i][2]) {Level monster
user}

```

user\_battle\_monster  $\leftarrow$  level\_buff(user\_battle\_monster\_level, user\_battle\_monster)  
 {Mengubah atribut sesuai level monster dan menjadikannya integer}

bot\_battle\_monster  $\leftarrow$  level\_buff(bot\_battle\_monster\_level, bot\_battle\_monster)  
 {Mengubah atribut monster musuh menjadi integer (level musuh di-cap di lvl 1)}

{Base atk, base def, HP user yang sudah dibuff level}

base\_atk\_user\_monster  $\leftarrow$  user\_battle\_monster[2]

base\_def\_user\_monster  $\leftarrow$  user\_battle\_monster[3]

base\_HP\_user\_monster  $\leftarrow$  user\_battle\_monster[4]

{Base atk, base def, HP bot yang sudah dibuff level}

base\_atk\_bot\_monster  $\leftarrow$  bot\_battle\_monster[2]

base\_def\_bot\_monster  $\leftarrow$  bot\_battle\_monster[3]

base\_HP\_bot\_monster  $\leftarrow$  bot\_battle\_monster[4]

{Proses battle}

if(arena) then

output(f"===== STAGE {bot\_battle\_monster\_level} =====")

time.sleep(1)

gambar\_monster(user\_battle\_monster[0])

monster\_dipilih\_user(user\_battle\_monster\_level, user\_battle\_monster)

time.sleep(1)

status\_potion  $\leftarrow$  {"strength":False,"resilience":False,"healing":False}

turn\_counter  $\leftarrow$  1

kabur  $\leftarrow$  False

total\_damage\_dealt  $\leftarrow$  0

total\_damage\_taken  $\leftarrow$  0

while base\_HP\_user\_monster > 0 and base\_HP\_bot\_monster > 0 do

{Turn user}

output\_user\_turn(turn\_counter, user\_battle\_monster[1])

```

while (True) do

    user_action ← input("Pilih perintah: ") {Aksi yang dipilih user pada turn}

    if user_action = "" then

        output("Aksi tidak boleh kosong!")

    else if(is_integer(user_action)) then

        user_action ← integer(user_action)

        if user_action = 1 then {Jika user memilih attack}

            user_attack_turn(user_battle_monster[1], bot_battle_monster[1]) {User menyerang bot}

            damage ← attack_user(base_atk_user_monster) {Randomize damage +- 30%}

            damage_reduction ← defense(base_def_bot_monster, damage) {Damage reduction akibat atribut defense}

            damage_after_reduction ← damage - defense(base_def_bot_monster, damage) {Damage setelah dikurangi defense musuh}

            total_damage_dealt ← total_damage_dealt + damage_after_reduction {khusus arena}

            base_HP_bot_monster ← math.floor(base_HP_bot_monster - damage_after_reduction) {HP berkurang karena damage musuh}

            if base_HP_bot_monster <= 0 then

                base_HP_bot_monster ← 0

                attack_result(bot_battle_monster, user_battle_monster, damage, damage_reduction, damage_after_reduction, base_HP_bot_monster, user_battle_monster_level, 1)

                time.sleep(2)

                break

    else if user_action = 2 then {Jika user memilih use potion}

        item_inventory, status_potion, code ←
        use_potion(user_id, item_inventory, status_potion, user_battle_monster[1])

        if(code = 1) then

            base_atk_user_monster ← base_atk_user_monster +
            int(base_atk_user_monster*5/100)

            user_battle_monster[2] ← base_atk_user_monster {untuk di print}

```

```

        break

else if(code=2) then

    base_def_user_monster ← base_def_user_monster +
    int(base_def_user_monster*5/100)

    user_battle_monster[3] ← base_def_user_monster {untuk di print}

    break

else if(code=3) then

    base_HP_user_monster ← base_HP_user_monster +
    integer(base_HP_user_monster*25/100)

    if(base_HP_user_monster>user_battle_monster[4]) then

        base_HP_user_monster ← user_battle_monster[4]

        user_battle_monster[4] ← base_HP_user_monster {untuk di print}

        break

else {code == 0, artinya user memilih Cancel pada potion}

    pass

else if user_action = 3 then {Jika user memilih quit}

    kabur ← True

    output("Anda berhasil kabur dari BATTLE!")

    break

else

    output("Perintah tidak valid, ulangi!")

else

    output("Perintah tidak valid, ulangi!")

if(kabur) then

    break


if base_HP_bot_monster = 0: {Jika saat user menyerang, musuh sudah mati}

    break


{Turn bot}

```

```

output _ bot _ turn(turn _ counter, bot _ battle _ monster[1], user _ battle _ monster[1])

damage ← attack _ bot(base _ atk _ bot _ monster) {Randomize damage +- 30%}

damage _ reduction ← defense(base _ def _ user _ monster, damage) {Damage
reduction akibat atribut defense}

damage _ after _ reduction ← damage - defense(base _ def _ user _ monster, damage)
{Damage setelah dikurangi defense musuh}

total _ damage _ taken ← total _ damage _ taken + damage _ after _ reduction {khusus
arena}

base _ HP _ user _ monster ← math.floor(base _ HP _ user _ monster -
damage _ after _ reduction) {HP berkurang karena damage musuh}

if base _ HP _ user _ monster <= 0 then
    base _ HP _ user _ monster ← 0

    attack _ result(user _ battle _ monster, bot _ battle _ monster, damage,
damage _ reduction, damage _ after _ reduction, base _ HP _ user _ monster, 1,
user _ battle _ monster _ level)

    time.sleep(2)

    turn _ counter ← turn _ counter + 1

{Selesai battle}

if base _ HP _ bot _ monster = 0 then {Kasus menang}
    output(f"Selamat, Anda berhasil mengalahkan monster
{bot _ battle _ monster[1]} !!!")

    win ← True

else if base _ HP _ user _ monster = 0 then {Kasus kalah}
    output(f"Yahhh, Anda dikalahkan monster {bot _ battle _ monster[1]}. Jangan
menyerah, coba lagi !!!")

if(arena) then
    → win, total _ damage _ dealt, total _ damage _ taken

→ win

```

#### k. F09 – Arena

```

from src.battle import *
{Algoritma}

function arena(monsterarray: array of data_monster, monster_inventory: array of
data_monster_inventory, item_inventory: array of data_item_inventory, user_id:
integer, stage: integer) → integer

    {Mulai program arena }

    output("Selamat datang di Arena!!")

    output()

    total_OC ← 0

    total_damage_dealt ← 0

    total_damage_taken ← 0

    damage_dealt ← 0

    damage_taken ← 0

    (monster_dipilih, pilihan_monster) ← pilihmonster(monsterarray,
    monster_inventory, user_id)

    while True

        (win, damage_dealt, damage_taken) ← battle(monsterarray, monster_inventory,
        item_inventory, user_id, stage, True, monster_dipilih, pilihan_monster)

        time.sleep(2)

        total_damage_dealt ← total_damage_dealt + damage_dealt

        total_damage_taken ← total_damage_taken + damage_taken

        OC ← [30, 50, 100, 200, 500]

        if(win) then

            total_OC ← total_OC + OC[stage-1]

            output()

            output(f"STAGE CLEARED! Anda akan mendapatkan {OC[stage-1]} OC
pada sesi ini!")

            output()

        if(stage=5) then

            output("Selamat, Anda berhasil menyelesaikan seluruh stage Arena !!!")

            output()

```

```

break

else

    output("Memulai stage berikutnya...")

    stage ← stage + 1

else

    output(f"GAME OVER! Sesi latihan berakhir pada stage {stage}!")

    ouput()

    break

time.sleep(2)

output(f"""

===== STATS =====

Total hadiah    : {total_OC} OC
Jumlah stage   : {stage}
Damage diberikan : {int(total_damage_dealt)}
Damage diterima  : {int(total_damage_taken)}
""")  

→ total_OC

```

## I. F10 - Shop & Currency

{ realisasi fungsi dan prosedur}

procedure shopmain(data\_monstershop: List of List of String,  
 data\_monster: List of List of String,  
 data\_potionshop: List of List of String,  
 data\_monster\_inventory: List of List of String,  
 data\_potion\_inventory: List of List of String,  
 data\_owca: List of List of String,  
 user\_id: Integer)

while True do

output("SHOP MENU:")  
output("1. Lihat")  
ouput("2. Beli")

```

output("3. Keluar")

choice <- input("Masukkan pilihan anda (1/2/3): ")

if choice = '1' then
    call display(data_monstershop, data_monster, data_potionshop)
else if choice = '2' then
    output("Jumlah O.W.C.A Coin-mu sekarang ", data_owca[user_id][4], ".")
    item_type <- input("Anda ingin membeli Potion/Monster: ")
    if item_type.lower = 'monster' then
        data_monster_inventory, data_monstershop, data_owca <-
        buy_monster(data_monster_inventory, data_monster, data_monstershop, data_owca,
        user_id)
    else if item_type.lower = 'potion' then
        data_potion_inventory, data_potionshop, data_owca <-
        buy_potion(data_potion_inventory, data_potionshop, data_owca, user_id)
    else
        output("Pilihan tidak valid. Silakan masukkan kembali!")
else if choice = '3' then
    output("Anda sudah keluar dari toko. Terima kasih sudah datang!")
    break
else
    output("Pilihan tidak valid. Masukkan kembali!")
-> data_monstershop, data_potionshop, data_monster_inventory,
data_potion_inventory, data_owca

procedure is_integer(user_input: String) -> Boolean
    for char in user_input do
        if ord(char) < ord('0') or ord(char) > ord('9') then
            -> False
        -> True

```

```

procedure table(array: List of List of String)
    max_lengths <- [0] * length(array[0])
    for line in array do
        i traversal [0.. len(line) - 1]
        item_length <- length(toString(line[i]))
        if item_length > max_lengths[i] then
            max_lengths[i] <- item_length
    for line in array do
        i traversal [0.. len(line) - 1]
        item <- line[i]
        num_spaces <- max_lengths[i] - length(toString(item))
        if i < len(line) - 1 then
            output(item, " " * num_spaces, " | ", end="")
        else
            output(item, " " * num_spaces, " ", end="")
    output()

```

```

procedure display(data_monstershop: List of List of String, data_monster: List of List of String, data_potionshop: List of List of String)

    item_type <- input("Anda ingin melihat Potion/Monster: ")

    if item_type.lower = 'monster' then
        table(read_filemonster(data_monstershop, data_monster))
    else if item_type.lower = 'potion' then
        table(read_filepotion(data_potionshop))
    else
        output("Masukkan anda tidak valid. Masukkan lagi!")

```

{function read monster}

function read\_filemonster() → 2D Matrix of Strings

```

data_monsternew <- [{"id", "type", "atk_power", "def_power", "hp", "stok",
"harga"}]

i traversal [1..len(data_monstershop) - 1)

j traversal [1..len(data_monster) - 1)

if data_monster[j][0] = data_monstershop[i][0] then

    append(data_monsternew, [data_monster[j][0], data_monster[j][1],
data_monster[j][2], data_monster[j][3], data_monster[j][4], data_monstershop[i][1],
data_monstershop[i][2]])

-> data_monsternew

```

function has\_monsterinven(user\_id: String, id: String, data\_inventory: List of List of String) -> Boolean

```

for m in data_inventory do

if m[0] = user_id and m[1] = id then

-> True

-> False

```

{function read potion}

```

function read_filepotion() → 2D Matrix of Strings

data_potion <- [{"id", data_potionshop[0][0], data_potionshop[0][1],
data_potionshop[0][2]}]

i traversal [1..len(data_potionshop) - 1)

append[data_potion, [toString(i), data_potionshop[i][0], data_potionshop[i][1],
data_potionshop[i][2]])

-> data_potion

```

{procedure buy monster}

```

procedure buy_monster(data_monster_inventory: List of List of String,
data_monster: List of List of String, data_monstershop: List of List of String,
data_owca: List of List of String, user_id: Integer)

```

```

table(read_filemonster(data_monstershop, data_monster))

item_id <- input("Masukkan ID monster: ")

```

```

while True do

    if is_integer(item_id) then

        if has_monsterinven(str(user_id), str(item_id), str(data_monster_inventory))
then

            output("Anda sudah memiliki monster ini, masukkan ID lain!")

            -> data_monster_inventory, data_monstershop, data_owca

        else

            break

        item_id <- input("Masukkan ID monster: ")

        item_id <- int(item_id)

    i traversal[1..len(data_monstershop) -1]

    id < int(data_monstershop[i][0])

    type <- data_monster[i][1]

    stok <-int(data_monstershop[i][1])

    price <- int(data_monstershop[i][2])

    owca <- int(data_owca[user_id][4])

    if item_id = id then

        if stok > 0 then

            if owca >= price then

                output("Kamu berhasil membeli", type, ". Item sudah masuk ke
inventory-mu!")

                output("Sisa O.W.C.A. Coin-mu", owca - price, ".")  

                owca <- owca - price  

                data_owca[user_id][4] <- str(owca)  

                stok <- stok - 1  

                data_monstershop[i][1] <- str(stok)  

                append(data_monster_inventory, [toString(user_id), toString(id),
toString(1)])
  

        else

```

```

output("O.W.C.A Coin-mu kurang.")

else

    output("Stok monster tidak mencukupi.")

-> data_monster_inventory, data_monstershop, data_owca

output("Masukkan anda tidak valid.")

-> data_monster_inventory, data_monstershop, data_owca

```

```

{procedure buy potion}

procedure buy_potion(data_potion_inventory: List of List of String, data_potionshop:
List of List of String, data_owca: List of List of String, user_id: Integer)

table(read_filepotion(data_potionshop))

item_id <- input("Masukkan ID potion: ")

while True do

    if int(item_id) then

        break

    item_id <- input("Masukkan ID potion: ")

    item_id <- int(item_id)

quantity <- input("Masukkan kuantitas: ")

while True do

    if int(quantity) then

        break

    quantity <- input("Masukkan kuantitas: ")

    quantity <- int(quantity)

j traversal[1..len(data_potionshop) - 1]

id_1 <- j

type_1 <- data_potionshop[j][0]

stok_1 <- int(data_potionshop[j][1])

```

```

harga_1 <- int(data_potionshop[j][2])

if item_id = id_1 then
    if stok_1 = 0 then
        output("Stok potion tidak mencukupi.")
    else if quantity > stok_1 then
        output("Stok potion tidak mencukupi.")
    else
        owca <- int(data_owca[user_id][4])
        jumlah <- quantity * harga_1
        if owca >= jumlah then
            output("Kamu berhasil membeli", quantity, type_1, "Potion. Item sudah
            masuk ke inventory-mu!")
            output("Sisa O.W.C.A. Coin-mu", owca - jumlah, ".")
            owca <- owca - jumlah
            data_owca[user_id][4] <- str(owca)
            stok_1 <- stok_1 - quantity
            data_potionshop[j][1] <- str(stok_1)
            k traversal[1..len(data_potion_inventory) - 1]
            if data_potion_inventory[k][0] = user_id and
            data_potion_inventory[k][1] == type_1 then
                stok <- int(data_potion_inventory[k][1])
                stok <- stok + quantity
                data_potion_inventory[k][1] <- str(stok)
                -> data_potion_inventory, data_potionshop, data_owca
                data_potion_inventory.append([toString(user_id), type_1,
                toString(quantity)])
            else
                output("O.W.C.A Coin-mu kurang.")
                -> data_potion_inventory, data_potionshop, data_owca
                output("Masukkan anda tidak valid.")

```

-> data\_potion\_inventory, data\_potionshop, data\_owca

### m. F11 – Laboratory

function laboratory(array\_user: list, array\_monster\_inventory: list, array\_monster: list, user\_id: int) → (list, list)

KAMUS LOKAL

monster\_amt, coin\_amt, price, monster\_id, upgrade\_id, level: integer

confirm: string

idx: array of integer

ALGORITMA

output('Kamu telah memasuki area lab monster!')

output('----- MONSTERS OWNED -----')

monster\_amt <- 0

idx <- []

i traversal [0...(len(array\_monster\_inventory) - 1)]

if (str(array\_monster\_inventory[i][0]) = str(user\_id)) then

    idx.append(i)

    monster\_id <- int(array\_monster\_inventory[i][1])

output(f'{monster\_amt + 1}. {array\_monster[monster\_id][1]} (Level : {array\_monster\_inventory[i][2]})")

    monster\_amt <- monster\_amt + 1

output('----- UPGRADE PRICE LIST -----')

output('1. Level 1 -> Level 2: 250 OC')

output('2. Level 2 -> Level 3: 500 OC')

output('3. Level 3 -> Level 4: 800 OC')

output('4. Level 4 -> Level 5: 1250 OC')

output()

coin\_amt <- int(array\_user[user\_id][4])

```

output(fJumlah koin yang dimiliki: {coin_amt}')

output('Pilih monster yang ingin diupgrade: ')
input(upgrade_id)
while not is_integer(upgrade_id) or not (1 <= int(upgrade_id) <= monster_amt) do
    output('Monster yang ingin diupgrade tidak ditemukan.')
    output('Pilih monster yang ingin diupgrade: ')
    input(upgrade_id)

upgrade_id <- int(upgrade_id)
level <- int(array_monster_inventory[idx[upgrade_id - 1]][2])
if level = 5 then
    output('Monster yang dipilih sudah memiliki level maksimum.')
else
    depend on (level)
        level = 1: price <- 250
        level = 2: price <- 500
        level = 3: price <- 800
else { level = 4 }
        price <- 1250
output(f {array_monster[int(array_monster_inventory[idx[upgrade_id - 1]][1])]
[1]} akan diupdate ke level {level + 1}.')
output('Harga upgrade ini adalah', str(price) + '.')
output('Lanjutkan upgrade (Y/N): ')
input(confirm)
while confirm not in ['Y', 'y', 'N', 'n'] do
    output('Input tidak valid!')
    output('Lanjutkan upgrade (Y/N): ')
    input(confirm)
if confirm in ['Y', 'y'] then

```

```

if price > coin_amt then
    output('Upgrade gagal! Anda tidak memiliki cukup koin.')
else
    output('Upgrade berhasil!')
    level <- level + 1
    array_monster_inventory[idx[upgrade_id - 1]][2] <- level
    coin_amt <- coin_amt - price
    array_user[user_id][4] <- coin_amt
    output(f'{array_monster[int(array_monster_inventory[idx[upgrade_id - 1]]]
[1])][1]} sekarang ada di level {level}.')
else
    output('Upgrade dibatalkan.')
→ (array_monster_inventory, array_user)

```

function is\_integer(user\_input: str) → bool

KAMUS LOKAL

-

ALGORITMA

```

for char in user_input do
    if (ord(char) <= ord('0')) or (ord(char) >= ord('9')) then
        → False
    else
        → True

```

## n. F12 - Shop Management

{ realisasi fungsi dan prosedur}

procedure shop\_management()

```

while True do
    output("Shop Management System (Admin)")
    output("1. Lihat")
    output("2. Tambah")

```

```

        output("3. Update")
output("4. Hapus")
output("5. Keluar")
choice <- input("Masukkan pilihan anda: ")

if choice = '1' then
    display(data_monstershop, data_monster, data_potionshop)
else if choice = '2' then
    data_monstershop, data_potionshop = add(data_monstershop, data_monster,
data_potionshop)
else if choice = '3' then
    data_monstershop, data_potionshop = update(data_monstershop,
data_monster, data_potionshop)
else if choice = '4' then
    data_monstershop, data_potionshop = delete(data_monstershop, data_monster,
data_potionshop)
else if choice = '5' then
    output("Anda sudah keluar dari toko. Terima kasih!")
    output("Terima kasih! Progress terakhir telah disimpan.")
    break
else
    output("Pilihan tidak valid, masukkan angka kembali")

{prosedur menambahkan monster atau potion}
-> data_monstershop, data_potionshop

```

```

procedure is_integer(user_input: String) -> Boolean
    for char in user_input do
        if ord(char) < ord('0') or ord(char) > ord('9') then
            -> False
        -> True

```

```

procedure table(array: List of List of String)
    max_lengths <- [0] * length(array[0])
    for line in array do
        i traversal [0.. len(line) - 1]
        item_length <- length(toString(line[i]))
        if item_length > max_lengths[i] then
            max_lengths[i] <- item_length
    for line in array do
        i traversal [0.. len(line) - 1]
        item <- line[i]
        num_spaces <- max_lengths[i] - length(toString(item))
        if i < len(line) - 1 then
            output(item, " " * num_spaces, " | ", end="")
        else
            output(item, " " * num_spaces, " ", end="")
    output()

```

```

procedure display(data_monstershop: List of List of String, data_monster: List of List of String, data_potionshop: List of List of String)

item_type <- input("Anda ingin melihat Potion/Monster: ")

if item_type.lower = 'monster' then
    table(read_filemonster(data_monstershop, data_monster))

else if item_type.lower = 'potion' then
    table(read_filepotion(data_potionshop))

else
    output("Masukkan anda tidak valid. Masukkan lagi!")

```

{function read monster}

function read\_filemonster() → 2D Matrix of Strings

```

data_monsternew <- [{"id", "type", "atk_power", "def_power", "hp", "stok",
"harga"}]

i traversal [1..len(data_monstershop) - 1)

j traversal [1..len(data_monster) - 1)

if data_monster[j][0] = data_monstershop[i][0] then

    append(data_monsternew, [data_monster[j][0], data_monster[j][1],
data_monster[j][2], data_monster[j][3], data_monster[j][4], data_monstershop[i][1],
data_monstershop[i][2]])

-> data_monsternew

```

{function read potion}

function read\_filepotion() → 2D Matrix of Strings

```

data_potion <- [{"id", data_potionshop[0][0], data_potionshop[0][1],
data_potionshop[0][2]}]

i traversal [1..len(data_potionshop) - 1)

append[data_potion, [toString(i), data_potionshop[i][0], data_potionshop[i][1],
data_potionshop[i][2])]

-> data_potion

```

function id\_shop(data: list)-> list:

```

id = []

i traversal [1...len(data)]

append[id, data[i][0])

```

-> id

procedure add(data\_monstershop: list, data\_monster: list, data\_potionshop: list)

while True do

```

add_things <- input("Mau menambahkan apa? (monster/potion): ").lower()

```

if add\_things == 'monster' then

```

id_monster <- id_shop('data/monster_shop.csv')

data_monsterid <- []
id_add <- []

i traversal [0..(data_monster)-1]
  if data_monster[i][0] not in id_monster then
    append(data_monsterid, data_monster[i].copy())
    if data_monster[i][0] ≠ "id" then
      append(id_add, data_monster[i][0])

if len(data_monsterid) = 1 then
  output("Tidak ada monster yang dapat ditambahkan!")
  break

table(data_monsterid)
monster_shop <- []

while True do
  id <- input("Masukkan ID monster: ")
  if id not in id_add then
    output("ID salah! Masukkan lagi")
    continue
  else
    append(monster_shop, id)
  break

while True do
  stok <- input("Masukkan stok awal: ")
  if int(stok) < 0 then

```

```

output("Stok awal harus lebih besar dari nol!")

    continue

else

    append(monster_shop, stok)

    break

while True do

    harga <- input("Masukkan harga: ")

    if int(harga) < 0 then

        output("Harga harus lebih besar dari nol!")

        continue

    else

        append(monster_shop, harga)

        break

append(data_monstershop, monster_shop)

output("Monster baru berhasil ditambahkan ke dalam toko!")

break

else if add_things = "potion" then

    potion_shop <- []

    i traversal [1..len(data_potionshop)-1]

    append(potion_shop, data_potionshop[i][0])

data_potionid <- []

append(data_potionid, ["ID", "TYPE"])

if "strength" not in potion_shop then

    append(data_potionid, [str(len(data_potionid)), "strength"])

```

```

if "resilience" not in potion_shop then
    append(data_potionid, [str(len(data_potionid)), "resilience"])

if "healing" not in potion_shop then
    append(data_potionid, [str(lendata_potionid)), "healing"])

if len(data_potionid) = 1 then
    output("Tidak ada potion yang dapat ditambahkan!")
    break

table(data_potionid)
id_tambah <- []
i traversal [1..(data_potionid)-1]
append(id_tambah, data_potionid[i][0])
potion_baru <- []

while True do
    id <- input("Masukkan ID potion: ")
    if id not in id_tambah then
        output("ID salah! Masukkan ID kembali")
        continue
    else
        append(potion_baru, data_potionid[int(id)][1])
    break

while True do
    stok <- input("Masukkan stok awal: ")
    if (is_int(stok)) then
        if int(stok) < 0 then
            output("Stok tidak bisa lebih kecil dari nol!")
            continue

```

```

else

    append(potion_baru, stok)

    break

while True do

    harga <- input("Masukkan harga: ")

    if (is_int(harga)) then

        if int(harga) < 0 then

            output("Harga tidak bisa lebih kecil dari nol!")

            continue

        else

            append(potion_baru, harga)

            break

    append(data_potionid,potion_baru)

    output("Potion baru berhasil ditambahkan ke dalam toko!")

    break

else

    output("Masukkan anda tidak valid. Masukkan lagi!")

-> data_monstershop, data_potionshop

{prosedur memperbarui monster atau potion}

procedure update(data_monstershop: list, data_monster: list, data_potionshop: list)

    while True do

        update <- input("Komponen apa yang ingin diperbarui? (monster/potion):").lower()

        if update = "monster" then

            id_monster <- id_shop("data/monster_shop.csv")

```

```

while True do
    id <- input("Masukkan ID monster: ")
    if id not in id_monster then
        output("ID salah! Masukkan ID kembali")
        continue
    else
        break

while True do
    harga <- input("Masukkan harga baru: ")
    if int(harga) < 0 then
        output("Harga tidak bisa lebih kecil dari nol!")
        continue
    else
        break

while True do
    stok <- input("Masukkan stok baru: ")
    if int(stok) < 0 then
        output("Stok tidak bisa lebih kecil dari nol!")
        continue
    else
        break

data_monstershop <- csv_to_array("data/monster_shop.csv")
i traversal [0..(data_monstershop)-1]
    if data_monstershop[i][0] = id then
        data_monstershop[i][1] = stok
        data_monstershop[i][2] = harga

```

break

```
else if update = "potion" then
    csv_to_array(read_filepotion())
    data_potionshop <- csv_to_array("data/item_shop.csv")
    id_potion <- [str(i) for i in range(1, length(data_potionshop))]
```

while True do

```
    id <- input("Masukkan ID potion: ")
    if id not in id_potion then
        output("ID salah! Masukkan ID kembali")
        continue
    else
        break
```

while True do

```
    harga <- input("Masukkan harga baru: ")
    if int(harga) < 0 then
        output("Harga tidak bisa lebih kecil dari nol!")
        continue
    else
        break
```

while True do

```
    stok <- input("Masukkan stok baru: ")
    if int(stok) < 0 then
        output("Stok tidak bisa lebih kecil dari nol!")
        continue
    else
        break
```

```

i traversal [0..(data_potionshop)-1]
    if str(i) = id then
        data_potionshop[i][1] = stok
        data_potionshop[i][2] = harga
        break
    else
        output("Masukkan anda tidak valid. Masukkan lagi!")
    -> data_monstershop, data_potionshop

{prosedur menghapus monster atau potion}
procedure delete(data_monstershop: list, data_monster: list, data_potionshop: list)
    while True do
        delete <- input("Komponen apa yang ingin dihapus? (monster/potion): ").lower()

        if delete = "monster" then
            csv_to_array(read_filemonster())
            id_monster <- id_shop("data/monster_shop.csv")

            while True do
                id_delete <- input("Masukkan id monster: ")
                if id_delete not in id_monster then
                    output("ID salah! Masukkan ID kembali")
                    continue
                else
                    break

            while True do
                check <- input("Apakah anda yakin ingin menghapus monster dari toko
(iya/tidak): ").lower()

```

```

if check = 'iya' then

    data_monstershop <- csv_to_array("data/monster_shop.csv")

    data_update <- []

    i traversal [0..(data_monstershop)-1]

        if data_monstershop[i][0] != id then

            append(data_update, data_monstershop[i])

        break

else if check = 'tidak' then

    break

else

    output("Masukkan salah!")

else if delete = 'potion' then

    csv_to_array(read_filepotion())

    data_potionshop <- csv_to_array("data/item_shop.csv")

    id_delete <- [str(i) for i in range(1, length(data_potionshop))]

while True do

    id <- input("Masukkan ID potion: ")

    if id not in id_delete then

        output("ID salah! Masukkan ID kembali")

        continue

    else

        break

while True do

    check <- input("Apakah anda yakin ingin menghapus potion dari toko
(iya/tidak): ").lower()

    if check = 'iya' then

        data_update <- []

```

```

    i traversal [0..(data_potionshop)-1]
    if str(i)≠ id then
        append(data_update, data_potionshop[i])
    break
else if check = 'tidak' then
    break
else
    output("Masukkan salah!")
break
else
    output("Masukkan anda tidak valid. Masukkan lagi!")
-> data_monstershop, data_potionshop

```

## o. F13 - Monster Management

function monster\_management(array\_monster: list) → list

KAMUS LOKAL

i: integer

new\_monster: array of string

unique\_type: boolean

action, monster\_type, monster\_def\_power, monster\_atk\_power, monster\_hp, confirm:  
string

ALGORITMA

output('Selamat datang di database monster!')

output('1. Tampilkan semua monster')

output('2. Tambahkan monster baru')

output(>>> Pilih aksi yang ingin dilakukan: ')

input(action)

while (action /= '1') and (action /= '2') do

output(>>> Pilih aksi yang ingin dilakukan: ')

input(action)

```

if (action == '1') then
    output(f'{{ID}<3} | {{Type}<15} | {{ATK Power}<10} | {{DEF Power}<10} |
    {{HP}<5}')
    i traversal [1...(len(array_monster) - 1)]
    output(f'{array_monster[i][0]} | {array_monster[i][1]} | 
    {array_monster[i][2]} | {array_monster[i][3]} | {array_monster[i]
    [4]}')
else { action == 2 }
    output('Saatnya membuat monster baru!')
    output('>>> Masukkan tipe atau nama: ')
    input(monster_type)
    unique_type <- True
    i <- 0
    while (i < len(array_monster)) and (unique_type) do
        if (array_monster [i] = monster_type) then
            unique_type <- False
        i <- i + 1
    while not(unique_type) then
        output('Input tidak valid!')
        output('>>> Masukkan tipe atau nama: ')
        input(monster_type)

    output('>>> Masukkan ATK Power: ')
    input(monster_atk_power)
    while not(is_integer(monster_atk_power)):
        output('Input tidak valid!')
        output('>>> Masukkan ATK Power: ')
        input(monster_atk_power)

```

```

output('>>> Masukkan DEF Power (0-50): ')
input(monster_def_power)
while not(is_integer(monster_def_power)) or (is_integer (monster_def_power)
and (int(monster_def_power) < 0 or (int(monster_def_power) > 50)):
    output('Input tidak valid!')
    output('>>> Masukkan DEF Power (0-50): ')
    input(monster_def_power)

output('>>> Masukkan HP: ')
input(monster_hp)
while not(is_integer(monster_hp)):
    output('Input tidak valid!')
    output('>>> Masukkan HP: ')
    input(monster_hp)
output()

output('Monser baru berhasil dibuat!')
output('Type:', monster_type)
output('ATK Power:', monster_atk_power)
output('DEF Power:', monster_def_power)
output('HP:', monster_hp)

output('>>> Tambahkan monster ke database (Y/N): ')
input(confirm)
while (confirm /= 'Y') and (confirm /= 'y') and (confirm /= 'N') and (confirm /=
'n') do
    output('Input tidak valid!')
    output('>>> Tambahkan monster ke database (Y/N): ')
    input(confirm)
if (confirm = 'Y') or (confirm = 'y') then

```

```

{ add new monster details }

new_monster <- [(int(array_monster [len(array_monster) - 1][0]) + 1),
                  monster_type, monster_atk_power, monster_def_power,
                  monster_hp]

array_monster.append(new_monster)

output('Monster berhasil ditambahkan ke database!')

else { (confirm = 'N') or (confirm = 'n') }

output('Monster gagal ditambahkan ke database!'

→ array_monster

```

function is\_integer(user\_input: string) → boolean

KAMUS LOKAL

-

ALGORITMA

```

for char in user_input do

  if (ord(char) <= ord('0')) or (ord(char) >= ord('9')) then

    → False

  → True

```

#### p. F14 – Load

```

import os
import sys
import argparse

```

parser <- argparse.ArgumentParser(description="load data dari folder yang dipilih")

parser.add\_argument("folder", nargs='?', help="nama folder yang berisi data yang ingin di load")

args <- parser.parse\_args()

if not args.folder then

```

output("Tidak ada nama folder yang diberikan!")

output("Usage : python main.py <nama_folder>")

sys.exit()

else

    folder_path <- "./data/" + args.folder

    check_folder <- os.path.exists(folder_path)

    if check_folder = True then

        output("Loading...")

        from load import csv_to_array

        array_user <- csv_to_array(folder_path + "/user.csv")

        array_monster <- csv_to_array(folder_path + "/monster.csv")

        array_monster_inventory <-
            csv_to_array(folder_path + "/monster_inventory.csv")

        array_monster_shop <- csv_to_array(folder_path + "/monster_shop.csv")

        array_item_inventory <- csv_to_array(folder_path + "/item_inventory.csv")

        array_item_shop <- csv_to_array(folder_path + "/item_shop.csv")

        output("Selamat datang di program OWCA!")

    else if check_folder = False then

        output('Folder "' + args.folder + '" tidak ditemukan')

```

**q. F15 – Save**

```

import os

import time

procedure save (array_user: list, array_monster: list, array_monster_inventory: list,
array_monster_shop: list, array_item_inventory: list, array_item_shop: list)

    input(folder)

    while True do

        list_folder ← []

        i traversal [0..<len(list_folder)] 

            list_folder.append(i)

        list_ascii ← [0 i traversal [0..<len(folder)]]
```

```

i traversal [0..<len(folder)]  

    list_ascii[i] ← ord(folder[i])  

i traversal [list_ascii]  

    if i = 34 or i = 42 or i = 47 or i = 58 or i = 60 or i = 62 or i = 63  

    or i = 92 or i = 124 then  

        Output("Nama folder tidak boleh mengandung karakter  

        sebagai berikut: \<>|#)  

Input(folder)  

if folder = "" then  

    output("Nama folder tidak boleh kosong")  

input(folder)  

else  

    Stop  

output("saving...")  

parent check <- os.path.exists("./data")  

if parent_check = False then  

    Os.makedirs("./data")  

    output("membuat folder data")  

folder_check <- os.path.exists("./data" + folder)  

if folder_check = False then  

    Os.makedirs("./data" + folder)  

    output("membuat folder data")  

write_csv(f"./data/{folder}/user.csv", array_user)  

write_csv(f"./data/{folder}/monster.csv", array_monster)  

write_csv(f"./data/{folder}/item_inventory.csv", array_item_inventory)  

write_csv(f"./data/{folder}/monster_inventory.csv", array_monster_inventory)  

write_csv(f"./data/{folder}/item_shop.csv", array_item_shop)  

write_csv(f"./data/{folder}/monster_shop.csv", array_monster_shop)  

output("Berhasil menyimpan data di folder data")

```

Procedure write\_csv(path: str, data: list)

with open(path, "w") as file:

```

    i traversal [data]
        j traversal [0..<len(i)]
            if j = len(i)-1:
                file.write(i[j])
            elif j < len(i)-1:
                file.write(i[j] + ',')
            file.write('\n')

```

#### r. F16 – Exit

function exit() → boolean

input(reassure)

while True do

```

        if reassure = “Y” then
            → True
        else if reassure = “N” then
            → False
        else {semua input selain “Y” dan “N”}
            output("Input tidak valid! Masukkan ulang input!")
            input(reassure)

```

## VIII. Hasil Pengujian Program

- a. F00 - Random Number Generator
- b. F01 – Register

Register berhasil

```
>>> register

Masukkan username: pengentidur
Masukkan password: kapanbobo123

Silahkan pilih salah satu monster sebagai monster awalmu.
1. Pikachu
2. Zuko
3. Chacha

Masukkan angka dari monster yang dipilih.
Monster pilihanmu: 1

Selamat datang Agent pengentidur. Mari kita mengalahkan Dr. Asep Spakbor dengan Pikachu!
```

Gambar 8. 1 Register berhasil

Username tidak valid

```
>>> register

Masukkan username: !@^!@8@!
Masukkan password: jasdjasdkj
Username hanya boleh berisi alfabet, angka, underscore, dan strip.

Masukkan username: okeinibeneran
Masukkan password: iasdaksdk

Silahkan pilih salah satu monster sebagai monster awalmu.
1. Pikachu
2. Zuko
3. Chacha

Masukkan angka dari monster yang dipilih.
Monster pilihanmu: 3

Selamat datang Agent okeinibeneran. Mari kita mengalahkan Dr. Asep Spakbor dengan Chacha!
```

Gambar 8. 2 Username tidak valid

Username sudah terdaftar

```
>>> register
Selamat datang di register!
Pilih aksi dibawah ini:
1. Register
2. Keluar

Masukkan angka dari aksi yang ingin kamu lakukan: 1
Masukkan username: Master_Sepuh
Masukkan password: sepuhsepuhsepuh

Username sudah terpakai, silahkan gunakan username lain.

Pilih aksi dibawah ini:
1. Register
2. Keluar

Masukkan angka dari aksi yang ingin kamu lakukan: 1
Masukkan username: Master_Sepuhh
Masukkan password: sepuhhh

Silahkan pilih salah satu monster sebagai monster awalmu.
1. Pikachow
2. Zuko
3. Chacha

Masukkan angka dari monster yang dipilih.
Monster pilihanmu: 1

Selamat datang Agent Master_Sepuhh. Mari kita mengalahkan Dr. Asep Spakbor dengan Pikachow!
```

*Gambar 8. 3 Username sudah terdaftar*

Monster awal yang dipilih tidak ditemukan

```
>>> register
Masukkan username: wiwiwiwi
Masukkan password: 6789998212

Silahkan pilih salah satu monster sebagai monster awalmu.
1. Pikachow
2. Zuko
3. Chacha

Masukkan angka dari monster yang dipilih.
Monster pilihanmu: 4
input tidak valid. Coba pilih ulang.

Monster pilihanmu: 2

Selamat datang Agent wiwiwiwi. Mari kita mengalahkan Dr. Asep Spakbor dengan Zuko!
```

*Gambar 8. 4 Monster awal yang dipilih tidak ditemukan*

### c. F02 – Login

Pengguna belum login dan semua input valid

```

Selamat datang Agent! Silahkan melakukan login terlebih dahulu!
Pilih aksi dibawah ini:
1. Login
2. Keluar

Masukkan angka dari aksi yang ingin kamu lakukan: 1
Masukkan Username: A_C_E
Masukkan Password: d0natbakar

Login berhasil!
Selamat datang agent A_C_E!

Masukkan perintah!
Ketik "HELP" untuk melihat menu

```

Gambar 8. 5 Kondisi berhasil login dengan input valid

#### Pengguna dalam keadaan sudah login

```

>>> login
Login gagal!
Kamu sudah login dengan akun A_C_E, lakukan logout terlebih dahulu sebelum melakukan login kembali.

```

Gambar 8. 6 Memanggil login dalam keadaan sudah login

#### Kasus input tidak valid (kosong)

```

Selamat datang Agent! Silahkan melakukan login terlebih dahulu!
Pilih aksi dibawah ini:
1. Login
2. Keluar

Masukkan angka dari aksi yang ingin kamu lakukan:
Input tidak valid! Silahkan melakukan input ulang.

```

Gambar 8. 7 Memasukkan input yang tidak valid

#### Username tidak terdaftar

```

Masukkan Username: hai
Masukkan Password: ululu

Username kamu belum terdaftar!
Lakukan registrasi terlebih dahulu sebelum melakukan login atau input username dengan benar.

```

Gambar 8. 88 Username tidak terdaftar

#### Username terdaftar namun password salah

```

Masukkan angka dari aksi yang ingin kamu lakukan: 1
Masukkan Username: A_C_E
Masukkan Password: hehehe

Password kamu salah!

Pilih aksi dibawah ini:
1. Login
2. Keluar

Masukkan angka dari aksi yang ingin kamu lakukan:

```

Gambar 8. 9 Username terdaftar dan password salah

Pengguna memilih aksi keluar

```
Pilih aksi dibawah ini:  
1. Login  
2. Keluar
```

```
Masukkan angka dari aksi yang ingin kamu lakukan: 2  
Anda keluar dari aksi "LOGIN"!
```

Masukkan perintah!

```
Ketik "HELP" untuk melihat menu  
>>> [ ]
```

Gambar 8. 9 Memilih aksi keluar (tidak jadi login)

#### d. F03 – Logout

Pengguna logout dalam keadaan sudah login

```
>>> logout  
Apakah anda ingin keluar dari permainan? (Y/N): y  
User berhasil melakukan logout!
```

Gambar 8. 1101 Berhasil logout

Input tidak valid (kosong)

```
Masukkan perintah!  
Ketik "HELP" untuk melihat menu  
>>> logout  
Apakah anda ingin keluar dari permainan? (Y/N):  
Input tidak valid. Masukkan antara Y/N.  
Apakah anda ingin keluar dari permainan? (Y/N): [ ]
```

Gambar 8. 12 Input tidak valid11

Pengguna tidak jadi logout

```
Apakah anda ingin keluar dari permainan? (Y/N): n  
Logout dibatalkan! Silahkan melanjutkan misi kembali.
```

```
Masukkan perintah!  
Ketik "HELP" untuk melihat menu  
>>> [ ]
```

Gambar 8. 13 Tidak jadi logout12

Pengguna memanggil command logout dalam keadaan belum login

```
Masukkan perintah!
Ketik "HELP" untuk melihat menu
>>> logout
User belum login!
```

Gambar 8. 14 Keadaan pengguna belum login13

#### e. F04 - Menu & Help

Pemanggilan help berhasil

```
Masukkan perintah!
Ketik "HELP" untuk melihat menu
>>> 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
3. Save: Menyimpan data ke dalam folder
4. Exit: Keluar dari program

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

Gambar 8. 114 Pemanggilan help berhasil

Pemanggilan help berhasil

```
Masukkan perintah!
Ketik "HELP" untuk melihat menu
>>> Help
=====

Halo Agent Kenny_agen_rahasia! 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: Melihat detail monster, potion, dan OC yang dimiliki oleh Agent
3. Battle: Bertarung lv1 melawan monster
4. Arena: Melatih monster yang dimiliki oleh Agent
5. Shop: Membeli item berupa monster atau potion
6. Laboratory: Meningkatkan level monster yang dimiliki oleh Agent
7. Save: Menyimpan data ke dalam folder
8. Exit: Keluar dari program

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

Gambar 8. 115 Pemanggilan help berhasil

Pemanggilan command dalam menu berhasil

```
Masukkan perintah!
Ketik "HELP" untuk melihat menu
>>> LOGIN
Selamat datang Agent! Silahkan melakukan login terlebih dahulu!
Pilih aksi dibawah ini:
1. Login
2. Keluar

Masukkan angka dari aksi yang ingin kamu lakukan: 2
```

Gambar 8. 116 Pemanggilan command dalam menu berhasil

Pemanggilan command dalam menu berhasil

```
Masukkan perintah!
Ketik "HELP" untuk melihat menu
>>> register

Masukkan username: Kenny_agen_rahasia
Masukkan password: kribogeming55
```

Gambar 8. 117 Pemanggilan command dalam menu berhasil 2

#### Pemanggilan command dalam menu berhasil

```
Masukkan perintah!
Ketik "HELP" untuk melihat menu
>>> LABoratory
Kamu telah memasuki area lab monster!

Pilih aksi yang ingin Anda lakukan!
1. Upgrade monster
2. Keluar

Ketik angka dari aksi yang ingin Anda lakukan: []
```

Gambar 8. 118 Pemanggilan command dalam menu berhasil 3

#### Pemanggilan command dalam menu tidak valid

```
Masukkan perintah!
Ketik "HELP" untuk melihat menu
>>> UPGRADE
Perintah tidak valid!
```

Gambar 8. 20 Pemanggilan command dalam menu tidak valid

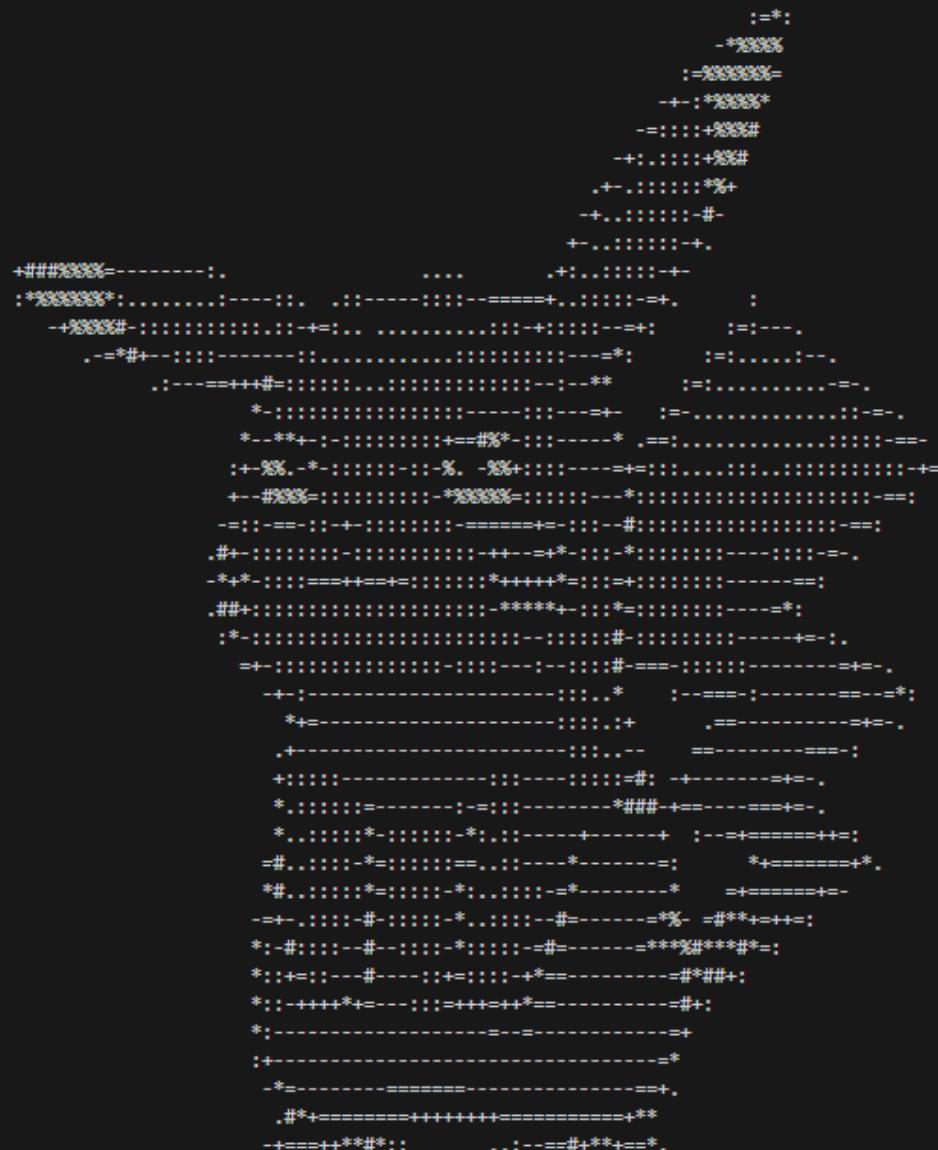
### f. F05 – Monster

RAWRRR, Agent X mengeluarkan monster Sukuna !!!

Name	:	Sukuna
ATK Power	:	650
DEF Power	:	30
HP	:	1000
Level	:	1

Gambar 8. 21 19tampilan monster pada battle\_1





RAWRRR, Agent X mengeluarkan monster Pikachuow !!!

Name	:	Pikachow
ATK Power	:	137
DEF Power	:	11
HP	:	660
Level	:	2

Gambar monster dimunculkan pada battle

*Gambar 8. 2321 tampilan monster pada battle\_3*

#### g. F06 – Potion

Pemanggilan potion di battle

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

Pilih perintah: 2

===== POTION LIST =====
1. Resilience Potion (Qty: 7) - Increases DEF Power
2. Cancel

Pilih potion untuk diminum: [REDACTED]
```

Gambar 8.24 22 Pemanggilan potion di battle

**Input yang valid**

```
===== POTION LIST =====
1. Resilience Potion (Qty: 7) - Increases DEF Power
2. Cancel

Pilih potion untuk diminum: 2

Pilih perintah: [REDACTED]
```

Gambar 8.25 23 Contoh input valid\_1

**Input yang valid**

```
===== POTION LIST =====
1. Resilience Potion (Qty: 7) - Increases DEF Power
2. Cancel

Pilih potion untuk diminum: 1

Setelah meminum ramuan ini, muncul sebuah energi pelindung di sekitar Pikachu yang membuatnya terlihat semakin tangguh dan sulit dilukai.
```

Gambar 8.26 Contoh input yang valid\_2

**Input tidak valid**

```
===== POTION LIST =====
1. Resilience Potion (Qty: 6) - Increases DEF Power
2. Cancel

Pilih potion untuk diminum: 3

Pilihan nomor tidak tersedia!

===== POTION LIST =====
1. Resilience Potion (Qty: 6) - Increases DEF Power
2. Cancel

Pilih potion untuk diminum: [REDACTED]
```

Gambar 8.2247 Contoh input tidak valid\_1

**Input tidak valid**

```
===== POTION LIST =====
1. Resilience Potion (Qty: 7) - Increases DEF Power
2. Cancel
```

Pilih potion untuk diminum: set  
Pilihan nomor tidak tersedia!

Pilih potion untuk diminum:

Gambar 8.28 25 Contoh input tidak valid\_2

Input tidak valid, kasus kosong

```
===== POTION LIST =====
1. Resilience Potion (Qty: 7) - Increases DEF Power
2. Cancel
```

Pilih potion untuk diminum:  
Input tidak boleh kosong!

Gambar 8.29 26 Input tidak valid kasus kosong

#### **h. F07 – Inventory**

Pemanggilan inventory dalam menu HELP

Masukkan perintah!  
Ketik "HELP" untuk melihat menu  
>>> inventory

Selamat datang di Inventory!  
Anda dapat melihat detail monster dan item yang dimiliki pada Inventory

Pilih aksi yang Anda ingin lakukan!  
1.Buka Inventory  
2.Keluar

Masukkan angka dari aksi yang ingin Anda lakukan:

Gambar 8.30 27 Pemanggilan inventory dalam menu HELP

Penggunaan input valid

Pilih aksi yang Anda ingin lakukan!  
1.Buka Inventory  
2.Keluar

Masukkan angka dari aksi yang ingin Anda lakukan: 1

```
===== INVENTORY LIST (User ID: 3) =====
```

Jumlah O.W.C.A Coin-mu sekarang 900.  
1. Monster (Name: Pikachu, Lvl. 2, HP: 600)  
2. Monster (Name: Sukuna, Lvl. 1, HP: 1000)  
3. Potion (Type: DEF, Qty: 7)

Ketikkan id untuk menampilkan detail item:

Gambar 8.31 28 Penggunaan input valid

Penggunaan input valid

```
Ketikkan id untuk menampilkan detail item: 2
```

Monster

```
Name      : Sukuna
ATK Power : 650
DEF Power : 30
HP       : 1000
Level    : 1
```

Pilih aksi yang Anda ingin lakukan!

- 1.Buka Inventory
- 2.Keluar

Masukkan angka dari aksi yang ingin Anda lakukan:

*Gambar 8.32 29 Penggunaan input yang valid pada inventory*

### Penggunaan input valid

```
Ketikkan id untuk menampilkan detail item: 2
```

Monster

```
Name      : Sukuna
ATK Power : 650
DEF Power : 30
HP       : 1000
Level    : 1
```

Pilih aksi yang Anda ingin lakukan!

- 1.Buka Inventory
- 2.Keluar

Masukkan angka dari aksi yang ingin Anda lakukan:

*Gambar 8.33 30 Penggunaan input yang valid pada inventory*

### Penggunaan input valid

Monster

```
Name      : Pikachu
ATK Power : 137
DEF Power : 11
HP       : 660
Level    : 2
```

Pilih aksi yang Anda ingin lakukan!

- 1.Buka Inventory
- 2.Keluar

Masukkan angka dari aksi yang ingin Anda lakukan: 2  
Anda keluar dari Inventory!

Masukkan perintah!

Ketik "HELP" untuk melihat menu

>>>

*Gambar 8.34 31 Penggunaan input valid pada inventory*

**Perintah keluar dari program**

```
Pilih aksi yang ingin Anda lakukan!
1. Upgrade monster
2. Keluar

Ketik angka dari aksi yang ingin Anda lakukan: 2

Anda keluar dari area lab monster!

Masukkan perintah!
Ketik "HELP" untuk melihat menu
>>> █
```

Gambar 8.35 32 Perintah keluar dari lab

**Penggunaan input tidak valid, input angka yang tidak terdaftar**

```
Selamat datang di Inventory!
Anda dapat melihat detail monster dan item yang dimiliki pada Inventory

Pilih aksi yang Anda ingin lakukan!
1.Buka Inventory
2.Keluar

Masukkan angka dari aksi yang ingin Anda lakukan: 3
Aksi tidak valid! Masukkan ulang input!

Pilih aksi yang Anda ingin lakukan!
1.Buka Inventory
2.Keluar

Masukkan angka dari aksi yang ingin Anda lakukan: █
```

Gambar 8.36 33 Input tidak valid dengan angka lain

**Penggunaan input tidak valid, input angka yang tidak terdaftar**

```
Masukkan angka dari aksi yang ingin Anda lakukan: 1

===== INVENTORY LIST (User ID: 3) =====
Jumlah O.W.C.A Coin-mu sekarang 900.
1. Monster      (Name: Pikachu, Lvl. 2, HP: 600)
2. Monster      (Name: Sukuna, Lvl. 1, HP: 1000)
3. Potion       (Type: DEF, Qty: 7)

Ketikkan id untuk menampilkan detail item: 0
Input tidak valid! Masukkan angka yang sesuai dengan jumlah jenis barang yang Anda punya!
```

Gambar 8.37 34 Input tidak valid dengan angka lain

**Penggunaan input tidak valid, input angka kosong**

Masukkan angka dari aksi yang ingin Anda lakukan: 1

===== INVENTORY LIST (User ID: 3) =====

Jumlah O.W.C.A Coin-mu sekarang 900.

1. Monster (Name: Pikachu, Lvl. 2, HP: 600)
2. Monster (Name: Sukuna, Lvl. 1, HP: 1000)
3. Potion (Type: DEF, Qty: 7)

Ketikkan id untuk menampilkan detail item:

Input tidak boleh kosong!

Gambar 8.38. 35 Input tidak valid dengan input kosong

Input tidak valid dengan string bebas

Potion

Type : DEF  
Quantity : 7

Pilih aksi yang Anda ingin lakukan!

- 1.Buka Inventory
- 2.Keluar

Masukkan angka dari aksi yang ingin Anda lakukan: asdw  
Aksi tidak valid! Masukkan ulang input!

Gambar 8.39 36 Input tidak valid dengan input string bebas

### i. F08 – Battle

Input masuk ke program battle tidak valid

Halo Agent catto! 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: Melihat detail monster, potion, dan OC yang dimiliki oleh Agent
3. Battle: Bertarung lvl melawan monster
4. Arena: Melatih monster yang dimiliki oleh Agent
5. Shop: Membeli item berupa monster atau potion
6. Laboratory: Meningkatkan level monster yang dimiliki oleh Agent
7. Save: Menyimpan data ke dalam folder
8. Exit: Keluar dari program

Footnote:

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

Masukkan perintah!

Ketik "HELP" untuk melihat menu

>>> 3

Perintah tidak valid!

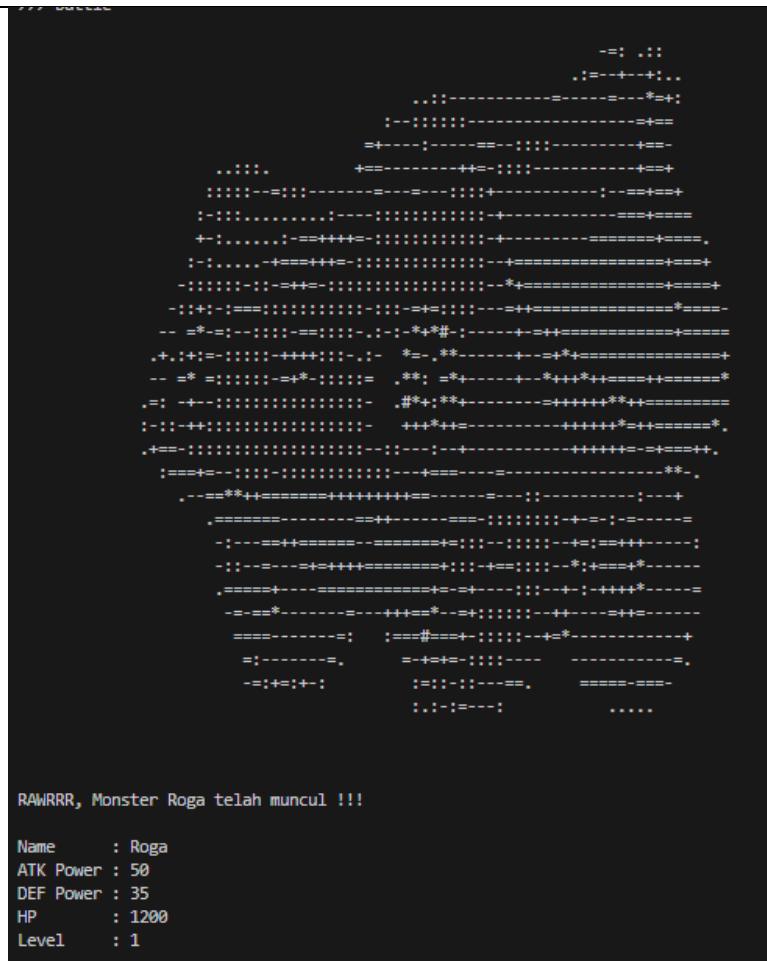
Gambar 8. 37 Input masuk ke program battle tidak valid

Input masuk ke program battle valid

Masukkan perintah!  
Ketik "HELP" untuk melihat menu  
>>> battle

Gambar 8. Input masuk ke program battle valid

Program memunculkan monster user



*Gambar 8. Program memunculkan monster user*

Program meminta user memilih monster lawan saat invalid dan valid

```
===== MONSTER LIST =====
1. Pikachu
2. Sukuna
Pilih monster untuk bertarung: sukuna

Pilihan nomor tidak tersedia!

Pilih monster untuk bertarung: 2
```

*Gambar 8. Program meminta user memilih monster lawan*

RAWRRR, Agent X mengeluarkan monster Sukuna !!!

Name : Sukuna  
ATK Power : 650  
DEF Power : 30  
HP : 1000  
Level : 1

*Gambar 8. Data dari monster lawan*

Giliran pertama dimana user memilih komponen apa yang akan digunakan untuk menyerang

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

Pilih perintah: 1

SCHWINKKK, Sukuna menyerang Roga !!!

Name      : Roga
ATK Power : 50
DEF Power : 35
HP        : 671
Level     : 1

Penjelasan: ATT: 812.5 (+25%), Reduced by: 284.375 (35%), ATT Results: 528.125

===== TURN 1 (Roga) =====
SCHWINKKK, Roga menyerang Sukuna !!!

Name      : Sukuna
ATK Power : 650
DEF Power : 30
HP        : 958
Level     : 1

Penjelasan: ATT: 59.5 (+19%), Reduced by: 17.84999999999998 (30%), ATT Results: 41.650000000000006
```

Gambar 8. Giliran pertama di dalam battle

Giliran kedua dimana user memilih komponen apa yang akan digunakan untuk menyerang

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

Pilih perintah: 1

SCHWINNKKK, Sukuna menyerang Roga !!!

Name      : Roga
ATK Power : 50
DEF Power : 35
HP        : 151
Level     : 1

Penjelasan: ATT: 799.5 (+23%), Reduced by: 279.825 (35%), ATT Results: 519.675

===== TURN 2 (Roga) =====
SCHWINNKKK, Roga menyerang Sukuna !!!

Name      : Sukuna
ATK Power : 650
DEF Power : 30
HP        : 923
Level     : 1

Penjelasan: ATT: 49.0 (-2%), Reduced by: 14.7 (30%), ATT Results: 34.3
```

*Gambar 8. Giliran kedua di dalam battle*

Giliran ketiga dimana user memilih komponen apa yang akan digunakan untuk menyerang

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

Pilih perintah: 1

SCHWINNKKK, Sukuna menyerang Roga !!!

Name      : Roga
ATK Power : 50
DEF Power : 35
HP        : 0
Level     : 1

Penjelasan: ATT: 760.5 (+17%), Reduced by: 266.175 (35%), ATT Results: 494.325

Selamat, Anda berhasil mengalahkan monster Roga !!!
OC yang diperoleh: 19
```

*Gambar 8. Giliran ketiga di dalam battle*

User memilih untuk melanjutkan battle

Apakah Anda ingin melakukan Battle lagi? (Y/N) y

*Gambar 8. User memilih untuk melanjutkan battle*

RAWR, Monster Chacha telah muncul !!!

Name : Chacha  
ATK Power : 100  
DEF Power : 30  
HP : 700  
Level : 1

*Gambar 8. Program mengeluarkan monster user*

User memilih untuk tidak melanjutkan battle

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

Pilih perintah: 3
Anda berhasil kabur dari BATTLE!

=====
Apakah Anda ingin melakukan Battle lagi? (Y/N) n
Sampai jumpa di BATTLE selanjutnya, Agent!
```

Gambar 8. use memilih untuk tidak melanjutkan battle setelah putaran pertama

#### j. F09 – Arena

Memanggil perintah arena

```
>>> arena
Selamat datang di Arena!!

=====
MONSTER LIST
1. Levi
Pilih monster untuk bertarung:
```

Gambar 8. 38 Memanggil arena

Input kosong (tidak valid)

```
Pilih monster untuk bertarung:

Input tidak boleh kosong!
Pilihan nomor tidak tersedia!
```

Gambar 8. 39 Input kosong arena

Memilih monster yang valid dan memulai arena

```
RAWRRR, Agent X mengeluarkan monster Levi !!!
```

```
Name      : Levi
ATK Power : 300
DEF Power : 15
HP       : 740
Level    : 1
```

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

Pilih perintah: 1

Gambar 8. 40 Input valid

Memilih aksi 1 (attack)

```
Pilih perintah: 1
```

```
SCHWINKKK, Levi menyerang Sukuna !!!
```

```
Name      : Sukuna
ATK Power : 650
DEF Power : 30
HP       : 817
Level    : 1
```

```
Penjelasan: ATT: 261.0 (-13%), Reduced by: 78.3 (30%), ATT Results: 182.7
```

```
===== TURN 1 (Sukuna) =====
```

```
SCHWINKKK, Sukuna menyerang Levi !!!
```

```
Name      : Levi
ATK Power : 300
DEF Power : 15
HP       : 82
Level    : 1
```

```
Penjelasan: ATT: 773.5 (+19%), Reduced by: 116.02499999999999 (15%), ATT Results: 657.475
```

```
===== TURN 2 (Levi) =====
```

```
1. Attack
2. Use Potion
3. Quit
```

Menggunakan potion



Menggunakan potion yang sama untuk kedua kali



Pengguna kalah

Yahhh, Anda dikalahkan monster Sukuna. Jangan menyerah, coba lagi !!!  
GAME OVER! Sesi latihan berakhir pada stage 1!

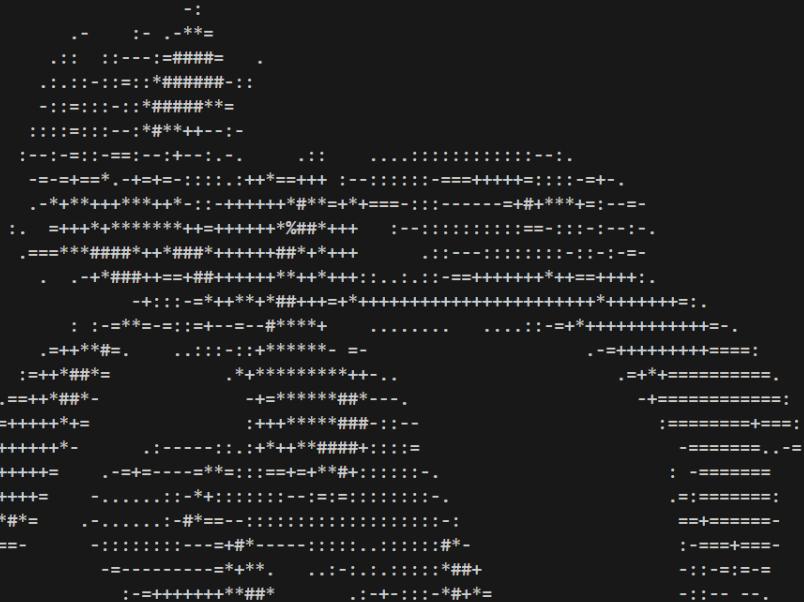
```
===== STATS =====
Total hadiah      : 0 OC
Jumlah stage     : 1
Damage diberikan  : 182
Damage diterima   : 1143
```

### Pengguna menang

Selamat, Anda berhasil mengalahkan monster Zuko !!!

STAGE CLEARED! Anda akan mendapatkan 30 OC pada sesi ini!

Memulai stage berikutnya...



Keluar dari arena dengan perintah

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

Pilih perintah: 3
Anda berhasil kabur dari BATTLE!
GAME OVER! Sesi latihan berakhir pada stage 2!

===== STATS =====
Total hadiah      : 30 OC
Jumlah stage     : 2
Damage diberikan  : 1286
Damage diterima   : 422
```

#### k. F10 - Shop & Currency

Penampilan seluruh pilihan monster atau potion berhasil

```
>>> shop
SHOP MENU:
1. Lihat
2. Beli
3. Keluar
Masukkan pilihan anda (1/2/3): 1
Anda ingin melihat Potion/Monster: monster
id | type      | atk_power | def_power | hp    | stok | harga
1  | Pikachuow | 125       | 10        | 600   | 6    | 500
2  | Zuko       | 85        | 20        | 800   | 2    | 800
3  | Chacha    | 100       | 30        | 700   | 10   | 2000
4  | Roga       | 50        | 35        | 1200  | 8    | 250
5  | Levi       | 300       | 15        | 740   | 5    | 600
```

Gambar 8.30 Penampilan monster berhasil

Penampilan seluruh pilihan monster atau potion berhasil

```
>>> shop
SHOP MENU:
1. Lihat
2. Beli
3. Keluar
Masukkan pilihan anda (1/2/3): 1
Anda ingin melihat Potion/Monster: potion
id | type      | stock | price
1  | strength  | 10    | 50
2  | resilience | 5    | 30
3  | healing    | 3    | 20
```

Gambar 8.31 Penampilan potion berhasil

Penampilan seluruh pilihan monster atau potion gagal

```
>>> shop
SHOP MENU:
1. Lihat
2. Beli
3. Keluar
Masukkan pilihan anda (1/2/3): 1
Anda ingin melihat Potion/Monster: dua duanya
Masukkan anda tidak valid. Masukkan lagi!
SHOP MENU:
1. Lihat
2. Beli
3. Keluar
Masukkan pilihan anda (1/2/3): 1
Anda ingin melihat Potion/Monster: potion
id | type      | stock | price
1  | strength  | 10    | 50
2  | resilience | 5     | 30
3  | healing    | 3     | 20
```

Gambar 8.32 Input penampilan monster atau potion tidak valid

## Pembelian monster atau potion berhasil

```
>>> shop
SHOP MENU:
1. Lihat
2. Beli
3. Keluar
Masukkan pilihan anda (1/2/3): 2
Jumlah O.W.C.A Coin-mu sekarang 900.
Anda ingin membeli Potion/Monster: monster
id | type      | atk_power | def_power | hp   | stok | harga
1  | Pikachow  | 125       | 10        | 600  | 6    | 500
2  | Zuko      | 85        | 20        | 800  | 2    | 800
3  | Chacha    | 100       | 30        | 700  | 10   | 2000
4  | Roga      | 50        | 35        | 1200 | 8    | 250
5  | Levi      | 300       | 15        | 740  | 5    | 600
Masukkan ID monster: 5
Kamu berhasil membeli Levi. Item sudah masuk ke inventory-mu!
Sisa O.W.C.A. Coin-mu 300.
```

Gambar 8.33 Pembelian monster berhasil

## Pembelian monster atau potion berhasil

```

SHOP MENU:
1. Lihat
2. Beli
3. Keluar
Masukkan pilihan anda (1/2/3): 2
Jumlah O.W.C.A Coin-mu sekarang 900.
Anda ingin membeli Potion/Monster: potion
id | type      | stock | price
1  | strength   | 10    | 50
2  | resilience | 5     | 30
3  | healing    | 3     | 20
Masukkan ID potion: 1
Masukkan kuantitas: 2
Kamu berhasil membeli 2 strength Potion. Item sudah masuk ke inventory-mu!
Sisa O.W.C.A. Coin-mu 800.

```

Gambar 8.34 Pembelian potion berhasil

## Pembelian monster atau potion gagal

```

SHOP MENU:
1. Lihat
2. Beli
3. Keluar
Masukkan pilihan anda (1/2/3): 2
Jumlah O.W.C.A Coin-mu sekarang 50.
Anda ingin membeli Potion/Monster: monster
id | type      | atk_power | def_power | hp    | stok | harga
1  | Pikachow  | 125       | 10        | 600   | 6    | 500
2  | Zuko       | 85        | 20        | 800   | 2    | 800
3  | Chacha    | 100       | 30        | 700   | 10   | 2000
4  | Roga       | 50         | 35        | 1200  | 7    | 250
5  | Levi       | 300       | 15        | 740   | 4    | 600
Masukkan ID monster: 3
O.W.C.A Coin-mu kurang.

```

Gambar 8.35 Pembelian monster gagal karena koin tidak mencukupi

## Pembelian monster atau potion gagal

```

SHOP MENU:
1. Lihat
2. Beli
3. Keluar
Masukkan pilihan anda (1/2/3): 2
Jumlah O.W.C.A Coin-mu sekarang 50.
Anda ingin membeli Potion/Monster: potion
id | type      | stock | price
1  | strength   | 10    | 50
2  | resilience | 5     | 30
3  | healing    | 3     | 20
Masukkan ID potion: 1
Masukkan kuantitas: 10
O.W.C.A Coin-mu kurang.

```

Gambar 8.36 Pembelian potion gagal karena koin tidak mencukupi

## Pembelian monster atau potion gagal

```

SHOP MENU:
1. Lihat
2. Beli
3. Keluar
Masukkan pilihan anda (1/2/3): 2
Jumlah O.W.C.A Coin-mu sekarang 50.
Anda ingin membeli Potion/Monster: potion
id | type      | stock | price
1  | strength  | 10    | 50
2  | resilience | 5     | 30
3  | healing   | 3     | 20
Masukkan ID potion: 1
Masukkan kuantitas: 10
O.W.C.A Coin-mu kurang.

```

Gambar 8.37 Pembelian potion gagal karena koin tidak mencukupi

## Pembelian monster atau potion gagal

```

SHOP MENU:
1. Lihat
2. Beli
3. Keluar
Masukkan pilihan anda (1/2/3): 2
Jumlah O.W.C.A Coin-mu sekarang 800.
Anda ingin membeli Potion/Monster: potion
id | type      | stock | price
1  | strength  | 8     | 50
2  | resilience | 5     | 30
3  | healing   | 3     | 20
Masukkan ID potion: 1
Masukkan kuantitas: 10
Stok potion tidak mencukupi.

```

Gambar 8.38 Pembelian potion gagal karena stok tidak mencukupi

## Pembelian monster atau potion gagal

```

SHOP MENU:
1. Lihat
2. Beli
3. Keluar
Masukkan pilihan anda (1/2/3): 2
Jumlah O.W.C.A Coin-mu sekarang 800.
Anda ingin membeli Potion/Monster: potion
id | type      | stock | price
1  | strength  | 8     | 50
2  | resilience | 5     | 30
3  | healing   | 3     | 20
Masukkan ID potion: 4
Masukkan kuantitas: 1
Masukkan anda tidak valid.

```

Gambar 8.39 Pembelian potion gagal karena input tidak valid

## Pembelian monster atau potion gagal

```

SHOP MENU:
1. Lihat
2. Beli
3. Keluar
Masukkan pilihan anda (1/2/3): 2
Jumlah O.W.C.A Coin-mu sekarang 837.
Anda ingin membeli Potion/Monster: monster
id | type      | atk_power | def_power | hp    | stok | harga
1  | Pikachuow | 125       | 10        | 600   | 5    | 500
2  | Zuko       | 85        | 20        | 800   | 2    | 800
3  | Chacha    | 100       | 30        | 700   | 10   | 2000
4  | Roga       | 50        | 35        | 1200  | 7    | 250
5  | Levi       | 300       | 15        | 740   | 4    | 600
Masukkan ID monster: 6
Masukkan anda tidak valid.

```

Gambar 8.40 Pembelian monster gagal karena input tidak valid

```

Pembelian monster atau potion gagal
>>> shop
SHOP MENU:
1. Lihat
2. Beli
3. Keluar
Masukkan pilihan anda (1/2/3): 2
Jumlah O.W.C.A Coin-mu sekarang 1337.
Anda ingin membeli Potion/Monster: monster
id | type      | atk_power | def_power | hp    | stok | harga
1  | Pikachuow | 125       | 10        | 600   | 6    | 500
2  | Zuko       | 85        | 20        | 800   | 2    | 800
3  | Chacha    | 100       | 30        | 700   | 10   | 2000
4  | Roga       | 50        | 35        | 1200  | 7    | 250
5  | Levi       | 300       | 15        | 740   | 4    | 600
Masukkan ID monster: 4
Anda sudah memiliki monster ini, masukkan ID lain!
Masukkan ID monster: 1
Kamu berhasil membeli Pikachuow. Item sudah masuk ke inventory-mu!
Sisa O.W.C.A. Coin-mu 837.

```

Gambar 8.41 Pembelian monster gagal karena sudah dimiliki

## I. F11 – Laboratory

```

Pemanggilan laboratory pada HELP
Ketik "HELP" untuk melihat menu
>>> laboratory
Kamu telah memasuki area lab monster!

Pilih aksi yang ingin Anda lakukan!
1. Upgrade monster
2. Keluar

Ketik angka dari aksi yang ingin Anda lakukan: 1

```

Gambar 8.42 Pemanggilan laboratory pada HELP

Input valid

```
Ketik angka dari aksi yang ingin Anda lakukan: 1
```

```
----- MONSTERS OWNED -----
```

1. Pikachu (Level : 2)
2. Sukuna (Level : 1)

```
----- UPGRADE PRICE LIST -----
```

1. Level 1 -> Level 2: 250 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1250 OC

```
Jumlah koin yang dimiliki: 900
```

```
Pilih angka dari monster yang ingin di-upgrade : 1
```

Gambar 8.43 42 Contoh input valid

#### Input valid

```
Ketik angka dari aksi yang ingin Anda lakukan: 1
```

```
----- MONSTERS OWNED -----
```

1. Pikachu (Level : 2)
2. Sukuna (Level : 1)

```
----- UPGRADE PRICE LIST -----
```

1. Level 1 -> Level 2: 250 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1250 OC

```
Jumlah koin yang dimiliki: 900
```

```
Pilih angka dari monster yang ingin di-upgrade : 2
```

```
Sukuna akan diupdate ke level 2.
```

```
Harga upgrade ini adalah 250.
```

```
Lanjutkan upgrade? (Y/N): y
```

```
Upgrade berhasil!
```

```
Sukuna sekarang ada di level 2!
```

```
Pilih aksi yang ingin Anda lakukan!
```

1. Upgrade monster
2. Keluar

```
Ketik angka dari aksi yang ingin Anda lakukan: 1
```

Gambar 8.44 43 Contoh input valid

#### Contoh input tidak valid

```
Pilih aksi yang ingin Anda lakukan!
```

1. Upgrade monster
2. Keluar

```
Ketik angka dari aksi yang ingin Anda lakukan: 0
```

```
Aksi tidak valid! Masukkan ulang aksi yang ingin Anda lakukan!
```

```
Pilih aksi yang ingin Anda lakukan!
```

1. Upgrade monster
2. Keluar

```
Ketik angka dari aksi yang ingin Anda lakukan: 1
```

Gambar 8.45 44 input tidak valid dengan angka lain

Contoh input tidak valid

```
Pilih aksi yang ingin Anda lakukan!
1. Upgrade monster
2. Keluar

Ketik angka dari aksi yang ingin Anda lakukan: str

Aksi tidak valid! Masukkan ulang aksi yang ingin Anda lakukan!
```

Gambar 8.46 45 Input tidak valid dengan string bebas

Contoh input tidak valid

```
Pilih aksi yang ingin Anda lakukan!
1. Upgrade monster
2. Keluar

Ketik angka dari aksi yang ingin Anda lakukan:

Aksi tidak valid! Masukkan ulang aksi yang ingin Anda lakukan!

Pilih aksi yang ingin Anda lakukan!
1. Upgrade monster
2. Keluar
```

Gambar 8.47 46 Input tidak valid dengan input kosong

## m. F12 - Shop Management

Melihat list potion

```
>>> shop management
Shop Management System (Admin)
1. Lihat
2. Tambah
3. Update
4. Hapus
5. Keluar
Masukkan pilihan anda: 1
Anda ingin melihat Potion/Monster: potion
id | type      | stock | price
1  | strength  | 10    | 50
2  | resilience | 5     | 30
3  | healing    | 3     | 20
```

Gambar 8. 4478 Melihat isi potion

Melihat list monster

```
Shop Management System (Admin)
1. Lihat
2. Tambah
3. Update
4. Hapus
5. Keluar
Masukkan pilihan anda: 1
Anda ingin melihat Potion/Monster: monster
id | type      | atk_power | def_power | hp   | stok | harga
1  | Pikachuow | 125      | 10        | 600  | 6    | 500
2  | Zuko       | 85       | 20        | 800  | 2    | 800
3  | Chacha     | 100      | 30        | 700  | 10   | 2000
4  | Roga        | 50       | 35        | 1200 | 8    | 250
5  | Levi        | 300      | 15        | 740  | 5    | 600
```

Gambar 8. 49 48 Melihat isi monster

Menambah stok monster invalid

```

Shop Management System (Admin)
1. Lihat
2. Tambah
3. Update
4. Hapus
5. Keluar
Masukkan pilihan anda: 2
Mau menambahkan apa? (monster/potion): MONSTER
id | type   | atk_power | def_power | hp
6  | Sukuna | 650      | 30        | 1000
Masukkan ID monster: 6
Masukkan stok awal: 3
Masukkan harga: -1
Masukkan harga: 3

```

Gambar 8.50 Stok monster tidak valid

```

Menambahkan stok monster valid
Mau menambahkan apa? (monster/potion): MONSTER
id | type   | atk_power | def_power | hp
6  | Sukuna | 650      | 30        | 1000
Masukkan ID monster: 6
Masukkan stok awal: 3
Masukkan harga: -1
Masukkan harga: 3
Monster baru berhasil ditambahkan ke dalam toko!

```

Gambar 8.51 Stok monster valid

```

Memperbarui potion invalid
Komponen apa yang ingin diperbaharui? (monster/potion): potion
id | type   | stock | price
1  | strength | 10    | 50
2  | resilience | 6    | 40
3  | healing   | 3    | 20
Masukkan ID potion: 5
ID salah! Masukkan ID kembali
Masukkan ID potion: 1

```

Gambar 8.52 ID potion tidak valid

```

Memperbarui potion valid
Komponen apa yang ingin diperbaharui? (monster/potion): potion
id | type   | stock | price
1  | strength | 10    | 50
2  | resilience | 5    | 30
3  | healing   | 3    | 20
Masukkan ID potion: 2
Masukkan harga baru: 40
Masukkan stok baru: 6

```

Gambar 8.53 ID potion valid

```

Menghapus monster invalid
Komponen apa yang ingin dihapus? (monster/potion): monster
id | type   | atk_power | def_power | hp   | stok | harga
1  | Pikachow | 125      | 10        | 600  | 6    | 500
2  | Zuko    | 85       | 20        | 800  | 2    | 800
3  | Chacha  | 100      | 30        | 700  | 10   | 2000
4  | Roga    | 50       | 35        | 1200 | 8    | 250
5  | Levi    | 300      | 15        | 740  | 5    | 600
Masukkan id monster: 6
ID salah! Masukkan ID kembali

```

Gambar 8.54 ID monster tidak valid

Menghapus monster valid

```
Komponen apa yang ingin dihapus? (monster/potion): monster
id | type      | atk_power | def_power | hp    | stok | harga
1  | Pikachuow | 125       | 10        | 600   | 6    | 500
2  | Zuko       | 85        | 20        | 800   | 2    | 800
3  | Chacha    | 100       | 30        | 700   | 10   | 2000
4  | Roga       | 50        | 35        | 1200  | 8    | 250
5  | Levi       | 300       | 15        | 740   | 5    | 600
Masukkan id monster: 6
ID salah! Masukkan ID kembali
Masukkan id monster: 5
Apakah anda yakin ingin menghapus monster dari toko (iya/tidak): iya
Shop Management System (Admin)
1. Lihat
2. Tambah
3. Update
4. Hapus
5. Keluar
Masukkan pilihan anda: 1
Anda ingin melihat Potion/Monster: monster
id | type      | atk_power | def_power | hp    | stok | harga
1  | Pikachuow | 125       | 10        | 600   | 6    | 500
2  | Zuko       | 85        | 20        | 800   | 2    | 800
3  | Chacha    | 100       | 30        | 700   | 10   | 2000
```

Gambar 8. 49

#### n. F13 - Monster Management

Memanggil perintah monster management

```
>>> monster management
```

Selamat datang di database monster!

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

```
>>> Pilih aksi yang ingin dilakukan: [ ]
```

Gambar 8.56 Memanggil command monster managemment

Input tidak valid

```
>>> Pilih aksi yang ingin dilakukan:  
Aksi yang dipilih tidak ditemukan.
```

Gambar 8.57 Aksi yang dipilih tidak ditemukan pada monster management

Memilih aksi 1 (menampilkan data semua monster)

```
>>> Pilih aksi yang ingin dilakukan: 1
```

ID	Type	ATK Power	DEF Power	HP
1	Pikachow	125	10	600
2	Zuko	85	20	800
3	Chacha	100	30	700
4	Roga	50	35	1200
5	Levi	300	15	740
6	Sukuna	650	30	1000

Gambar 8. 50

Memilih aksi 2 (menambahkan monster baru) dengan semua input valid

```
>>> Pilih aksi yang ingin dilakukan: 2
```

Saatnya membuat monster baru!

```
>>> Masukkan tipe atau nama: Naruto
```

```
>>> Masukkan ATK Power: 299
```

```
>>> Masukkan DEF Power (0-50): 45
```

```
>>> Masukkan HP: 1000
```

Monser baru berhasil dibuat!

Type: Naruto

ATK Power: 299

DEF Power: 45

HP: 1000

```
>>> Tambahkan monster ke database (Y/N): y
```

Monster berhasil ditambahkan ke database!

Gambar 8. 51

```

Selamat datang di database monster!
1. Tampilkan semua monster
2.Tambahkan monster baru
3. Keluar

>>> Pilih aksi yang ingin dilakukan: 1

ID | Type          | ATK Power | DEF Power | HP
1  | Pikachuow    | 125       | 10         | 600
2  | Zuko          | 85        | 20         | 800
3  | Chacha        | 100       | 30         | 700
4  | Roga           | 50        | 35         | 1200
5  | Levi           | 300       | 15         | 740
6  | Sukuna         | 650       | 30         | 1000
7  | Puyeng          | 1000      | 50         | 10
8  | Naruto          | 299       | 45         | 1000

```

Gambar 8. 52

Gambar 8.59 Membuat monster baru berhasil

Kasus input tidak dalam range 0-50, kosong, dan berupa string

```

Saatnya membuat monster baru!
>>> Masukkan tipe atau nama: contoh
>>> Masukkan ATK Power: 999
>>> Masukkan DEF Power (0-50): 100
Input tidak valid!
>>> Masukkan DEF Power:
Input tidak valid!
>>> Masukkan DEF Power: 09
>>> Masukkan HP: ads
Input tidak valid!
>>> Masukkan HP: 3488

```

Pengguna tidak jadi menambahkan monster

**Monser baru berhasil dibuat!**

Type: contoh  
 ATK Power: 999  
 DEF Power: 09  
 HP: 3488

>>> Tambahkan monster ke database (Y/N): n

**Monster gagal ditambahkan ke database!**

Gambar 8. 53

Pengguna memilih aksi 3 (keluar dari monster management)

```

Selamat datang di database monster!
1. Tampilkan semua monster
2. Tambahkan monster baru
3. Keluar

>>> Pilih aksi yang ingin dilakukan: 3

Anda keluar dari database monster!

Masukkan perintah!
Ketik "HELP" untuk melihat menu
>>> █

```

Gambar 8. 54

#### **o. F14 – Load**

Folder yang dipanggil untuk load ada

```

PS C:\Data lain\BERTHA\TUBES_DASPRO\if1210-2024-tubes-k07-a> python main.py default
Loading...
Selamat datang di program OWCA!

Masukkan perintah!
Ketik "HELP" untuk melihat menu
>>> █

```

Gambar 8. 55

Tidak ada nama folder yang diberikan

```

PS C:\Data lain\BERTHA\TUBES_DASPRO\if1210-2024-tubes-k07-a> python main.py
Tidak ada nama folder yang diberikan!
Usage : python main.py <nama_folder>

```

Gambar 8. 56

Folder yang diberikan tidak ada

```

PS C:\Data lain\BERTHA\TUBES_DASPRO\if1210-2024-tubes-k07-a> python main.py contoh
Folder "contoh" tidak ditemukan!

```

Gambar 8. 57

#### **p. F15 – Save**

Penyimpanan data dengan dimasukkan nama folder:

```

Masukkan perintah!
Ketik "HELP" untuk melihat menu
>>> save
Masukkan nama folder: testing.csv

Saving...

Membuat folder data/testing.csv
Berhasil menyimpan data di folder data/testing.csv!

```

Gambar 8. 5860 Penimpaenan

#### **q. F16 – Exit**

Penampilan saat exit dilakukan dengan penyimpanan data terlebih dahulu:

```
Masukkan perintah!
Ketik "HELP" untuk melihat menu
>>> exit
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (Y/N) Y
Masukkan nama folder: data_monstershop

Saving...

Membuat folder data/data_monstershop
Berhasil menyimpan data di folder data/data_monstershop!
PS C:\Users\ACER\Documents\Arduino\DasPRO\if1210-2024-tubes-k07-a>
```

Gambar 8. 59 Exit dengan penyimpanan data

Penampilan saat exit dilakukan tanpa menyimpan data terlebih dahulu:

```
Masukkan perintah!
Ketik "HELP" untuk melihat menu
>>> exit
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (Y/N) N
PS C:\Users\ACER\Documents\Arduino\DasPRO\if1210-2024-tubes-k07-a>
```

Gambar 8. 60 Exit tanpa penyimpanan data

## IX. Lampiran

### Lampiran 1: Form MoM Asistensi 1

**Form MoM Asistensi Tugas Besar  
IF1210/Dasar Pemrograman  
Sem. 2 2023/2024**

Nomor Asistensi	:	1														
No. Kelompok/Kelas	:	A/K-07														
Tanggal asistensi	:	2 Mei 2024														
<b>Anggota kelompok</b> <table border="1"> <thead> <tr> <th colspan="2">NIM / Nama (Hanya yang Hadir)</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>16523037 / Adrian Sami Pratama</td> </tr> <tr> <td>2</td> <td>16523167 / Felicia Kannitha Ruth</td> </tr> <tr> <td>3</td> <td>19623087 / Naomi Risaka Sitorus</td> </tr> <tr> <td>4</td> <td>19623107 / Bertha Soliany Frandi</td> </tr> <tr> <td>5</td> <td>19623157 / Michael Alexander Angkawijaya</td> </tr> <tr> <td>6</td> <td></td> </tr> </tbody> </table>			NIM / Nama (Hanya yang Hadir)		1	16523037 / Adrian Sami Pratama	2	16523167 / Felicia Kannitha Ruth	3	19623087 / Naomi Risaka Sitorus	4	19623107 / Bertha Soliany Frandi	5	19623157 / Michael Alexander Angkawijaya	6	
NIM / Nama (Hanya yang Hadir)																
1	16523037 / Adrian Sami Pratama															
2	16523167 / Felicia Kannitha Ruth															
3	19623087 / Naomi Risaka Sitorus															
4	19623107 / Bertha Soliany Frandi															
5	19623157 / Michael Alexander Angkawijaya															
6																
<b>Asisten pembimbing</b> <table border="1"> <thead> <tr> <th colspan="2">NIM / Nama</th> </tr> </thead> <tbody> <tr> <td colspan="2">13521019 / Ditra Rizqa Amadia</td> </tr> </tbody> </table>			NIM / Nama		13521019 / Ditra Rizqa Amadia											
NIM / Nama																
13521019 / Ditra Rizqa Amadia																

Catatan Asistensi:

<b>Rangkuman Diskusi</b>
Diskusi dilakukan dengan mengajukan pertanyaan. Berikut adalah jawaban dari pertanyaan-pertanyaan tersebut.
<ul style="list-style-type: none"> <li>- Saat battle, level monster musuh tergantung pilihan kelompok. Boleh flat level 1, boleh di random.</li> <li>- Untuk input yang tidak diwajibkan untuk divalidasi pada spesifikasi, jika ditambahkan akan masuk ke kreativitas. Contohnya, input password saat register tidak bisa berupa spasi atau list kosong. Begitu pula dengan fitur lainnya yang tidak dijelaskan pada spesifikasi.</li> <li>- Argparse digunakan hanya untuk load.</li> <li>- Tidak memungkinkan untuk load di tengah program. Load hanya dijalankan pada awal program dijalankan. User harus keluar dari program jika ingin load data lain.</li> <li>- Untuk load, harus ada nama folder yang dipanggil dan nama folder harus valid (sesuai dengan nama folder yang ada dan folder sudah ada)</li> <li>- Database CSV yang sudah diubah oleh kelompok (isinya berbeda dengan template), sebaiknya dimasukkan pula ke github. Hal ini berguna apabila terdapat asisten yang ingin mencoba untuk menjalankan program menggunakan database kelompok.</li> <li>- Maksud atribut monster yang balance adalah tidak ada monster yang terlalu OP (tidak bisa dikalahkan). Namun jika ingin membuat monster yang mempunyai kecenderungan “kuat” di salah satu atribut, masih diperbolehkan.</li> <li>- Jika admin memanggil command help, semua command yang bisa dipanggil oleh admin, ditampilkan. Hal ini juga termasuk dengan command save.</li> <li>- Kalau belum di save (memanggil command save), data pada database csv tidak akan berubah. Oleh karena itu, simpan segala sesuatu yang diubah di program (sebelum save) ke dalam suatu variabel.</li> <li>- Ketika save, pada suatu folder khusus untuk save, akan tersimpan satu folder yang berisi hasil database terbaru.</li> <li>- Kondisi sekarang pada F0 - Register adalah langsung menulis pada database csv mengenai data yang ditambahkan bila sudah valid. Namun dikarenakan poin sebelumnya, asisten memberikan solusi berupa mengubah hasil load menjadi sebuah variabel. Boleh menjadi array, matriks, dan array of library.</li> <li>- Kalau mulai branching/commit dari sekarang sudah termasuk bonus.</li> <li>- Tidak boleh digunakan: <ul style="list-style-type: none"> <li>- split</li> <li>- pop</li> <li>- read</li> <li>- readline</li> <li>- keyword global</li> </ul> </li> </ul>

**Form MoM Asistensi Tugas Besar**  
**IF1210/Dasar Pemrograman**  
**Sem. 2 2023/2024**

<ul style="list-style-type: none"> <li>- Boleh digunakan:           <ul style="list-style-type: none"> <li>- slicing</li> <li>- string formatting</li> </ul> </li> <li>- Diperbolehkan untuk membuat fungsi sendiri. Hal yang dimaksud adalah membuat fungsi split secara manual dan tidak menggunakan fungsi bawaan python.</li> <li>- Untuk F05 - Monster dan F06 - Potion, tidak diharuskan ada output (print). Namun, jika menambahkan fitur akan terhitung kreativitas.</li> <li>- Atribut DEF monster, hanya boleh dalam range 0-50. Lakukan pengecekan mengenai input base attribute monster pada F13 - Monster_management.</li> <li>- Boleh menghubungi asisten kapanpun itu melalui LINE. Disarankan hanya satu orang yang menghubungi (perwakilan kelompok) agar chat tidak tenggelam dan tidak terbaca oleh asisten.</li> </ul>
<p><b>Saran dari asisten:</b></p> <ul style="list-style-type: none"> <li>- Melakukan bonus 1 dan bonus 2 mulai dari sekarang dikarenakan hal ini akan membantu dalam hal nilai tugas besar dan untuk semester kedepannya.</li> <li>- Pelajari dan gunakan github (branching, command line, semantic, dsb).</li> </ul>
<p><b>Tindak Lanjut</b></p> <ul style="list-style-type: none"> <li>- Perbaikan pada register mengenai hasil input yang valid langsung tertuliskan pada database. Seharusnya penulisan ke database hanya terjadi pada saat command save dipanggil.</li> <li>- Pada F04 - Menu &amp; Help, command save dan exit juga ditampilkan.</li> <li>- Melakukan pembagian tugas untuk F06 sampai F16.</li> <li>- Menentukan deadline pengerjaan tugas dan asistensi 2.</li> <li>- Menentukan waktu untuk pengerjaan laporan dan pengecekan akhir program.</li> </ul>
<p><b>Dokumentasi</b></p> 

## Lampiran 2: Form MoM Asistensi 2

**Form MoM Asistensi Tugas Besar  
IF1210/Dasar Pemrograman  
Sem. 2 2023/2024**

Nomor Asistensi	:	2																					
No. Kelompok/Kelas	:	A/K-07																					
Tanggal asistensi	:	12 Mei 2024																					
<table border="1"> <thead> <tr> <th colspan="2">Anggota kelompok</th> <th>NIM / Nama (Hanya yang Hadir)</th> </tr> </thead> <tbody> <tr><td>1</td><td>16523037</td><td>/ Adrian Sami Pratama</td></tr> <tr><td>2</td><td>16523167</td><td>/ Felicia Kannitha Ruth</td></tr> <tr><td>3</td><td>19623087</td><td>/ Naomi Risaka Sitorus</td></tr> <tr><td>4</td><td>19623107</td><td>/ Bertha Soliany Frandi</td></tr> <tr><td>5</td><td></td><td></td></tr> <tr><td>6</td><td></td><td></td></tr> </tbody> </table>			Anggota kelompok		NIM / Nama (Hanya yang Hadir)	1	16523037	/ Adrian Sami Pratama	2	16523167	/ Felicia Kannitha Ruth	3	19623087	/ Naomi Risaka Sitorus	4	19623107	/ Bertha Soliany Frandi	5			6		
Anggota kelompok		NIM / Nama (Hanya yang Hadir)																					
1	16523037	/ Adrian Sami Pratama																					
2	16523167	/ Felicia Kannitha Ruth																					
3	19623087	/ Naomi Risaka Sitorus																					
4	19623107	/ Bertha Soliany Frandi																					
5																							
6																							
<table border="1"> <thead> <tr> <th colspan="2">Asisten pembimbing</th> <th>NIM / Nama</th> </tr> </thead> <tbody> <tr><td colspan="2"></td><td>13521019 / Ditra Rizqa Amadia</td></tr> </tbody> </table>			Asisten pembimbing		NIM / Nama			13521019 / Ditra Rizqa Amadia															
Asisten pembimbing		NIM / Nama																					
		13521019 / Ditra Rizqa Amadia																					

**Catatan Asistensi:**

- 19623157 / Michael Alexander Angkawijaya tidak hadir dikarenakan mengikuti lomba

Rangkuman Diskusi
Diskusi dilakukan dengan mengajukan pertanyaan. Berikut adalah jawaban dari pertanyaan-pertanyaan tersebut. <ul style="list-style-type: none"> <li>- semua import pada modul ditaruh di paling atas (lakukan git best practice)</li> <li>- pada load, file-file terjamin ada dan tidak harus di validasi, tapi untuk folder yang berisi file harus di validasi</li> <li>- bagusnya di dalam folder data ada folder lagi, parent folder nya sama seperti save</li> <li>- pada laporan, hanya fungsi utama yang dibuat notasi algoritmiknya. Untuk utilities tidak perlu dibuat notasinya</li> <li>- apabila membuat fungsi utility tidak perlu diubah menjadi notal (untuk shop management perlu dibuat)</li> <li>- line pada modul tidak apa panjang, tidak masuk ke dalam penilaian (tidak perlu efisien)</li> <li>- efisiensi kode akan memudahkan kelompok untuk mengerjakan tubes ini</li> <li>- fungsi map, filter, radius tidak boleh digunakan</li> <li>- pada demonstrasi program dari awal sampai akhir, asisten akan memberikan test case. Disarankan semua case di <i>handle</i> agar saat di test berjalan lancar. Selain itu, akan ditanya pula mengenai cara kerja program (kemungkinan ada asisten yang menanyakan cara kerja suatu fitur yang bukan suatu pribadi kerjakan)</li> <li>- demonstrasi belum ada kepastian diadakan offline atau online. Akan diberitahukan lebih lanjut</li> <li>- pada laporan notasi algoritmiknya saja dan pengujian programnya (saat program berhasil jalan dan tidak berhasil jalan)</li> <li>- print("   no/t   Nama \t   Tanggal\t ) biar tabelnya rapi, cari alternatif lain buat tabel yg rapi</li> <li>- apakah masing masing fungsi harus dijelaskan lagi kegunaannya? iya, sesuaikan dengan hasil dari program kelompok</li> <li>- pada laporan, semua penjelasan harus sesuai dengan program</li> <li>- pikirkan segala kondisi yang mungkin dan <i>handle</i> kasus itu</li> </ul>
Tindak Lanjut
<ul style="list-style-type: none"> <li>- Melakukan pengecekan ulang untuk function register dikarenakan terdapat validasi yang terlewati</li> <li>- Menggabungkan semua function agar menjadi satu program yang utuh di main.py</li> <li>- Membagi tugas untuk penggerjaan laporan dan mulai mengerjakan laporan</li> <li>- Mempertimbangkan untuk mengerjakan bonus</li> <li>- Melakukan testing dan memvalidasi kondisi yang mungkin terjadi</li> </ul>

**Form MoM Asistensi Tugas Besar  
IF1210/Dasar Pemrograman  
Sem. 2 2023/2024**

**Dokumentasi**

