

**LAPORAN TUGAS BESAR**  
**IF1210 DASAR PEMROGRAMAN**  
**PROGRAM *TURN-BASED GAME***

Disusun oleh:

**Kelas K-05**  
**Kelompok D**

<b>Carlo Angkisan</b>	<b>19623005</b>
<b>Matthew Sebastian Kurniawan</b>	<b>19623205</b>
<b>Abednego Ozoralael S</b>	<b>16523005</b>
<b>Dahayu Aqila Suseno</b>	<b>16523155</b>
<b>Mahardwitiyajati Satrio Prabowo</b>	<b>16523255</b>



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

**2024**

## HALAMAN PERNYATAAN

*“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.”*

**Kami yang menyatakan:**

- |   |                 |
|---|-----------------|
| <b>1. Carlo Angkisan</b>                  | <b>19623005</b> |
| <b>2. Matthew Sebastian Kurniawan</b>     | <b>19623205</b> |
| <b>3. Abednego Ozorael S</b>              | <b>16523005</b> |
| <b>4. Dahayu Aqila Suseno</b>             | <b>16523155</b> |
| <b>5. Mahardwitiyajati Satrio Prabowo</b> | <b>16523255</b> |

## DAFTAR ISI

HALAMAN PERNYATAAN.....	2
DAFTAR ISI.....	3
DAFTAR TABEL.....	4
DAFTAR GAMBAR.....	5
DESKRIPSI PERSOALAN.....	8
DAFTAR PEMBAGIAN TUGAS.....	11
CHECKLIST HASIL RANCANGAN.....	13
DESAIN COMMAND.....	15
DESAIN KAMUS DATA.....	17
DESAIN DEKOMPOSISI ALGORITMIK DAN FUNGSIONAL.....	23
SPESIFIKASI.....	24
HASIL PENGUJIAN PROGRAM.....	75
LAMPIRAN.....	100

## DAFTAR TABEL

Tabel 1 Daftar Pembagian Tugas.....	8
Tabel 2 Checklist Hasil Rancangan.....	10

## DAFTAR GAMBAR

Gambar 2.1.....	63
Gambar 2.2.....	63
Gambar 2.3.....	64
Gambar 3.1.....	64
Gambar 3.2.....	64
Gambar 3.3.....	64
Gambar 3.4.....	65
Gambar 4.1.....	65
Gambar 4.2.....	65
Gambar 5.1.....	65
Gambar 5.2.....	66
Gambar 5.3.....	66
Gambar 6.1.....	67
Gambar 6.2.....	67
Gambar 6.3.....	68
Gambar 7.1.....	68
Gambar 7.2.....	69
Gambar 7.3.....	70
Gambar 7.4.....	70
Gambar 7.6.....	70
Gambar 7.7.....	70
Gambar 7.8.....	70
Gambar 7.10.....	71
Gambar 7.11.....	71
Gambar 7.12.....	72
Gambar 8.1.....	72
Gambar 8.2.....	73
Gambar 8.3.....	73

Gambar 8.4.....	74
Gambar 8.5.....	75
Gambar 8.6.....	75
Gambar 8.7.....	76
Gambar 8.8.....	76
Gambar 9.1.....	76
Gambar 9.2.....	77
Gambar 9.3.....	77
Gambar 9.4.....	77
Gambar 9.5.....	77
Gambar 9.6.....	78
Gambar 9.7.....	78
Gambar 9.8.....	78
Gambar 10.1.....	79
Gambar 10.2.....	79
Gambar 10.3.....	79
Gambar 11.1.....	80
Gambar 11.2.....	80
Gambar 11.3.....	80
Gambar 11.4.....	81
Gambar 11.5.....	81
Gambar 11.6.....	81
Gambar 11.7.....	82
Gambar 11.8.....	82
Gambar 11.9.....	82
Gambar 11.10.....	83
Gambar 11.11.....	83
Gambar 11.12.....	83
Gambar 11.13.....	84

Gambar 12.1.....	84
Gambar 12.2.....	85
Gambar 12.3.....	85
Gambar 12.4.....	85

## DESKRIPSI PERSOALAN

Pada tugas besar ini, mahasiswa ditugaskan untuk membuat program *turn-based game*. Pada program ini, user berperan sebagai agent O.W.C.A (Organisasi Warga Cool Abiez) dengan objektif membantu Agent Purry mengalahkan Dr. Asep Spakbor yang telah menciptakan monster-monster kuat yang mengancam keamanan kota Danville. Agent Purry merasa monster-monster ini terlalu kuat untuknya, sehingga ia meminta user untuk mencari dan melatih monster-monster untuk digunakan dalam melawan Dr. Asep Spakbor. Misi mencari dan melatih monster-monster ini dimulai di hutan terpencil, karena diyakini bahwa di hutan terdapat banyak monster.

Untuk melaksanakan misi ini terdapat beberapa spesifikasi program, yaitu:

1. F00 - RNG

RNG akan menghasilkan bilangan acak dari sebuah *range* angka. Pada program tugas besar ini, RNG digunakan dalam jumlah OC (O.W.C.A Coin) yang didapatkan dan kemunculan monster yang acak.

2. F01 - Register

*Command Register* digunakan agar user dapat melakukan registrasi awal. Jika username dan password sudah valid, maka user akan memilih monster awal, mendapatkan *role* agent, dan 0 OC.

3. F02 - Login

Jika sudah melakukan registrasi, maka user dapat menggunakan *command Login* untuk *login* ke dalam sistem.

4. F03 - Logout

Jika sudah melakukan *login*, user dapat keluar dari akun yang sedang dipakai dengan *command Logout*. Setelah melakukan *logout*, user tidak bisa mengakses isi dari akun sebelumnya.

5. F04 - Menu & Help

User dapat menggunakan *command Help* untuk mendapat bantuan tentang apa yang dapat dilakukan selanjutnya. *Command* ini juga mengingatkan user tentang validasi input yang perlu diperhatikan.

6. F05 - Monster

Untuk menunjukkan monster-monster pada database “owca-dex” beserta type (tipe atau nama dari monster), ATK Power (kekuatan serangan monster), DEF Power (kekuatan pertahanan monster dari serangan), dan HP (jumlah darah monster), dan Level.

7. F06 - Potion



Potion dapat digunakan oleh user untuk membantu dalam *battle*. Terdapat Strength Potion (meningkatkan ATK Power), Resilience Potion (meningkatkan DEF Power), dan Healing Potion (mengisi darah).

8. F07 - Inventory

*Command* ini menunjukkan barang-barang apa saja yang dimiliki oleh user, seperti jumlah/jenis potion, informasi monster, jumlah OC, ataupun monster ball. Pada program ini, user juga dapat melihat informasi barang dengan lebih detail dengan cara mengetik id dari barang tersebut.

9. F08 - Battle

*Command* ini akan membuat user bertarung melawan monster dalam *turn-based 1v1 game*. Monster akan muncul secara random, dan user akan memilih monster yang akan digunakan. Setiap putaran, user bisa memilih untuk menyerang, menggunakan potion, atau keluar, sedangkan monster hanya bisa menyerang. User akan menang jika HP dari monster musuh 0, dan jika menang user akan mendapatkan OC dengan jumlah acak. User akan kalah jika HP monster user 0.

10. F09 - Arena

*Command* ini digunakan untuk melatih monster-monster yang sudah dimiliki oleh user. Monster user dilatih dengan melawan monster musuh dalam 5 stage. Mekanisme pertarungan sama seperti program battle. Sesi latihan akan selesai jika user berhasil memenangkan kelima stage atau jika HP monster user mencapai 0. Jika sesi latihan sudah selesai, user akan mendapatkan pesan "Game Over" dan informasi tentang total hadiah yang diterima, berhasil menang hingga stage berapa, total damage yang diberikan, dan total damage yang diterima.

11. F10 - Shop & Currency

*Command* ini digunakan untuk menampilkan dan membeli potion atau monster dengan OC.

12. F11 - Laboratory

*Command* ini digunakan untuk meng-*upgrade* monster yang sudah dimiliki di inventory dengan menggunakan OC.

13. F12 - Shop Management

*Command* ini hanya dapat diakses dengan user yang memiliki role admin. *Command* ini digunakan untuk mengatur (menambah/mengubah/menghapus) barang-barang yang akan dijual untuk para agent.

14. F13 - Monster Management

*Command* ini hanya dapat diakses dengan user yang memiliki role admin. *Command* ini digunakan untuk mengatur monster di dalam “owca-dex”.

15. F14 - Load

*Command* ini digunakan untuk melanjutkan *progress* dari terakhir kali user menyimpan data.

16. F15 - Save

*Command* ini digunakan untuk menyimpan *progress* user ke dalam file. Program ini akan meminta nama folder yang akan digunakan sebagai tempat penyimpanan file.

17. F16 - Exit

*Command* ini digunakan jika user ingin keluar dari permainan. User akan ditawarkan untuk melakukan save terlebih dahulu sebelum melakukan Exit.

## DAFTAR PEMBAGIAN TUGAS

<b>Fitur</b>	<b>Implementasi</b>	<b>NIM Desainer</b>	<b>NIM Coder</b>	<b>NIM Tester</b>
F00-Random Number Generator	19623005	19623005	19623005	19623005
F01-Register	16523255	16523255	16523255	16523255
F02-Login	16523155	16523155	165232155	165232155
F03-Logout	19623205	19623205	19623205	19623205
F04-Menu & Help	16523005	16523005	16523005	16523005
F05-Monster	19623005	19623005	19623005	19623005
F06-Potion	19623005	19623005	19623005	19623005
F07-Inventory	19623005	19623005	19623005	19623005
F08-Battle	19623205	19623205	19623205	19623205
F09-Arena	19623205	19623205	19623205	19623205
F10-Shop & Currency	16523005	16523005	16523005	16523005
F11-Laboratory	16523005	16523005	16523005	16523005
F12-Shop Management	19623005	19623005	19623005	19623005
F13-Monster Management	19623005	19623005	19623005	19623005
F14-Load	16523155	16523155	165232155	165232155
F15-Save	16523155	16523155	165232155	165232155

F16-Exit	16523155	16523155	165232155	165232155
B01-Git Best Practice	19623005	19623005	19623005	19623005
	19623205	19623205	19623205	19623205
	16523005	16523005	16523005	16523005
	16523155	16523155	16523155	16523155
	16523255	16523255	16523255	16523255
B02-Typing	19623005	19623005	19623005	19623005
	19623205	19623205	19623205	19623205
	16523005	16523005	16523005	16523005
	16523155	16523155	16523155	16523155
	16523255	16523255	16523255	16523255
Laporan	-	1652355	-	-

**Tabel 1** Daftar Pembagian Tugas

## CHECKLIST HASIL RANCANGAN

<b>Fitur</b>	<b>Desain</b>	<b>Implementasi</b>	<b>Testing</b>
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
B01-Git Best Practice	-	V	-
B02-Typing	-	V	-
B03-Monster Ball	X	X	X
B04-JACKPOT	X	X	X
B05-Peta Kota Danville	X	X	X

**Tabel 2** Checklist Hasil Rancangan

## DESAIN *COMMAND*

### 1. **F01- Register**

Nama command : register  
Masukan : username, password, monster\_pilihan  
Keluaran : pesan tentang keberhasilan register

### 2. **F02 - Login**

Nama Command : login  
Masukan : username, password  
Keluaran : pesan tentang keberhasilan login

### 3. **F03 - Logout**

Nama Command : logout  
Masukan : -  
Keluaran : pesan tentang keberhasilan logout

### 4. **F04 - Menu & Help**

Nama Command : help  
Masukan : -  
Keluaran : pesan tentang apa yang sebaiknya dilakukan oleh user

### 5. **F07 - Inventory**

Nama Command : inventory  
Masukan : id  
Keluaran : informasi detail tentang barang dengan id yang dimasukan

### 6. **F08 - Battle**

Nama Command : battle  
Masukan : choiceMonster, actionTurn, choicePotion  
Keluaran : UI pertarungan dengan monster, pilihan monster, pilihan action

### 7. **F09 - Arena**

Nama Command : arena  
Masukan : choiceMonster, actionTurn, choicePotion,  
Keluaran : UI pertarungan dengan monster, pilihan monster, pilihan action

## **8. F10 - Shop & Currency**

Nama Command : shop  
Masukan : action, option, idMonster, idPotion, quantity,  
Keluaran : UI shop

## **9. F11 - Laboratory**

Nama Command : laboratory  
Masukan : pilihMonster, confirm  
Keluaran : UI untuk meng-upgrade monster

## **10. F12 - Shop Management**

Nama Command : shop  
Masukan : action, option, newIdMonster, stock, price, newIdPotion,  
updateIdMonster,  
Keluaran : UI untuk mengatur isi dari shop

## **11. F13 - Monster Management**

Nama Command : monster  
Masukan : action  
Keluaran : UI untuk mengatur monster dalam “owca-dex”

## **12. F14 - Load**

Nama Command : python main.py <nama\_folder>  
Masukan : nama\_folder  
Keluaran : pesan tentang keberhasilan load file

## **13. F15 - Save**

Nama Command : save  
Masukan : folder\_name  
Keluaran : pesan tentang keberhasilan nge-save data

## **14. F16 - Exit**

Nama Command : exit  
Masukan : response  
Keluaran : konfirmasi untuk keluar dari program



# DESAIN KAMUS DATA

## 1. F00 - Random Number Generator

```
function log (a,b,c : integer, seed : real) -> Integer
{mendapatkan angka acak}

function getRandomNumber ([,]) -> Integer
{mendapatkan angka acak dalam range tertentu}
```

## 2. F01- Register

```
function username_valid (input username: string) -> Boolean
{memvalidasi username yang diinputkan}

function username_unik (input username: string, dataUser : list) -> Boolean
{memvalidasi apakah username yang diinputkan unik}

function isIdExist (input monster_pilihan : integer, dataMonster : list) -> Boolean
{memvalidasi apakah id yang diinputkan ada dalam file data monster}

procedure register (dataUser : list, isLogin : Boolean, input username : string, dataMonster :
list, dataMonsterInventory : list)
{I.S memvalidasi apakah isLogin True atau False
F.S menghasilkan pesan tentang keberhasilan register}
```

## 3. F02 - Login

```
function getDataByUsername (newUsername : string, dataUser : list) -> string
{mengembalikan username yang sudah berhasil melakukan login}

function isUsernameExist (newUsername : string, dataUser : list) -> Boolean
{memvalidasi jika username sudah tersimpan dalam file data user}

function isValidPassword (newUsername : string, password : string, dataUser: list) -> Boolean
```

{memvalidasi apakah password sudah valid}

**function** login (isLogin : Boolean, dataUser : list, userId : integer, input username : string, coin : integer, role : string) -> (Boolean, integer, string, integer, string)  
{memvalidasi username dan password yang diinputkan dan menetapkan role sebagai “agent” dan coin sebagai 0}

#### 4. F03 - Logout

**function** logout (isLogin : Boolean, userId : Integer, username: string, coin : Integer, role : string) -> (Boolean, Integer, string, Integer, String)  
{mengembalikan pesan tentang keberhasilan Logout}

#### 5. F04 - Menu & Help

**procedure** help (isLogin : Boolean, role : string, username : string)  
{I.S isLogin dapat True atau False  
F.S memberikan pesan tentang apa yang sebaiknya dilakukan oleh user}

#### 6. F07 - Inventory

**procedure** inventory (dataMonster : list, dataItemInventory : list, dataMonsterInventory : list, userId : string, coin : int)  
{I.S list Inventory kosong  
F.S menampilkan UI inventory user}

#### 7. F08 - Battle

**function** attributeByLevel (level : integer, atkPower, defPower, hp : integer) → (integer, integer, integer)  
{mengembalikan atkPower, defPower, hp sesuai level monster}

**procedure** statusMonster(name : string, atkPower, defPower, hp, level : integer)  
{I.S name, atkPower, defPower, hp, level

F.S menampilkan status dari monster}

**function** getMonsterAgent(dataMonsterInventory, dataMonster : array of record, userId : integer) → list

{mengembalikan list monster yang dimiliki user}

**procedure** displayMonsterAgent(allMonsterAgent : list)

{I.S x = 1

F.S menghasilkan tampilan list monster yang dimiliki oleh user}

**procedure** displayPotionAgent(allPotionAgent : list)

{I.S validasi apakah allPotionAgent kosong

F.S menampilkan potion yang dimiliki oleh user}

**function** attack(name : string, atkPower : integer, nameRival : string, defPowerRival, atkPowerRival, hpRival, levelRival : integer) → integer

{menyerang monster musuh dan mengembalikan hp monster musuh setelah diserang}

**function** battle (userId : integer, username : string, dataMonster, dataMonsterInventory, dataItemInventory : array of record, coin : integer) → (integer, list)

{mengembalikan jumlah coin, dan potion yang dimiliki}

## 8. F09 - Arena

**procedure** displayMonsterAgent(allMonsterAgent : list, x : Integer)

{I.S x = 1

F.S menghasilkan tampilan list monster yang dimiliki oleh user}

**function** arena (username : string, dataMonster : list, allPotionAgent : list, coin : Integer) -> (Integer, list)

{menampilkan UI pertarungan dengan monster}

## 9. F10 - Shop & Currency

**function** isMonsterExistInInventory (userId : integer, idMonster : integer, dataMonsterInventory : list) -> Boolean  
{mengecek apakah monster ada di dalam inventory}

**function** isPotionExistInInventory (userId : integer, namePotion : string, dataItemInventory : list) -> Boolean  
{mengecek apakah potion ada dalam inventory}

**procedure** shop (dataMonster : list, dataMonsterShop : list, dataItemShop : list, dataMonsterInventory : list, dataItemInventory : list, coin : integer, userId : integer)  
{I.S kondisi True  
F.S menghasilkan UI shop}

## 10. F11 - Laboratory

**function** laboratory (dataMonsterInventory : list, dataMonster : list, userId : integer, coin : integer) -> Integer  
{meng-upgrade monster dan mengembalikan jumlah coin yang baru}

## 11. F12 - Shop Management

**function** getMonsterNotInShop (dataMonsterShop : list, dataMonster : list) -> list  
{Jika tidak ditemukan id monster pada monster\_shop, maka akan ditambahkan ke dalam list monsterNotInShop}

**function** getPotionNotInShop (dataItemShop : list, dataPotion : list) -> list  
{Mendapatkan data potion yang tidak terdapat di shop}

**procedure** displayMonsterShop (dataMonsterShop : list, dataMonster : list)  
{Menampilkan monster yang terdaftar di Shop}

**procedure** displayPotionShop (dataItemShop : list)  
{Menampilkan potion yang terdaftar di Shop}

```
procedure shopManagement (dataItemShop : list, dataMonsterShop : list, dataMonster : list)
{I.S kondisi True
F.S menampilkan UI untuk mengatur shop}
```

## 12. F13 - Monster Management

```
function isNum (x : string) -> Boolean
{mengembalikan True atau False}

function maxIDMonster (dataMonster : list) -> Integer
{mengembalikan id maximum monster dari file monster.csv}

function foundMonster (dataMonster : list, newMonster : string) -> Boolean
{mengembalikan True atau False}

procedure displayMonster (dataMonster : list)
{menampilkan informasi monster}

procedure addMonster (dataMonster : list)
{I.S memvalidasi masukan monster type, ATK power, DEF power, HP
F.S menampilkan informasi monster baru}

procedure monsterManagement (dataMonster : list)
{menampilkan UI untuk mengatur monster}
```

## 13. F14 - Load

```
function read_CSV (filename : string, folder_path : string) -> list
{membaca file CSV}

procedure load_data(folder : string)
{ I.S.: Nama file CSV (filename) dan path folder (folder_path) terdefinisi }
{ F.S.: Data dari file CSV dibaca dan disimpan dalam bentuk matriks }
```

```
function main_load ()  
{ngeload data file ke dalam program}
```

#### 14. F15 - Save

```
procedure writeCSV (folder_path : string, csv_file : string, array_data : list)  
{I.S.: Path folder (folder_path), nama file CSV (csv_file), dan data dalam bentuk array 2D  
(array_data) terdefinisi}  
{ F.S.: menulis data yang terdapat dalam file CSV}  
  
procedure save (dataUser : list, dataMonster : list, dataMonsterInventory : list,  
dataItemInventory : list, dataMonsterShop : list, dataItemShop : list)  
{I.S.: Path folder (folder_path), nama file CSV (csv_file), dan data dalam bentuk array 2D  
(array_data) terdefinisi}  
{F.S.: Data disimpan dalam file CSV di path folder dengan nama yang diberikan}
```

#### 15. F16 - Exit

```
procedure exit (dataUser : list, dataMonster : list, dataMonsterInventory : list,  
dataItemInventory : list, dataMonsterShop : list, dataItemShop : list)  
{I.S kondisi True dan semua array terdefinisi  
F.S keluar dari program}
```

# **DESAIN DEKOMPOSISI ALGORITMIK DAN FUNGSIONAL**

▢ DESAIN DEKOMPOSISI ALGORITMIK DAN FUNGSIONAL

# SPEKIFIKASI

## 1. F00 - Random Number Generator

**Program** Random Number Generator

{RNG akan menghasilkan bilangan acak dari sebuah *range* angka}

**function** lcg (a,b,c : integer, seed : real) -> Integer

{men

### KAMUS

a,b,c : integer

seed : real

### ALGORITMA

a <- 1103515245

c <- 12345

m <- 2\*\*31

seed <- time()\*1000

return int((a\*seed+c) % m)

**function** getRandomNumber ([,]) -> Integer

{mendapatkan angka acak dalam range tertentu}

### KAMUS LOKAL

a,c,m,xn : integer

### ALGORITMA

a <- 1103515245

c <- 12345

m <- 2\*\*31

xn <- (a\*lcg()+c) % m

if range is None then

    range <- [0,m-1]

return int((xn/(m-1)) \* (range[1] - range[0]) + range[0])



## 2. F01- Register

### Program Register

{*Command* Register digunakan agar user dapat melakukan registrasi awal. Jika username dan password sudah valid, maka user akan memilih monster awal, mendapatkan *role* agent, dan 0 OC}

### KAMUS

{Spesifikasi fungsi dan prosedur}

**procedure** register (dataUser : list, isLogin : Boolean, input username : string, dataMonster : list, dataMonsterInventory : list)

{I.S memvalidasi apakah isLogin True atau False

F.S menghasilkan pesan tentang keberhasilan register}

### ALGORITMA PROGRAM UTAMA

If isLogin == True then

    Output ('Register gagal')

    Output ('Anda telah login dengan username',username,'silakan lakukan "LOGOUT" sebelum melakukan register.\n')

Else

    {Validasi username dan password}

    Repeat

        Output (

"=====

=====")

    Output( "        )

    Output( "        )

    Output(

"\n=====

=====")

        {Input username dan password}

input (username)

Input (password)

Depend on (username, password, username\_valid, username\_unik)

Not username : output ('username tidak boleh kosong')

Not password : output ('password tidak boleh kosong')

username\_valid(username) == False : output('Username hanya boleh berisi alfabet, angka, underscore, dan strip')

username\_unik(username, dataUser) == False : output('Username', username, 'sudah terpakai, silakan gunakan username lain!')

Else

Break

{Pilih Monster}

Output ('silakan pilih salah satu monster sebagai monster awalmu!')

{Menampilkan list monster}

Monster traversal [dataMonster]

Output (f' {monster[0]}. {monster[1]}')

{Validasi Input pilihan monster}

Repeat

input(monster\_pilihan)

If isIdExist(monster\_pilihan,dataMonster) == True then

{Menambahkan data user baru}

userId = newId(dataUser)

newUser = [str(userId),username,password,'agent','0']

dataUser.append(newUser)

{Menambahkan data monster inventory user}

dataMonsterInventory.append([str(userId),monster\_pilihan,'1'])

```

        output(f'\nSelamat datang Agent {username}. Mari kita mengalahkan
        Dr. Asep Spakbor dengan {getDataById(monster_pilihan, dataMonster)[1]}!')
        Break
    Else
        output('Pilihan Tidak Valid')

```

**function** username\_valid (input username: string) -> Boolean  
 {memvalidasi username yang diinputkan}

#### **KAMUS LOKAL**

username : string

#### **ALGORITMA**

```

char traversal [username]
if not(97<=ord(char)<=122 or 65<=ord(char)<=90 or 48<=ord(char)<=57 or ord(char)==95 or
ord(char)==45) then
    return False
return True

```

**function** username\_unik (input username: string, dataUser : list) -> Boolean  
 {memvalidasi apakah username yang diinputkan unik}

#### **KAMUS LOKAL**

username : string

dataUser : list

#### **ALGORITMA**

```

user traversal [dataUser]
    if user[1] == username then
        return False
return True

```

**function** isIdExist (input monster\_pilihan : integer, dataMonster : list) -> Boolean  
 {memvalidasi apakah id yang diinputkan ada dalam file data monster}

### **KAMUS LOKAL**

monster\_pilihan : integer

dataMonster : list

### **ALGORITMA**

Monster traversal [dataMonster]:

    if monster[0] == monster\_pilihan then

        return True

    return False

## **3. F02 - Login**

### **Program Login**

{Jika sudah melakukan registrasi, maka user dapat menggunakan *command* Login untuk *login* ke dalam sistem}

### **KAMUS**

{Spesifikasi fungsi dan prosedur}

**function** getDataByUsername (newUsername : string, dataUser : list) -> string

{mengembalikan username yang sudah berhasil melakukan login}

**function** isUsernameExist (newUsername : string, dataUser : list) -> Boolean

{memvalidasi jika username sudah tersimpan dalam file data user}

**function** isValidPassword (newUsername : string, password : string, dataUser: list) -> Boolean

{memvalidasi apakah password sudah valid}

**function** login (isLogin : Boolean, dataUser : list, userId : integer, input username : string, coin : integer, role : string) -> (Boolean, integer, string, integer, string)

{memvalidasi username dan password yang diinputkan dan menetapkan role sebagai “agent” dan coin sebagai 0}

{ Deklarasi variabel }

isLogin : boolean

userId : integer

username : string

coin : integer

role : string

newUsername : string

password : string

user : array of string

userNow : array of string

### ALGORITMA

If isLogin = False then

Output

"=====

=====

Output "\n ████████ ████████ ████████ ████████"

Output " ████████ ████████ ████████ ████████"

Output

"\n=====

=====

// Validasi Username dan Password

While True do

newUsername ← Input("Username: ")

password ← Input("Password: ")

If isUsernameExist(newUsername, dataUser) = False then

Output "\n'Username tidak terdaftar!\n"

Else

If isValidPassword(newUsername, password, dataUser) = False then

Output "\n'Password salah!\n"

Lainnya

isLogin ← True

userNow ← getDataByUsername(newUsername, dataUser)

userId ← ConvertToInteger(userNow[0])

username ← userNow[1]

role ← userNow[3]

coin ← ConvertToInteger(userNow[4])

Output "Selamat datang kembali, ", Capitalize(role), " ", username, "! Mari lanjutkan petualangan kita!"

Output "Masukkan command 'help' untuk daftar command yang dapat kamu panggil.\n"

Return isLogin, userId, username, coin, role

Else

Output "Login gagal!"

Output "Anda telah login dengan username ", username, ", silahkan lakukan 'LOGOUT' sebelum melakukan login kembali.\n"

Return isLogin, userId, username, coin, role

#### 4. F03 - Logout

##### **Program** Logout

{Jika sudah melakukan *login*, user dapat keluar dari akun yang sedang dipakai dengan *command* Logout. Setelah melakukan *logout*, user tidak bisa mengakses isi dari akun sebelumnya}

##### **KAMUS**

{Spesifikasi fungsi dan prosedur}

**function** logout (isLogin : Boolean, userId : Integer, username: string, coin : Integer, role : string) -> (Boolean, Integer, string, Integer, String)

{mengembalikan pesan tentang keberhasilan Logout}

{ Deklarasi variabel }

isLogin : boolean

userId : integer

username : string

coin : integer

role : string

### **ALGORITMA**

if (isLogin = true) then

    output("Anda Berhasil Logout dari Akun dengan username ", username, ".\n")

    isLogin ← false

    userId ← 0

    username ← ""

    coin ← 0

    role ← ""

else

    output("Logout gagal!")

    output("Anda belum login, silahkan lakukan 'LOGIN' terlebih dahulu sebelum melakukan logout.\n")

→ isLogin, userId, username, coin, role

## **5. F04 - Menu & Help**

### **Program Menu & Help**

{menghasilkan pesan bantuan tentang apa yang dapat dilakukan selanjutnya}

### **KAMUS**

isLogin : Boolean

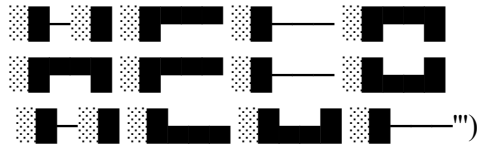
role : string

username : string

### **ALGORITMA**

output('\n=====')

output(""



```
output("\n=====")
```

```
if isLogin == False then
```

```
    output("\nKamu belum login sebagai role apapun. Silahkan login terlebih dahulu.")
```

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

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

```
else:
```

```
    if role == 'agent' then
```

```
        output(f'\nHalo Agent {username}. Kamu memanggil command HELP. Kamu  
memilih jalan yang benar, semoga kamu tidak sesat kemudian.')
```

```
        output('Berikut adalah hal-hal yang dapat kamu lakukan sekarang:\n')
```

```
        output("\t1. LOGOUT: Keluar dari akun yang sedang digunakan')
```

```
        output("\t2. INVENTORY: Melihat owca-dex yang dimiliki oleh Agent')
```

```
        output("\t3. BATTLE: Bertarung melawan monster secara random')
```

```
        output("\t4. ARENA: Pertarungan dengan aturan seperti battle selama 5 stage')
```

```
        output("\t5. SHOP: Tempat Agent membeli monster dan potion menggunakan  
owca-coin')
```

```
        output("\t6. LABORATORY: Tempat Agent melakukan upgrade level monster  
yang dimiliki di inventory')
```

```
    else: {jika role == 'admin'}
```

```
        output(f'\nSelamat datang, Admin {username}. Berikut adalah hal-hal yang  
dapat kamu lakukan:\n')
```

```
        output("\t1. LOGOUT: Keluar dari akun yang sedang digunakan')
```

```
        output("\t2. SHOP: Melakukan manajemen pada SHOP sebagai tempat jual beli  
perlatan Agent')
```

```
        output("\t3. MONSTER: Mengatur manajemen monster di dalam owca-dex')
```

```
{nge-print Footnote}
```

```
output("\nFootnote:')
```

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



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

## 6. F07 - Inventory

### **Program** Inventory

{program untuk menampilkan menampilkan UI inventory user}

### **KAMUS**

{spesifikasi fungsi dan prosedur}

**procedure** inventory (dataMonster : list, dataItemInventory : list, dataMonsterInventory : list,

userId : string, coin : int)

{I.S list Inventory kosong

F.S menampilkan UI inventory user}

{ Deklarasi tipe dan variabel }

-dataMonster : list of lists

-dataItemInventory : list of lists,

-dataMonsterInventory : list of lists

-userId : string

-coin : integer

-listInventory : list of lists

-idx : integer

-monster : list

-item : list

-idMonster : string

-level : string

-infoMonster : list,

-id : string,

-detailItem : list

### **ALGORTIMA**

{read csv}

read ('data/item\_inventory.csv',dataItemInventory)

```

read('data/item_inventory.csv',dataMonsterInventory)
read('data/monster.csv',dataMonster)

userId ← '2'
coin ← '9999'

{algoritma utama}

procedure inventory(dataMonster: list, dataItemInventory: list, dataMonsterInventory: list,
userId: string, coin: int) → void

    output("===== INVENTORY LIST (User ID: " + userId + ") =====")
    output("Jumlah O.W.C.A Coin-mu sekarang " + coin + ".")
    listInventory ← [ ]
    idx ← 1

    for each monster in dataMonsterInventory do
        if monster[0] = userId then
            idMonster ← monster[1]
            level ← monster[2]
            append(listInventory, ['Monster', idMonster, level])
            infoMonster ← getDataById(idMonster, dataMonster)
            output(idx + ". Monster (Name: " + infoMonster[1] + ", Lvl: " + level + ", HP: " +
infoMonster[4] + ")")
            idx ← idx + 1

    for each item in dataItemInventory do
        if item[0] = userId and item[1] ≠ 'monster_ball' then
            append(listInventory, ['Potion', item[1], item[2]])
            output(idx + ". Potion (Type: " + capitalize(item[1]) + ", Qty: " + item[2] + ")")
            idx ← idx + 1

    for each item in dataItemInventory [1:] do

```

```
if item[0] = userId and item[1] = 'monster_ball' then
  append(listInventory, ['Monster Ball', item[1], item[2]])
  output(idx + ". Monster Ball (Qty: " + item[2] + ")")
  idx ← idx + 1
```

iterate

```
output("Ketikkan id untuk menampilkan detail item :")
output("(Ketik 'KELUAR' jika ingin keluar dari INVENTORY)")
id ← toUpperCase(input('>>> '))
```

stop (id = 'KELUAR')

```
if not isNum(id) or (toInt(id) < 1 or toInt(id) > length(listInventory)) then
  output("id tidak valid!")
```

else

```
detailItem ← listInventory[toInt(id) - 1]
```

```
output(detailItem[0])
```

```
depends on (detailItem[0])
```

```
idetailItem[0] = 'Monster':
```

```
idMonster ← detailItem[1]
```

```
infoMonster ← getDataById(idMonster, dataMonster)
```

```
output("Name      : " + infoMonster[1])
```

```
output("ATK Power : " + infoMonster[2])
```

```
output("DEF Power : " + infoMonster[3])
```

```
output("HP       : " + infoMonster[4])
```

```
output("Level    : " + detailItem[2])
```

```
detailItem[0] = 'Potion' :
```

```
output("Type      : " + capitalize(detailItem[1]))
```

```
output("Quantity : " + detailItem[2])
```

else

```
output("Quantity : " + detailItem[2])
```

```
inventory(dataMonster, dataItemInventory, dataMonsterInventory, userId, coin)
```

## 7. F08 - Battle

### **Program Battle**

{Mengelola proses battle antara monster agent dan monster musuh}

### **KAMUS**

{Spesifikasi fungsi dan prosedur}

**function** attributeByLevel (level : integer, atkPower, defPower, hp : integer) → (integer, integer, integer)

{mengembalikan atkPower, defPower, hp sesuai level monster}

**procedure** statusMonster(name : string, atkPower, defPower, hp, level : integer)

{I.S name, atkPower, defPower, hp, level

F.S menampilkan status dari monster}

**function** getMonsterAgent(dataMonsterInventory, dataMonster : array of record, userId : integer) → list

{mengembalikan list monster yang dimiliki user}

**procedure** displayMonsterAgent(allMonsterAgent : list)

{I.S x = 1

F.S menghasilkan tampilan list monster yang dimiliki oleh user}

**procedure** displayPotionAgent(allPotionAgent : list)

{I.S validasi apakah allPotionAgent kosong

F.S menampilkan potion yang dimiliki oleh user}

**function** attack(name : string, atkPower : integer, nameRival : string, defPowerRival, atkPowerRival, hpRival, levelRival : integer) → integer

{menyerang monster musuh dan mengembalikan hp monster musuh setelah diserang}

**function** battle (userId : integer, username : string, dataMonster, dataMonsterInventory, dataItemInventory : array of record, coin : integer) → (integer, list)

{mengembalikan jumlah coin, dan potion yang dimiliki}

{ Deklarasi tipe dan variabel }

type Monster : <name: string, atkPower: integer, defPower: integer, hp: integer, level: integer>

type Potion : <name: string, quantity: integer>

type Inventory : array of Potion

userId : integer

username : string

dataMonster : array of Monster

dataMonsterInventory : array of array of string

dataItemInventory : array of array of string

coin : integer

allPotionAgent : Inventory

enemy : Monster

nameEnemy : string

atkPowerEnemy : integer

defPowerEnemy : integer

hpEnemy : integer

levelEnemy : integer

allMonsterAgent : array of Monster

choiceMonster : integer

monsterAgent : Monster

nameMonsterAgent : string

atkPowerMonsterAgent : integer

defPowerMonsterAgent : integer

hpMonsterAgent : integer

levelMonsterAgent : integer

turn : integer

countStrength : integer

countResilience : integer

countHealing : integer

actionTurn : integer

choicePotion : integer

## ALGORITMA

**procedure** attributeByLevel(level : integer, atkPower, defPower, hp : integer) → (integer, integer, integer)

if level  $\neq$  1 then

    atkPower  $\leftarrow$  atkPower + int(((level-1) \* 10 / 100) \* atkPower)

    defPower  $\leftarrow$  defPower + int(((level-1) \* 10 / 100) \* defPower)

    hp  $\leftarrow$  hp + int(((level-1) \* 10 / 100) \* hp)

→ (atkPower, defPower, hp)

**procedure** statusMonster(name : string, atkPower, defPower, hp, level : integer)

output("Name : " + name)

output("ATK Power : " + str(atkPower))

output("DEF Power : " + str(defPower))

output("HP : " + str(hp))

output("Level : " + str(level))

**procedure** getMonsterAgent(dataMonsterInventory, dataMonster : array of record, userId : integer) → array of record

AllMonsterAgent  $\leftarrow$  [ ]

for each monster in dataMonsterInventory do

    if monster.UserId = userId then

        MonsterId  $\leftarrow$  monster.MonsterId

        Level  $\leftarrow$  monster.Level

        InfoMonster  $\leftarrow$  getDataById(MonsterId, dataMonster)

        Name  $\leftarrow$  InfoMonster.Name

        AtkPower  $\leftarrow$  InfoMonster.AtkPower

        DefPower  $\leftarrow$  InfoMonster.DefPower

        Hp  $\leftarrow$  InfoMonster.Hp

        append(AllMonsterAgent, (Name, AtkPower, DefPower, Hp, Level))

→ AllMonsterAgent

**procedure** displayMonsterAgent(allMonsterAgent : array of record)

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

x ← 1

for each monster in allMonsterAgent do

output(str(x) + ". " + monster.Name)

x ← x + 1

**procedure** getPotionAgent(dataItemInventory : array of record, userId : integer) → array of record

AllPotionAgent ← [ ]

for each item in dataItemInventory do

if item.UserId = userId and item.Name ≠ 'monster\_ball' then

append(AllPotionAgent, (item.Name, item.Quantity))

→ AllPotionAgent

**procedure** displayPotionAgent(allPotionAgent : array of record)

if length(allPotionAgent) = 0 then

output("Anda tidak memiliki Potion dalam inventory!")

else

output("===== POTION LIST =====")

x ← 1

for each potion in allPotionAgent do

depends on potion[0] do

case 'strength': utility ← 'Increases ATK Power'

case 'resilience': utility ← 'Increases DEF Power'

Else

utility ← 'Restores Health'

output(str(x) + ". " + capitalize(potion.Name) + " Potion (Qty: " + str(potion.Quantity) + ")  
- " + Utility)

x ← x + 1

output(str(x) + ". Cancel")

```

procedure attack(name : string, atkPower : integer, nameRival : string, defPowerRival,
atkPowerRival, hpRival, levelRival : integer) → integer
    output("SHEESHHHHH, " + name + " menyerang " + nameRival + " !!!")
    att ← getRandomNumber([0.7 * atkPower, 1.3 * atkPower])
    attPercent ← (att - atkPower) / atkPower * 100
    reducedBy ← defPowerRival / 100 * att
    attResult ← att - reducedBy
    hpRival ← floor(hpRival - attResult)
    if hpRival < 0 then
        hpRival ← 0
    statusMonster(nameRival, atkPowerRival, defPowerRival, hpRival, levelRival)
    output("Penjelasan: ATT: " + str(att) + " (" + str(attPercent) + "%), Reduced By: " +
str(reducedBy) + " (" + str(defPowerRival) + "%), ATT Results: " + str(attResult))
    → hpRival

```

```

procedure battle(userId : integer, username : string, dataMonster, dataMonsterInventory,
dataItemInventory : array of record, coin : integer) → (integer, array of record)
    AllPotionAgent ← getPotionAgent(dataItemInventory, userId)
    Enemy ← dataMonster[getRandomNumber([1, length(dataMonster)])]
    NameEnemy ← Enemy.Name
    AtkPowerEnemy ← Enemy.AtkPower
    DefPowerEnemy ← Enemy.DefPower
    HpEnemy ← Enemy.Hp
    LevelEnemy ← getRandomNumber([1, 5])
    (AtkPowerEnemy, DefPowerEnemy, HpEnemy) ← attributeByLevel(LevelEnemy,
AtkPowerEnemy, DefPowerEnemy, HpEnemy)

```

```

output("=====
=====")
output("  █ █ █ █ █ █ █")
output("  █ █ █ █ █ █ █")

```



```

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

output("RAWRRR, Monster " + NameEnemy + " telah muncul !!!")
statusMonster(NameEnemy, AtkPowerEnemy, DefPowerEnemy, HpEnemy, LevelEnemy)

AllMonsterAgent ← getMonsterAgent(dataMonsterInventory, dataMonster, userId)
displayMonsterAgent(AllMonsterAgent)

iterate
  output("Pilih monster untuk bertarung: ")
  read(choiceMonster)
  if isNum(choiceMonster) = false then
    output("Pilihan tidak valid!")
  else
    stop  $1 \leq \text{int}(\text{choiceMonster}) \leq \text{length}(\text{allMonsterAgent})$ 
    output("Pilihan nomor tidak tersedia!")

MonsterAgent ← AllMonsterAgent[ChoiceMonster - 1]
NameMonsterAgent ← MonsterAgent.Name
AtkPowerMonsterAgent ← MonsterAgent.AtkPower
DefPowerMonsterAgent ← MonsterAgent.DefPower
HpMonsterAgent ← MonsterAgent.Hp
LevelMonsterAgent ← MonsterAgent.Level
    (AtkPowerMonsterAgent,    DefPowerMonsterAgent,    HpMonsterAgent) ←
attributeByLevel(LevelMonsterAgent,    AtkPowerMonsterAgent,    DefPowerMonsterAgent,
HpMonsterAgent)
  output("RAWRRR, Agent " + username + " mengeluarkan Monster " + NameMonsterAgent +
" !!!")
    statusMonster(NameMonsterAgent,    AtkPowerMonsterAgent,    DefPowerMonsterAgent,
HpMonsterAgent, LevelMonsterAgent)

Turn ← 1

```

CountStrength  $\leftarrow$  0

CountResilience  $\leftarrow$  0

CountHealing  $\leftarrow$  0

iterate

output("TURN " + str(Turn) + " (" + NameMonsterAgent + ")")

output("1. Attack")

output("2. Use Potion")

output("3. Quit")

validAction  $\leftarrow$  false

iterate

input(actionTurn)

if isNum(actionTurn) = False then

output('Perintah tidak valid!')

else

stop actionTurn = '1' or actionTurn = '2' or actionTurn = '3'

output("Perintah tidak valid!")

depends on (actionTurn)

actionTurn = 1 :

HpEnemy  $\leftarrow$  attack(NameMonsterAgent, AtkPowerMonsterAgent, NameEnemy, DefPowerEnemy, AtkPowerEnemy, HpEnemy, LevelEnemy)

If hpEnemy = 0 then

output("Selamat, Anda berhasil mengalahkan monster " + nameEnemy + " !!!")

rewardCoin  $\leftarrow$  getRandomNumber([5, 30])

coin  $\leftarrow$  coin + rewardCoin

output("Total OC yang diperoleh: " + str(rewardCoin))

stop

else

```

        output("TURN " + str(turn) + " (" + nameEnemy + ")")
        hpMonsterAgent ← attack(nameEnemy, atkPowerEnemy, nameMonsterAgent,
defPowerMonsterAgent, atkPowerMonsterAgent, hpMonsterAgent, levelMonsterAgent)

        If (hpMonsterAgent = 0) then
            output("Yahhh, Anda dikalahkan monster " + nameEnemy + ". Jangan menyerah,
coba lagi !!!")
            stop

        turn ← turn + 1

    actionTurn = 2 :
        displayPotionAgent(allPotionAgent)

    iterate
        choicePotion ← input("Pilih perintah: ")
        if isNum(choicePotion) = false then
            output("Perintah tidak valid!")
        else
            depends on (int(choicePotion),len(allPotionAgent))
            1<=int(choicePotion)<=len(allPotionAgent)
            if allPotionAgent[int(choicePotion)-1][1] = 0 then
                output ("Maaf {allPotionAgent[int(choicePotion)-1][0].capitalize()} Potion
kamu telah habis, silahkan beli lagi di SHOP!")
                stop

            depends on (allPotionAgent[int(choicePotion)-1][0])
            allPotionAgent[int(choicePotion)-1][0] == 'strength':
                if countStrength = 1 then
                    output("Kamu mencoba memberikan ramuan ini kepada " +
nameMonsterAgent + ", namun dia menolaknya seolah-olah dia memahami ramuan tersebut
sudah tidak bermanfaat lagi.")
                else
                    output("Setelah meminum ramuan ini, aura kekuatan terlihat
mengelilingi " + nameMonsterAgent + " dan gerakannya menjadi lebih cepat dan mematikan.")

```

```

allPotionAgent[int(choicePotion) - 1][1] ←
allPotionAgent[int(choicePotion) - 1][1] - 1
countStrength ← countStrength + 1
atkPowerMonsterAgent ← atkPowerMonsterAgent + 0.05 *
atkPowerMonsterAgent

allPotionAgent[int(choicePotion)-1][0] == 'resilience':
    if countResilience = 1 then
        output("Kamu mencoba memberikan ramuan ini kepada " +
nameMonsterAgent + ", namun dia menolaknya seolah-olah dia memahami ramuan tersebut
sudah tidak bermanfaat lagi.")
    else
        output("Setelah meminum ramuan ini, muncul sebuah energi
pelindung di sekitar {nameMonsterAgent} yang membuatnya terlihat semakin tangguh dan
sulit dilukai.")
        allPotionAgent[int(choicePotion) -1][1]
←allPotionAgent[int(choicePotion) - 1][1] - 1
        countResilience ← countResilience + 1
        defPowerMonsterAgent ←defPowerMonsterAgent +
0.05*defPowerMonsterAgent

    else
        if (countHealing = 1) then
            output("\nKamu mencoba memberikan ramuan ini kepada ",
nameMonsterAgent, ", namun dia menolaknya seolah-olah dia memahami ramuan tersebut
sudah tidak bermanfaat lagi.")
        else
            output("\nSetelah meminum ramuan ini, luka-luka yang ada di dalam
tubuh ", nameMonsterAgent, " sembuh dengan cepat. Dalam sekejap, Pikachu terlihat
kembali prima dan siap melanjutkan pertempuran.")
            allPotionAgent[choicePotion-1][1] ← allPotionAgent[choicePotion-1][1] -
1
            countHealing ← countHealing + 1
            hpMonsterAgent ← hpMonsterAgent + 0.25 * hpMonsterAgent
            if (hpMonsterAgent > monsterAgent[3]) then
                hpMonsterAgent ← monsterAgent[3]

```

```
        stop
        int(choicePotion) == len(allPotionAgent)+1:
            stop

    else
        Output('Perintah tidak valid!')

else
    output ('\nAnda berhasil kabur dari BATTLE!')
    stop

for item in dataItemInventory do
    for potion in allPotionAgent do
        if item[0] = str(userId) and item[1] = potion[0] then
            item[2] ← potion[1]

→ coin
```

## **8. F09 - Arena**

## **Program Arena**

{ Spesifikasi Program: mensimulasikan sebuah arena pertarungan di mana pemain dapat memilih monster mereka dan bertarung melawan monster musuh dalam beberapa tahap. }

## **KAMUS**

{ Deklarasi tipe dan variabel }

{Fungsi}

**function** readCSV(file: string) -> list

{Membaca file CSV dan mengembalikan data dalam bentuk list.}

**function** displayMonsterAgent(monsterList: list) -> void

{Menampilkan daftar monster agent yang tersedia.}

**function** displayPotionAgent(potionList: list) -> void

{Menampilkan daftar potion yang dimiliki agent}

**function** attributeByLevel(level: int, atkPower: int, defPower: int, hp: int) -> tuple

{Mengatur atribut monster berdasarkan level.}

**function** statusMonster(name: string, atkPower: int, defPower: int, hp: int, level: int) -> void

{Menampilkan status monster.}

**function** attack(attackerName: string, attackerAtk: int, defenderName: string, defenderDef: int, defenderAtk: int, defenderHp: int, defenderLevel: int) -> int

{Melakukan serangan pada monster lawan dan mengembalikan sisa HP lawan.}

**function** getRandomNumber(range: list) -> int

{Menghasilkan angka acak dalam rentang yang diberikan.}

**function** getPotionAgent(itemInventory: list, userId: int) -> list

{Mengambil daftar potion yang dimiliki agent berdasarkan ID user.}

**function** getMonsterAgent(monsterInventory: list, monsterData: list, userId: int) -> list

{Mengambil daftar monster agent berdasarkan ID user.}

**function** isNum(inputStr: string) -> bool

{Memeriksa apakah input adalah angka.}

{Variabel}

- dataItemInventory: list

- dataMonsterInventory: list

- dataMonster: list

- userId: integer

- username: string

- coin: integer

- allPotionAgent: list

- allMonsterAgent: list

- monsterAgent: list

- nameMonsterAgent: string.

- atkPowerMonsterAgent: integer

- defPowerMonsterAgent: integer
- hpMonsterAgent: integer
- levelMonsterAgent: integer
- currentHpMonsterAgent: integer
- levelEnemy: integer
- reward: list
- stage: integer
- sumCoin: integer
- enemy: list
- nameEnemy: string
- atkPowerEnemy: integer
- defPowerEnemy: integer
- hpEnemy: integer
- currentHpEnemy: integer
- turn: integer
- countStrength: integer
- countResilience: integer
- countHealing: integer
- isWin: bool
- actionTurn: string
- choicePotion: string
- rewardCoin: integer
- choiceMonster: string

### ALGORITMA

function arena(username: string, dataMonster: list, allPotionAgent: list, coin: int) -> tuple:

  Output ( 'Selamat Datang di Arena!!')

  displayMonsterAgent(allMonsterAgent)

  iterate

    if (not isNum(choiceMonster)) then

      output("Pilihan tidak valid!")

  else

    stop (1 <= choiceMonster <= length(allMonsterAgent)) then

      output("Pilihan nomor tidak tersedia!")

  monsterAgent ← allMonsterAgent[choiceMonster - 1]

  nameMonsterAgent ← monsterAgent[0]

  atkPowerMonsterAgent ← monsterAgent[1]

  defPowerMonsterAgent ← monsterAgent[2]

  hpMonsterAgent ← monsterAgent[3]

  levelMonsterAgent ← monsterAgent[4]

  (atkPowerMonsterAgent, defPowerMonsterAgent, hpMonsterAgent) ←

  attributeByLevel(levelMonsterAgent, atkPowerMonsterAgent, defPowerMonsterAgent, hpMonsterAgent)

  currentHpMonsterAgent ← hpMonsterAgent

  output("Gambar Monster")



```

output("RAWRRR, Agent " + username + " mengeluarkan Monster " + nameMonsterAgent + "
!!!")
statusMonster(nameMonsterAgent, atkPowerMonsterAgent, defPowerMonsterAgent,
currentHpMonsterAgent, levelMonsterAgent)

levelEnemy ← 0
stage ← 1
sumCoin ← 0

iterate
  enemy ← dataMonster[getRandomNumber([1, length(dataMonster)])]
  nameEnemy ← enemy[1]
  atkPowerEnemy ← enemy[2]
  defPowerEnemy ← enemy[3]
  hpEnemy ← enemy[4]
  levelEnemy ← levelEnemy + 1

  (atkPowerEnemy, defPowerEnemy, hpEnemy) ←
  attributeByLevel(levelEnemy, atkPowerEnemy, defPowerEnemy, hpEnemy)
  currentHpEnemy ← hpEnemy

output("===== STAGE " + stage + " =====")
output("RAWRRR, Monster " + nameEnemy + " telah muncul !!!")
statusMonster(nameEnemy, atkPowerEnemy, defPowerEnemy, currentHpEnemy, levelEnemy)

turn ← 1
countStrength ← 0
countResilience ← 0
countHealing ← 0
isWin ← true

while isWin or stage <= 5 do
  output("===== TURN " + turn + " (" + nameMonsterAgent + ") =====")
  output("1. Attack")
  output("2. Use Potion")
  output("3. Quit")

  iterate
    input(actionTurn)
    if (not isNum(actionTurn)) then
      output("Perintah tidak valid!")
    else
      stop (actionTurn = 1 or actionTurn = 2 or actionTurn = 3)
      output("Perintah tidak valid!")

  depends on (actionTurn)
  actionTurn = 1
  currentHpEnemy ← attack(nameMonsterAgent, atkPowerMonsterAgent, nameEnemy,
defPowerEnemy, atkPowerEnemy, currentHpEnemy, levelEnemy)
  if (currentHpEnemy = 0) then

```

```

        output("Selamat, Anda berhasil mengalahkan monster " + nameEnemy + " !!!")
rewardCoin ← reward[stage - 1]
sumCoin ← sumCoin + rewardCoin
output("Total OC yang diperoleh: " + rewardCoin)
output("HP Anda saat ini: " + currentHpMonsterAgent)
isWin ← true
stop

else
    output("===== TURN " + turn + " (" + nameEnemy + ") =====")
    currentHpMonsterAgent ← attack(nameEnemy, atkPowerEnemy, nameMonsterAgent,
defPowerMonsterAgent, atkPowerMonsterAgent, currentHpMonsterAgent, levelMonsterAgent)
    if (currentHpMonsterAgent = 0) then
        output("Yahhh, Anda dikalahkan monster " + nameEnemy + ". Jangan menyerah, coba
lagi !!!")
        output("GAME OVER! Sesi latihan berakhir pada stage " + stage + "!!")
        isWin ← false
        stop

    turn ← turn + 1

actionTurn = 2
displayPotionAgent(allPotionAgent)

Iterate
input(choicePotion)

if (not isNum(choicePotion)) then
    output("Perintah tidak valid!")
else
    depends on (int(choicePotion, len(allPotionAgent))
1<=int(choicePotion)<=len(allPotionAgent):
        if allPotionAgent[int(choicePotion)-1][1] = 0 then
            output("Maaf {allPotionAgent[int(choicePotion)-1][0].capitalize()} Potion kamu
telah habis, silahkan beli lagi di SHOP!")
            stop
        depends on (allPotionAgent[int(choicePotion)-1][0])
        allPotionAgent[int(choicePotion)-1][0] == 'strength':
            if (countStrength = 1) then
                output("Kamu mencoba memberikan ramuan ini kepada " +
nameMonsterAgent + ", namun dia menolaknya.")
            else
                output("Setelah meminum ramuan ini, aura kekuatan terlihat mengelilingi "
+ nameMonsterAgent + ".")
                allPotionAgent[choicePotion - 1][1] ← allPotionAgent[choicePotion -
1][1] - 1
                countStrength ← countStrength + 1
                atkPowerMonsterAgent ← atkPowerMonsterAgent + 0.05 *
atkPowerMonsterAgent

```

```

        allPotionAgent[int(choicePotion)-1][0] == 'resilience':
            if (countResilience = 1) then
                output("Kamu mencoba memberikan ramuan ini kepada " +
nameMonsterAgent + ", namun dia menolaknya.")
            else
                output("Setelah meminum ramuan ini, energi pelindung muncul di sekitar "
+ nameMonsterAgent + ".")
                allPotionAgent[choicePotion - 1][1] ← allPotionAgent[choicePotion -
1][1] - 1
                countResilience ← countResilience + 1
                defPowerMonsterAgent ← defPowerMonsterAgent + 0.05 *
defPowerMonsterAgent

            else
                if (countHealing = 1) then
                    output("Kamu mencoba memberikan ramuan ini kepada " +
nameMonsterAgent + ", namun dia menolaknya.")
                else
                    output("Setelah meminum ramuan ini, luka-luka pada " +
nameMonsterAgent + " sembuh dengan cepat.")
                    allPotionAgent[choicePotion - 1][1] ← allPotionAgent[choicePotion -
1][1] - 1
                    countHealing ← countHealing + 1
                    hpMonsterAgent ← hpMonsterAgent + 0.25 * hpMonsterAgent
                    if (hpMonsterAgent > monsterAgent[3]) then
                        hpMonsterAgent ← monsterAgent[3]
                    stop

                int(choicePotion) == len(allPotionAgent)+1:
                    stop
                else
                    output("Perintah tidak valid!")

            else
                output("GAME OVER! Anda mengakhiri sesi latihan!!")
                isWin ← false
                stop

        if (isWin = false) then
            output("===== STATS =====")
            output("Total hadiah: " + sumCoin)
            output("Jumlah stage: " + stage)
            output("Damage diberikan: " + (hpEnemy - currentHpEnemy))
            output("Damage diterima: " + (hpMonsterAgent - currentHpMonsterAgent))
            coin ← coin + sumCoin
            stop
        else
            damageDealt ← damageDealt + hpEnemy-currentHpEnemy
            damageReceived ← damageDealt + hpMonsterAgent-currentHpMonsterAgent
            Output ("\n\nMemulai stage berikutnya...")

```



{Mengelola pembelian monster dan potion yang dilakukan agent}

## KAMUS

{Spesifikasi fungsi dan prosedur}

**function** isMonsterExistInInventory(userId, idMonster, dataMonsterInventory)

{Memeriksa apakah sebuah monster terdapat pada inventory agent}

**function** isPotionExistInInventory(userId, namePotion, dataItemInventory)

{Memeriksa apakah sebuah monster terdapat pada inventory agent}

**procedure** shop (dataMonster, dataMonsterShop, dataItemShop, dataMonsterInventory, dataItemInventory, coin, userId)

{Menampilkan UI Shop}

{Deklarasi tipe dan variabel}

-dataMonster : list of lists

-dataItemInventory : list of lists,

-dataMonsterInventory : list of lists

-userId : string

-coin : integer

-monster : list

-item : list

-idMonster : string

-id : string

-namePotion : string

-dataMonsterShop : list of lists

-dataItemShop: list of lists

## ALGORITMA

**procedure** shop (dataMonster, dataMonsterShop, dataItemShop, dataMonsterInventory, dataItemInventory, coin, userId)

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

output("SHOP")

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

iterate

    action ← toUpperCase(input('>>> Pilih aksi (LIHAT/BELI/KELUAR)')

    if action = 'LIHAT' then

        iterate

            option ← toUpperCase(input('>>> Mau lihat apa ? (MONSTER/POTION)')

            if option = 'MONSTER' then

                displayMonsterShop(dataMonsterShop, dataMonster)

                stop

            elif option = 'POTION' then

                displayPotionShop(dataItemShop)

```

        stop
    else
        output("Pilihan tidak valid!")

    elif action = 'BELI' then
        output("Jumlah O.W.C.A. Coin-mu sekarang {coin}.")
        iterate
            option ← toUpperCase(input('>>> Mau beli apa ? (MONSTER/POTION)'))
            if option = 'MONSTER' then
                displayMonsterShop(dataMonsterShop, dataMonster)

                if len(dataMonsterShop) - 1 > 0 then
                    iterate
                        idMonster ← input('>>> Masukkan id monster: ')
                        isValid = False
                        for monster in dataMonsterShop do
                            if monster[0] = idMonster then
                                isValid = true
                                stop
                            if isValid = True then
                                stop
                            else
                                output('Id Monster tidak valid!')

                                nameMonster = getDataById(idMonster, dataMonster)[1]
                                price = int(getDataById(idMonster, dataMonsterShop)[2])
                                stock = int(getDataById(idMonster, dataMonsterShop)[1])

                                if isMonsterExistInInventory(userId, idMonster, dataMonsterInventory) = True
then
                                    output('Monster {nameMonster} sudah ada dalam inventory-mu! Pembelian
dibatalkan.')
                                else
                                    if coin < price then
                                        output('O.W.C.A Coin Anda tidak cukup untuk membeli
{nameMonster}.')
                                    else
                                        if stock > 0 then
                                            output('Berhasil membeli item: {nameMonster}. Item sudah masuk ke
inventory-mu!')
                                            coin ← coin - price
                                            stock ← stock - 1
                                            for monster in dataMonsterShop do
                                                if monster[0] = idMonster then
                                                    monster[1] ← stock
                                                    append(dataMonsterInventory, [str(userId), idMonster, '1'])
                                                else
                                                    output("Yahh stok {nameMonster} pada shop sudah habis..")
                                            else
                                                output("Tidak ada monster yang dapat dibeli...")

```

```

stop

elif option = 'POTION' then
    displayPotionShop(dataItemShop)

    if len(dataItemShop) - 1 > 0 then
        iterate
        idPotion ← input('>>> Masukkan id potion: ')
        if isNum(idPotion) = False then
            output("Id Potion tidak valid!")
        else
            if 1 ≤ int(idPotion) ≤ len(dataItemShop) - 1:
                stop
            else
                print("Id Potion tidak valid!")

        namePotion = dataItemShop[int(idPotion)][0]
        stock = int(dataItemShop[int(idPotion)][1])
        price = int(dataItemShop[int(idPotion)][2])

        iterate
        quantity ← input('>>> Masukkan jumlah: ')
        if isNum(quantity) = False then
            output("Jumlah tidak valid!")
        else
            quantity ← int(quantity)
            if 1 ≤ quantity ≤ stock then
                stop
            elif quantity > stock then
                output("Stok {namePotion} Potion pada shop tidak mencukupi")
            else
                output("Jumlah tidak valid!")

        if coin < quantity*price then
            output("O.W.C.A Coin Anda tidak cukup untuk membeli {quantity}
{namePotion} Potion.")
        else
            if stock > 0 then
                output("Berhasil membeli item: {quantity} {namePotion} Potion. Item
sudah masuk ke inventory-mu!")
                coin ← coin - quantity*price
                stock ← stock - quantity

                for potion in dataItemShop do
                    if potion[0] = namePotion then
                        potion[1] = stock
                    if isPotionExistInInventory(userId, namePotion, dataItemInventory) =
True then
                        for potionInvent in dataItemInventory do

```





```

{Menampilkan Monster pada Inventory}
output("\n===== MONSTER LIST =====")
allMonsterAgent <- getMonsterAgent(dataMonsterInventory,dataMonster,userId)
x <- 1
monster traversal [allMonsterAgent]:
    output(f'{x}. {monster[0]} (Level: {monster[4]})')
    x <- x + 1

{Validasi Pilihan Monster}
repeat
    input(pilihmonster)
    if isNum(pilihMonster) == False then
        output('Pilihan tidak valid!')
    Else:
        if 1<=int(pilihMonster)<=len(allMonsterAgent):
            Break
        Else:
            output('Pilihan nomor tidak tersedia!')

monster <- allMonsterAgent[int(pilihMonster)-1]
nameMonster <- monster[0]
level <- int(monster[4])
if level == 5 then
    output("\nMaaf, monster yang Anda pilih sudah memiliki level maksimum')
else:
    output(f'\n{nameMonster} akan di-upgrade ke level {level+1}.'.)

{Menampilkan Harga Upgrade Berdasarkan Level}
depend on (level, price)
    level == 1 : price == 200
    level == 2 : price == 400
    level == 3 : price == 600
    level == 4 : price == 800

    output(f'Harga untuk melakukan upgrade {nameMonster} adalah {price} OC.')

{Konfirmasi Upgrade}
repeat
    confirm <- input("\nLanjutkan upgrade (Y/N): ').upper()
    if confirm == 'Y' then
        if coin<price then
            output(f'\nMaaf OC Anda tidak cukup untuk melakukan upgrade
{nameMonster}.'.)
        else:
            coin <- coin - price
            level <- level +1
            allMonsterAgent[int(pilihMonster)-1][4] <- level
            output(f'\nSelamat, {nameMonster} berhasil di-upgrade ke level
{level} !')

```

```

        break
    elif confirm == 'N':
        output(f'\nYah sayang sekali Anda tidak jadi melakukan upgrade
        terhadap {nameMonster}:(')
        break
    Else:
        output('Konfirmasi tidak valid !')

{Merubah level monster pada data monster inventory}
row traversal [dataMonsterInventory]:
    if row[0] == str(userId) then
        nameMonster <- getDataById(row[1],dataMonster)[1]
        monster traversal [allMonsterAgent]:
            if monster[0] == nameMonster then
                row[2] <- monster[4]

return coin

```

## 11. F12 - Shop Management

**Program** shopManagement  
 {Program untuk mengatur isi shop}


**Kamus**  
 {Spesifikasi fungsi dan prosedur}

**procedure** shopManagement (dataItemShop : list, dataMonsterShop : list, dataMonster : list)  
 {I.S kondisi True  
 F.S menampilkan UI untuk mengatur shop}

### ALGORITMA PROGRAM UTAMA

```

output("\n=====
=====')

output('

')

output("\n=====
=====')

output(f'\nSelamat Datang Kembali, {username} !')

repeat
    input('\n>>> Pilih aksi (LIHAT/TAMBAH/UBAH/HAPUS/KELUAR): ').upper()
    if action == 'LIHAT' then

```

```

Repeat
    input('\n>>> Mau lihat apa? (MONSTER/POTION): ').upper()
    Depend on (option):
    option == 'Monster' :
        displayMonsterShop(dataMonsterShop,dataMonster, Break
    option == 'potion' : displayPotionShop(dataItemShop)
        break
    else:
        output('Input tidak valid!')
elif action == 'TAMBAH':
    repeat
        input('\n>>> Mau nambahin apa? (MONSTER/POTION): ').upper()

{ Menambah monster yang akan di jual}
if option == 'MONSTER' then
    { Menampilkan seluruh monster yang ada di database tetapi belum ada pada shop
    monsterNotInShop:list = getMonsterNotInShop(dataMonsterShop,dataMonster)
    if len(monsterNotInShop)-1>=0:
        output(f'\n{'ID':<3} | {'Type':<15} | {'ATK Power':<9} | {'DEF Power':<9} |
{'HP':<5}")
        traversal i in range(len(monsterNotInShop)):
            output(f'{'monsterNotInShop[i][0]:<3} | {'monsterNotInShop[i][1]:<15} |
{'monsterNotInShop[i][2]:<9} | {'monsterNotInShop[i][3]:<9} | {'monsterNotInShop[i][4]:<5}")
            { Validasi input id monster
            repeat
                newIdMonster:str = input('\n>>> Masukkan id monster: ')
                isValid:bool = False
                traversal monster in monsterNotInShop:
                    if monster[0] == newIdMonster:
                        isValid = True
                        break
                if isValid == True:
                    break
                else:
                    output('Id Monster tidak valid!')
            { Validasi stok awal}
            repeat
                stock:str = input('\n>>> Masukkan stok awal: ')
                if isNum(stock) == True:
                    break
                else:
                    output('Stok tidak valid')
            { Validasi harga

```

```

repeat
    price:str = input("\n>>> Masukkan harga: ")
    if isNum(price) == True:
        break
    else:
        output('Harga tidak valid')
    { Tambah data baru ke dataMonsterShop
    newMonsterShop:list = [newIdMonster,stock,price]
    dataMonsterShop.append(newMonsterShop)
    output(f'\n {dataMonster[int(newIdMonster)][1]} telah berhasil ditambahkan ke
dalam shop!')
    else:
        output('\nSeluruh monster pada database telah ditambahkan pada shop...')
    break

{ Menambah potion yang akan di jual
elif option == 'POTION':
    { Menampilkan seluruh potion yang ada di database tetapi belum ada pada shop
    potionNotInShop:list = getPotionNotInShop(dataItemShop,dataPotion)
    if len(potionNotInShop)-1>=0:
        output(f'\n {'ID':<3} | {'Type':<20}')
        traversal i in range(len(potionNotInShop)):
            output(f' {len(dataItemShop)+i:<3} | {' {potionNotInShop[i].capitalize()}
Potion':<20}')
        { Validasi input id potion
        repeat
            newIdPotion:str = input("\n>>> Masukkan id potion: ")
            if isNum(newIdPotion)==False:
                output('Id Potion tidak valid!')
            else:
                if
len(dataItemShop)<=int(newIdPotion)<=len(dataItemShop)+len(potionNotInShop)-1:
                    break
                else:
                    output('Id Potion tidak valid!')
        { Validasi stok awal
        repeat
            stock:str = input("\n>>> Masukkan stok awal: ")
            if isNum(stock) == True:
                break
            else:
                output('Stok tidak valid')
        { Validasi harga

```

```

repeat
    price:str = input("\n>>> Masukkan harga: ")
    if isNum(price) == True:
        break
    else:
        output('Harga tidak valid')
    { Tambah data baru ke dataItemShop
    newItemShop:list =
[potionNotInShop[int(newIdPotion)-len(dataItemShop)],stock,price]
    output(f' {potionNotInShop[int(newIdPotion)-len(dataItemShop)].capitalize()}
Potion telah berhasil ditambahkan ke dalam shop!')
    dataItemShop.append(newItemShop)
else:
    output("\nSeluruh potion pada database telah ditambahkan pada shop...")
break
else:
    output('Input tidak valid!')

elif action == 'UBAH':
    repeat
        option:str = input("\n>>> Mau ubah apa? (MONSTER/POTION): ").upper()
        if option == 'MONSTER':
            { Menampilkan seluruh monster yang ada di shop
            displayMonsterShop(dataMonsterShop,dataMonster)

            if len(dataMonsterShop)-1>0:
                { Validasi input id monster
                repeat
                    updateIdMonster = input("\n>>> Masukkan id monster: ")
                    isValid:bool = False
                    traversal monster in dataMonsterShop:
                        if monster[0] == updateIdMonster:
                            isValid = True
                            break
                        if isValid == True:
                            break
                    else:
                        output('Id Monster tidak valid!')
                { Validasi stok baru
                repeat
                    stock:str = input("\n>>> Masukkan stok baru: ")
                    if isNum(stock) == True or not stock:
                        break

```

```

        else:
            output('Stok tidak valid')
        { Validasi harga baru
repeat
    price:str = input("\n>>> Masukkan harga baru: ")
    if isNum(price) == True or not price:
        break
    else:
        output('Harga tidak valid')

    { Update dataMonsterShop dengan data yang baru
    dataMonsterShop:list =
updateData(dataMonsterShop,updateIdMonster,[updateIdMonster,stock,price])

    nameMonster:str = getDataById(updateIdMonster,dataMonster)[1]
    if not stock and not price:
        output(f'\nTidak ada data {nameMonster} yang diubah.')
    elif not stock:
        output(f'\n{nameMonster} telah berhasil diubah dengan harga baru {price}')
    elif not price:
        output(f'\n{nameMonster} telah berhasil diubah dengan stok baru sejumlah
{stock}')
    else:
        output(f'\n{nameMonster} telah berhasil diubah dengan stok baru sejumlah
{stock} dan dengan harga baru {price}')
    else:
        output('Tidak ada data monster yang dapat diubah...')
        break
elif option == 'POTION':
    { Menampilkan seluruh potion yang ada di shop
displayPotionShop(dataItemShop)

if len(dataItemShop)-1>0:
    { Validasi input id potion
repeat
    updateIdPotion:str = input("\n>>> Masukkan id potion: ")
    if isNum(updateIdPotion)==False:
        output('Id Potion tidak valid!')
    else:
        if 1<=int(updateIdPotion)<=len(dataItemShop)-1:
            break
        else:
            output('Id Potion tidak valid!')

```

```

        { Validasi stok baru
repeat
    stock:str = input('\n>>> Masukkan stok baru: ')
    if isNum(stock) == True or not stock:
        break
    else:
        output('Stok tidak valid')
    { Validasi harga baru
repeat
    price:str = input('\n>>> Masukkan harga baru: ')
    if isNum(price) == True or not price:
        break
    else:
        output('Harga tidak valid')

    { Update dataItemShop dengan data yang baru
    namePotion:str = dataItemShop[int(updateIdPotion)][0]
    dataItemShop:list =
updateData(dataItemShop,namePotion,[namePotion,stock,price])
    if not stock and not price:
        output(f'\nTidak ada data {namePotion.capitalize()} Potion yang diubah.')
    elif not stock:
        output(f'\n{namePotion.capitalize()} Potion telah berhasil diubah dengan
harga baru {price}')
    elif not price:
        output(f'\n{namePotion.capitalize()} Potion telah berhasil diubah dengan stok
baru sejumlah {stock}')
    else:
        output(f'\n{namePotion.capitalize()} Potion telah berhasil diubah dengan stok
baru sejumlah {stock} dan dengan harga baru {price}')
    else:
        output('\nTidak ada data potion yang dapat diubah...')
        break
    else:
        output('Input tidak valid!')

elif action == 'HAPUS':
    repeat
        option:str = input('\n>>> Mau hapus apa? (MONSTER/POTION): ').upper()
        if option == 'MONSTER' then
            { Menampilkan seluruh monster yang ada di shop
            displayMonsterShop(dataMonsterShop,dataMonster)

```

```

if len(dataMonsterShop)-1>0:
    { Validasi input id monster
    repeat
        delIdMonster:str = input("\n>>> Masukkan id monster: ")
        isValid:bool = False
        Monster traversal dataMonsterShop:
            if monster[0] == delIdMonster then
                isValid <- True
                break
            if isValid == True then
                break
            else:
                output('Id Monster tidak valid!')
        nameMonster:str = getDataById(delIdMonster,dataMonster)[1]

    { Validasi y/n delete
    repeat
        delete:str = input(f'\n>>> Apakah anda yakin ingin menghapus
{nameMonster} dari shop (Y/N)? ').upper()
        if delete == 'Y':
            { Menghapus data monster pada shop yang memiliki delIdMonster
            dataMonsterShop = deleteData(dataMonsterShop,delIdMonster)
            output(f'\n {nameMonster} telah berhasil dihapus dari shop!')
            break
        elif delete == 'N':
            output(f'\n {nameMonster} tidak jadi Anda hapus.')
            break
        else:
            output('Input tidak valid!')
    else:
        output('\nTidak ada data monster yang dapat dihapus...')
    break
elif option == 'POTION':
    { Menampilkan seluruh potion yang ada di shop
    displayPotionShop(dataItemShop)

if len(dataItemShop)-1>0:
    { Validasi input id potion
    repeat
        delIdPotion:str = input("\n>>> Masukkan id potion: ")
        if isNum(delIdPotion)==False:
            output('Id Potion tidak valid!')
        else:

```



```

        if 1<=int(delIdPotion)<=len(dataItemShop)-1:
            break
        else:
            output('Id Potion tidak valid!')
    { Validasi y/n delete
delNamePotion:str = dataItemShop[int(delIdPotion)][0]
repeat
    delete:str = input(f'\n>>> Apakah anda yakin ingin menghapus
{delNamePotion.capitalize()} Potion dari shop (y/n)? ').lower()
    if delete == 'y':
        output(f'\n{delNamePotion.capitalize()} Potion telah berhasil dihapus dari
shop!')

        { Menghapus data potion pada shop yang memiliki delIdMonster
dataItemShop:list = deleteData(dataItemShop,delNamePotion)
        break
    elif delete == 'n':
        output(f'\n{delNamePotion.capitalize()} Potion tidak jadi Anda hapus.')
        break
    else:
        output('Input tidak valid!')
else:
    output('\nTidak ada data potion yang dapat dihapus...')
    break
else:
    output('Input tidak valid!')
elif action == 'KELUAR':
    output(f'\nSampai Jumpa Lagi {username} !!!')
    break
else:
    output('Aksi tidak valid!')

```

## 12. F13 - Monster Management

### **Program** MonsterManagement

{Spesifikasi: program untuk mengatur monster di dalam “owca-dex” oleh admin}

### **KAMUS**

{Spesifikasi fungsi dan prosedur}

**function** foundMonster(dataMonster:list, newMonster:str) -> bool

{I.S. dataMonster dan newMonster terdefinisi

F.S. Boolean ditemukan monster}

**procedure** displayMonster(dataMonster:list)

{I.S. dataMonster terdefinisi  
F.S. menampilkan UI monster pada shop}

**procedure** addMonster(dataMonster:list)

{I.S. dataMonster terdefinisi  
F.S. monster ditambahkan ke monster shop}

**procedure** monsterManagement(dataMonster:list, username:str)

{I.S. dataMonster dan username terdefinisi  
F.S. menampilkan menu monster management}

### ALGORITMA

**function** foundMonster(dataMonster:list, newMonster:str) -> bool

repeat len(dataMonster) times

if monster[1] == newMonster then  
-> True

-> False

**procedure** displayMonster(dataMonster:list)

if len(dataMonster)-1>0 then

output('ID | Type | ATK Power | DEF Power | HP |')

repeat len(dataMonster) times

output( {dataMonster[i][0]} | {dataMonster[i][1]} | {dataMonster[i][2]} |  
{dataMonster[i][3]} | {dataMonster[i][4]} |")

else

output('Tidak ada data yang dapat ditampilkan...')

**procedure** addMonster(dataMonster:list)

output('Memulai pembuatan monster baru')

repeat

input(newType)

if isNum(newType) == True then

output('Nama harus string, coba lagi!')

else

output('Nama sudah terdaftar, coba lagi!')

until foundMonster(dataMonster, newType) == False

repeat

input(newATKPower)

output('Masukkan input bertipe Integer, coba lagi!')

until isNum(newATKPower) == True

repeat

input('newDEFPower)

if isNum(newDEFPower) == True then

print('DEF Power harus bernilai 0-50, coba lagi!')

else

print('Masukkan input bertipe Integer, coba lagi!')

until 0<=newDEFPower<=50

repeat

input(newHP)

output('Masukkan input bertipe Integer, coba lagi!')

until isNum(newHP) == True

```

    output(newType)
    output(newATKPower)
    output(newDEFPower)
    output(newHP)
    repeat
        input(confirm)
        depend on (confirm)
        confirm == 'Y' :
            newMonster ->
[str(newId(dataMonster)),newType,newATKPower,newDEFPower,newHP]
            output('Monster baru telah ditambahkan!')
        confirm == 'N'
            output('Monster gagal ditambahkan!')
    else
        output('Konfirmasi tidak valid!')
    until confirm == 'N' or confirm == 'Y'

procedure monsterManagement(dataMonster:list,username:str)
    output('=====')
    output('SELAMAT DATANG DI DATABASE PARA MONSTER {username} !!!')
    output('1. Tampilkan Semua Monster')
    output('2. Tambah Monster baru')
    output('3. Keluar')
    repeat
        input(action)
        depend on (action)
        action == '1'
            displayMonster(dataMonster)
        action == '2':
            addMonster(dataMonster)
    else:
        output('Aksi tidak valid!')
    until action == '3'

```

### 13. F14 - Load

```

{ Spesifikasi fungsi read_csv }
{ I.S.: Nama file CSV (filename) dan path folder (folder_path) terdefinisi }
{ F.S.: Data dari file CSV dibaca dan disimpan dalam bentuk matriks }

```

#### KAMUS

```

{ Deklarasi variabel }
filename, folder_path : string
data : array of array of string

```

file\_path : string

### ALGORITMA

```
IF Karakter terakhir dari folder_path ≠ '/' THEN
    folder_path ← folder_path + '/' { Menambahkan '/' di akhir folder_path }
    file_path ← folder_path + filename { Menggabungkan folder_path dan filename untuk
membentuk path lengkap file }
```

```
data ← array kosong
```

```
{ Cek keberadaan file }
```

```
IF Tidak ada file pada file_path THEN
```

```
    Output "Error: The file filename was not found in the folder folder_path."
```

```
    RETURN data
```

```
{ Baca file CSV sebagai matriks }
```

```
OPEN file di file_path UNTUK DIBACA sebagai file
```

```
UNTUK SETIAP baris DALAM file LAKUKAN
```

```
    row ← array kosong
```

```
    elmt ← "
```

```
    UNTUK SETIAP karakter DALAM baris LAKUKAN
```

```
        IF karakter ≠ '\n' DAN karakter ≠ '\r' THEN
```

```
            IF karakter = ';' THEN
```

```
                row ← row + [elmt]
```

```
                elmt ← "
```

```
            ELSE
```

```
                elmt ← elmt + karakter
```

```
        IF elmt ≠ " THEN
```

```
            row ← row + [elmt]
```

```
        data ← data + [row]
```

```
RETURN data
```

```
{ Spesifikasi fungsi load_data }  
{ I.S.: Nama folder (folder) terdefinisi }  
{ F.S.: Data dari file-file CSV dalam folder tersebut dimuat }
```

### **KAMUS**

```
{ Deklarasi variabel }  
data_directory : string  
folder_path : string  
file_csv : array of string  
missing_file : boolean  
dataUser, dataMonster, dataMonsterInventory, dataItemInventory, dataItemShop,  
dataMonsterShop : array of array of string
```

### **ALGORITMA**

```
data_directory ← direktori tempat data berada (../data/)  
folder_path ← gabungkan data_directory dengan folder { membuat path lengkap folder }  
  
file_csv ← ['user.csv', 'monster.csv', 'item_inventory.csv', 'monster_inventory.csv',  
'item_shop.csv', 'monster_shop.csv']  
  
{ Jika terdapat folder dengan nama yang diberikan }  
IF Terdapat folder pada folder_path THEN  
    missing_file ← false  
  
    { Cek keberadaan file csv }  
    UNTUK i <- 0 HINGGA PANJANG file_csv LAKUKAN  
        IF Tidak ada file pada folder_path dan file_csv[i] THEN  
            missing_file ← true  
  
    { Jika ada file csv yang hilang }  
    IF missing_file THEN
```

```
        Output "Terdapat file csv yang hilang. Harap periksa kembali file csv pada folder
Anda!"
```

```
    ELSE
```

```
        Output "Loading..."
```

```
        Tunda proses selama 1 detik
```

```
        { Memuat data CSV }
```

```
        dataUser ← read_csv('user.csv', folder_path)
```

```
        dataMonster ← read_csv('monster.csv', folder_path)
```

```
        dataMonsterInventory ← read_csv('monster_inventory.csv', folder_path)
```

```
        dataItemInventory ← read_csv('item_inventory.csv', folder_path)
```

```
        dataItemShop ← read_csv('item_shop.csv', folder_path)
```

```
        dataMonsterShop ← read_csv('monster_shop.csv', folder_path)
```

```
        Output "Selamat datang di program OWCA!"
```

```
        RETURN dataUser, dataMonster, dataMonsterInventory, dataItemInventory,
dataMonsterShop, dataItemShop
```

```
    ELSE
```

```
        Output "Folder folder tidak ditemukan."
```

```
        Hentikan eksekusi program
```

#### 14. F15 - Save

```
{ Spesifikasi fungsi writeCSV }
```

```
{ I.S.: Path folder (folder_path), nama file CSV (csv_file), dan data dalam bentuk array 2D
(array_data) terdefinisi }
```

```
{ F.S.: Data disimpan dalam file CSV di path folder dengan nama yang diberikan }
```

##### **KAMUS**

```
{ Deklarasi variabel }
```

```
folder_path, csv_file : string
```

```
array_data : array of array of string
```

file : file

i, j : integer

line : string

### ALGORITMA

OPEN file di folder\_path dengan nama csv\_file UNTUK ditulis sebagai file

UNTUK i <- 0 SAMPAI PANJANG array\_data LAKUKAN

    line ← ""

    UNTUK j <- 0 SAMPAI PANJANG array\_data[i] LAKUKAN

        line ← line + KonversiKeString(array\_data[i][j])

    IF j < PANJANG array\_data[i] - 1 THEN

        line ← line + ";" { Tambahkan ';' jika bukan elemen terakhir dalam baris }

    ELSE

        line ← line + "\n" { Tambahkan '\n' untuk pindah baris setelah baris terakhir }

    TULIS line ke dalam file

TUTUP file

{ Spesifikasi fungsi save }

{ I.S.: Data pengguna (dataUser), data monster (dataMonster), inventori monster (dataMonsterInventory), inventori item (dataItemInventory), toko monster (dataMonsterShop), dan toko item (dataItemShop) terdefinisi }

{ F.S.: Data disimpan dalam file CSV di folder data yang ditentukan }

### KAMUS

{ Deklarasi variabel }

dataUser, dataMonster, dataMonsterInventory, dataItemInventory, dataMonsterShop, dataItemShop : array of array of string

folder\_name, data\_directory, folder\_path : string

### ALGORITMA

folder\_name ← Input("Masukkan nama folder : ") { Meminta nama folder }

data\_directory ← direktori tempat data berada (../data/)

```
folder_path ← gabungan data_directory dengan folder_name { membuat path lengkap  
folder }
```

```
{ Jika folder data_directory ada }
```

```
IF Terdapat folder pada data_directory THEN
```

```
{ Jika folder folder_path sudah ada }
```

```
IF Terdapat folder pada folder_path THEN
```

```
Output "Folder data/folder_name sudah ada."
```

```
Tunda proses selama 1 detik
```

```
Output "Saving..."
```

```
Tunda proses selama 1 detik
```

```
{ Konversi data ke file CSV }
```

```
writeCSV(folder_path, "user.csv", dataUser)
```

```
writeCSV(folder_path, "monster.csv", dataMonster)
```

```
writeCSV(folder_path, "monster_inventory.csv", dataMonsterInventory)
```

```
writeCSV(folder_path, "item_inventory.csv", dataItemInventory)
```

```
writeCSV(folder_path, "monster_shop.csv", dataMonsterShop)
```

```
writeCSV(folder_path, "item_shop.csv", dataItemShop)
```

```
Output "Berhasil menyimpan data di folder data/folder_name!"
```

```
{ Jika folder folder_path belum ada }
```

```
Else:
```

```
Buat folder folder_path
```

```
Output "Saving..."
```

```
Tunda proses selama 1 detik
```

```
Output f"Membuat folder data/{folder_name}..."
```

```
{ Konversi data ke file CSV }
```

```
writeCSV(folder_path, "user.csv", dataUser)
```

```
writeCSV(folder_path, "monster.csv", dataMonster)
```

```
writeCSV(folder_path, "monster_inventory.csv", dataMonsterInventory)
```



```
writeCSV(folder_path, "item_inventory.csv", dataItemInventory)
writeCSV(folder_path, "monster_shop.csv", dataMonsterShop)
writeCSV(folder_path, "item_shop.csv", dataItemShop)
Output "Berhasil menyimpan data di folder data/folder_name!"
```

```
{ Jika folder data_directory tidak ada }
```

```
Else:
```

```
Output "Folder data tidak ditemukan."
```

```
Output "Saving..."
```

```
Tunda proses selama 1 detik
```

```
Output "Membuat folder data/..."
```

```
Tunda proses selama 1 detik
```

```
Buat folder data_directory
```

```
Output f"Membuat folder data/{folder_name}..."
```

```
Tunda proses selama 1 detik
```

```
Buat folder folder_path
```

```
{ Konversi data ke file CSV }
```

```
writeCSV(folder_path, "user.csv", dataUser)
```

```
writeCSV(folder_path, "monster.csv", dataMonster)
```

```
writeCSV(folder_path, "monster_inventory.csv", dataMonsterInventory)
```

```
writeCSV(folder_path, "item_inventory.csv", dataItemInventory)
```

```
writeCSV(folder_path, "monster_shop.csv", dataMonsterShop)
```

```
writeCSV(folder_path, "item_shop.csv", dataItemShop)
```

```
Output "Berhasil menyimpan data di folder data/folder_name!"
```

## 15. F16 - Exit

```
{ Spesifikasi fungsi exit }
```

```
{ I.S.: Data pengguna (dataUser), data monster (dataMonster), inventori monster (dataMonsterInventory), inventori item (dataItemInventory), toko monster (dataMonsterShop), dan toko item (dataItemShop) terdefinisi }
```

{ F.S.: Pengguna diminta untuk menyimpan data yang telah diubah. Jika pengguna memilih 'Y', data disimpan; jika 'N', program berakhir }

## **KAMUS**

{ Deklarasi variabel }

dataUser, dataMonster, dataMonsterInventory, dataItemInventory, dataMonsterShop,  
dataItemShop : array of array of string

response : string

## **ALGORITMA**

While True Do

    response ← Input("\nApakah Anda mau melakukan penyimpanan file yang sudah diubah?

(Y/N): ").dikonversiMenjadiHurufBesar()

    IF response = "Y" THEN

        Output "\nSaving...\n"

        Tunda proses selama 1 detik

        save(dataUser, dataMonster, dataMonsterInventory, dataItemInventory,  
dataMonsterShop, dataItemShop)

        KELUAR\_DARI\_PERULANGAN

    ELSE IF response = "N" THEN

        Output "\nYahh progress kamu hilang, selamat tinggal :(\n"

        KELUAR\_DARI\_PERULANGAN

    ELSE

        Output "\nInvalid input. Please enter Y or N.\n"

SELESAI

Keluar dari program

## HASIL PENGUJIAN PROGRAM

### 1. F01- Register

Kondisi 1: register awal

```
Masukkan Command :
>>> register

=====

REGISTER

=====

Masukan Username: bankitganteng
Masukan Password: inipassword124

Silakan pilih salah satu monster sebagai monster awalmu!
1. Pikachow
2. Bulbu
3. Zeze
4. Zuko
5. Chacha

Monster pilihanmu: 1

Selamat datang Agent bankitganteng. Mari kita mengalahkan Dr. Asep Spakbor dengan Pikachow!
```

Gambar 2.1

Kondisi 2: jika username sudah terpakai

```
Masukkan Command :
>>> register

=====

REGISTER

=====

Masukan Username: Mr_Monogram
Masukan Password: monogrammer77

Username Mr_Monogram sudah terpakai, silakan gunakan username lain!
```

Gambar 2.2

Kondisi 3: jika sudah melakukan register

```
>>> REGISTER
Register gagal
Anda telah login dengan username bangkitganteng silakan lakukan "LOGOUT" sebelum melakukan register.
```

**Gambar 2.3**

## 2. F02 - Login

Kondisi 1: Login berhasil

```
=====
          LOGIN
=====

Username: bububobo
Password: prabowo

Selamat datang kembali, Agent bububobo. Mari lanjutkan petualangan kita!
Masukkan command "HELP" untuk daftar command yang dapat kamu panggil.
```

**Gambar 3.1**

Kondisi 2: Login ketika sudah login sebelumnya

```
Masukkan Command :
>>> login

Login gagal!
Anda telah login dengan username bububobo, silahkan lakukan "LOGOUT" sebelum melakukan login kembali.
```

**Gambar 3.2**

Kondisi 3: Login namun password salah

```
Username: bububobo
Password: nana

Password salah!
```

**Gambar 3.3**

Kondisi 4: Login ketika username tidak terdaftar

```
Username: hashu
Password: zerrek

Username tidak terdaftar!
```

**Gambar 3.4**

### 3. F03 - Logout

Kondisi 1 : logout jika sudah login

```
Masukkan Command :
>>> logout

Anda Berhasil Logout dari Akun dengan username matt.
```

**Gambar 4.1**

Kondisi 2 : logout jika belum login

```
Masukkan Command :
>>> logout

Logout gagal!
Anda belum login, silahkan lakukan "LOGIN" terlebih dahulu sebelum melakukan logout.
```

**Gambar 4.2**

### 4. F04 - Menu & Help

Kondisi 1 : sebelum login

```
Masukkan Command :
>>> help

=====

  H E L P

=====

Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu.
  1. Login: Masuk ke dalam akun yang sudah terdaftar
  2. Register: Membuat akun baru

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

**Gambar 5.1**

Kondisi 2 : setelah login sebagai agent

```
Masukkan Command :
>>> help

=====

HELP

=====

Halo Agent Agen_P. 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 owca-dex yang dimiliki oleh Agent
3. BATTLE: Bertarung melawan monster secara random
4. ARENA: Pertarungan dengan aturan seperti battle selama 5 stage
5. SHOP: Tempat Agent membeli monster dan potion menggunakan owca-coin
6. LABORATORY: Tempat Agent melakukan upgrade level monster yang dimiliki di inventory

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

**Gambar 5.2**

Kondisi 3 : setelah login sebagai admin

```
Masukkan Command :
>>> help

=====

HELP

=====

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

1. LOGOUT: Keluar dari akun yang sedang digunakan
2. SHOP: Melakukan manajemen pada SHOP sebagai tempat jual beli peralatan Agent
3. MONSTER: Mengatur manajemen monster di dalam owca-dex

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

**Gambar 5.3**

## 5. F07 - Inventory

Kondisi 1 : memanggil inventory

```

Masukkan Command :
>>> inventory

=====
INVENTORY
=====

===== INVENTORY LIST (User ID: 3) =====
Jumlah O.W.C.A Coin-mu sekarang 0.
1. Monster      (Name: Bulbu, Lvl: 2, HP: 1200)
2. Monster      (Name: Zeze, Lvl: 1, HP: 100)
3. Potion       (Type: Resilience, Qty: 7)

Ketikkan id untuk menampilkan detail item :
(Ketik "KELUAR" jika ingin keluar dari INVENTORY)

```

**Gambar 6.1**

Kondisi 2 : mengetik id barang

```

Ketikkan id untuk menampilkan detail item :
(Ketik "KELUAR" jika ingin keluar dari INVENTORY)

>>> 1

Monster
Name      : Bulbu
ATK Power : 50
DEF Power : 50
HP        : 1200
Level     : 2

Ketikkan id untuk menampilkan detail item :
(Ketik "KELUAR" jika ingin keluar dari INVENTORY)

```

**Gambar 6.2**

Kondisi 3 : keluar inventory

```

Ketikkan id untuk menampilkan detail item :
(Ketik "KELUAR" jika ingin keluar dari INVENTORY)

>>> keluar

Sampai Jumpa Lagi Admin Agen_P !!!

```

**Gambar 6.3**

## 6. F08 - Battle

Kondisi 1 :

```
Masukkan Command :
>>> battle

=====
BATTLE
=====



RAWRRR, Monster Zuko telah muncul !!!

Name      : Zuko
ATK Power : 120
DEF Power : 30
HP        : 960
Level     : 3
```

```
===== MONSTER LIST =====
1. Bulbu
2. Zeze

Pilih monster untuk bertarung: 
```

**Gambar 7.1**

Kondisi 2 : jika monster yang dipilih tidak ada



```

===== MONSTER LIST =====
1. Bulbu
2. Zeze

Pilih monster untuk bertarung: 3
Pilihan nomor tidak tersedia!

Pilih monster untuk bertarung: █

```


**Gambar 7.2**

Kondisi 3 : jika monster yang dipilih ada

```

Pilih monster untuk bertarung: 2

```



```

RAWRRR, Agent Agen_P mengeluarkan Monster Zeze !!!

Name      : Zeze
ATK Power : 300
DEF Power : 10
HP        : 100
Level     : 1

```

```

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

```

**Gambar 7.3**

Kondisi 4 : jika memilih quit

```
Pilih perintah: 3

Anda berhasil kabur dari BATTLE!
```

**Gambar 7.4**

Kondisi 5 : jika memilih potion namun tidak ada potion

```
Pilih perintah: 2

Anda tidak memiliki Potion dalam inventory!
```

**Gambar 7.5**

Kondisi 6 : jika memilih potion

```
Pilih perintah: 2

===== POTION LIST =====
1. Resilience Potion (Qty: 7) - Increases DEF Power
2. Cancel
```

**Gambar 7.6**

Kondisi 7 : jika memilih Resilience potion

```
Pilih perintah: 1

Setelah meminum ramuan ini, muncul sebuah energi pelindung di sekitar Zeze yang membuatnya terlihat semakin tangguh dan sulit dilukai.
```

**Gambar 7.7**

Kondisi 8 : jika memilih potion yang sudah dipilih

```
Pilih perintah: 1

Kamu mencoba memberikan ramuan ini kepada Zeze, namun dia menolaknya seolah-olah dia memahami ramuan tersebut sudah tidak bermanfaat lagi.
```

**Gambar 7.8**

Kondisi 9 : jika memilih potion yang tidak dimiliki

```
===== POTION LIST =====
1. Strength Potion (Qty: 0) - Increases ATK Power
2. Cancel

Pilih perintah: 1
Maaf Strength Potion kamu telah habis, silahkan beli lagi di SHOP!
```

**Gambar 7.9**

Kondisi 10 : jika memilih attack

```
Pilih perintah: 1

SHEESHSHHHH, Zeze menyerang Zuko !!!

Name      : Zuko
ATK Power : 120
DEF Power : 30
HP        : 784
Level     : 3
-----
Penjelasan: ATT: 251 (-16.33%), Reduced By: 75.30 (30%), ATT Results: 175.70

===== TURN 1 (Zuko) =====

SHEESHSHHHH, Zuko menyerang Zeze !!!

Name      : Zeze
ATK Power : 300
DEF Power : 10.5
HP        : 14
Level     : 1
-----
Penjelasan: ATT: 96 (-20.00%), Reduced By: 10.08 (10.5%), ATT Results: 85.92
```

**Gambar 7.10**

Kondisi 11 : WIN

```
Pilih perintah: 1

SHEESHSHHHH, Bulbu menyerang Zeze !!!

Name      : Zeze
ATK Power : 360
DEF Power : 12
HP        : 0
Level     : 3
-----
Penjelasan: ATT: 67 (21.82%), Reduced By: 8.04 (12%), ATT Results: 58.96

Selamat, Anda berhasil mengalahkan monster Zeze !!!

Total OC yang diperoleh: 29
```

**Gambar 7.11**

Kondisi 12 : LOSE

```

Pilih perintah: 1

SHEESHHHHH, Zeze menyerang Zuko !!!

Name      : Zuko
ATK Power : 120
DEF Power : 30
HP        : 547
Level     : 3
-----
Penjelasan: ATT: 338 (12.67%), Reduced By: 101.40 (30%), ATT Results: 236.60

===== TURN 2 (Zuko) =====

SHEESHHHHH, Zuko menyerang Zeze !!!

Name      : Zeze
ATK Power : 300
DEF Power : 10.5
HP        : 0
Level     : 1
-----
Penjelasan: ATT: 111 (-7.50%), Reduced By: 11.65 (10.5%), ATT Results: 99.34

Yahhh, Anda dikalahkan monster Zuko. Jangan menyerah, coba lagi !!!

```

**Gambar 7.12**

## 7. F09 - Arena

Kondisi 1 : masuk Arena

```

Masukkan Command :
>>> arena

=====
  ARENA
=====

Selamat Datang di Arena!!

===== MONSTER LIST =====
1. Bulbu
2. Zeze

```

**Gambar 8.1**

Kondisi 2 : jika monster yang dipilih tidak ada

```

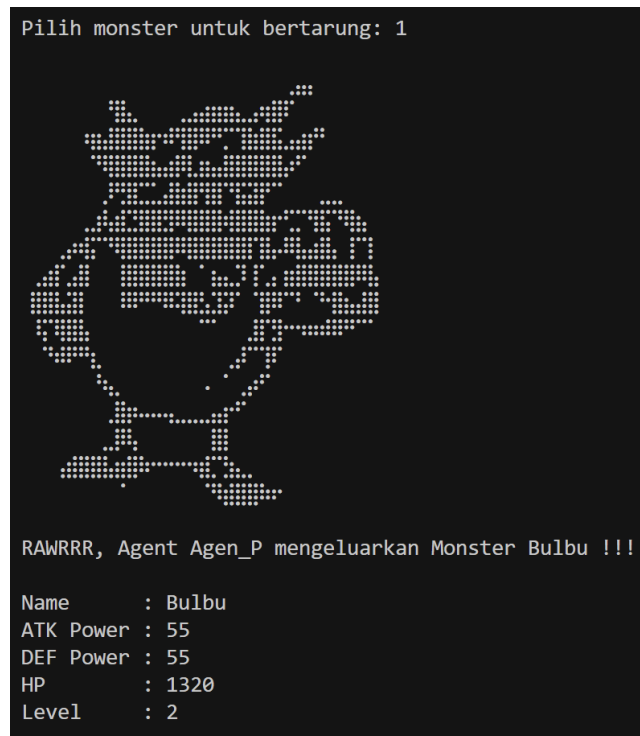
===== MONSTER LIST =====
1. Bulbu
2. Zeze

Pilih monster untuk bertarung: 3
Pilihan nomor tidak tersedia!

```

### Gambar 8.2

Kondisi 3 : jika monster yang dipilih ada



**Gambar 8.3**

Kondisi 4 : mulai stage 1

```
===== STAGE 1 =====  
  
        
  
RAWRRR, Monster Zeze telah muncul !!!  
  
Name      : Zeze  
ATK Power : 300  
DEF Power : 10  
HP        : 100  
Level     : 1  
  
===== TURN 1 (Bulbu) =====  
1. Attack  
2. Use Potion  
3. Quit
```

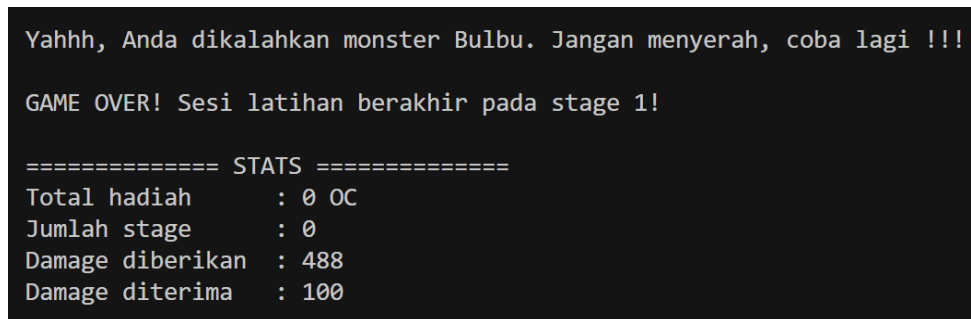
**Gambar 8.4**

Kondisi 5 : menang stage 1



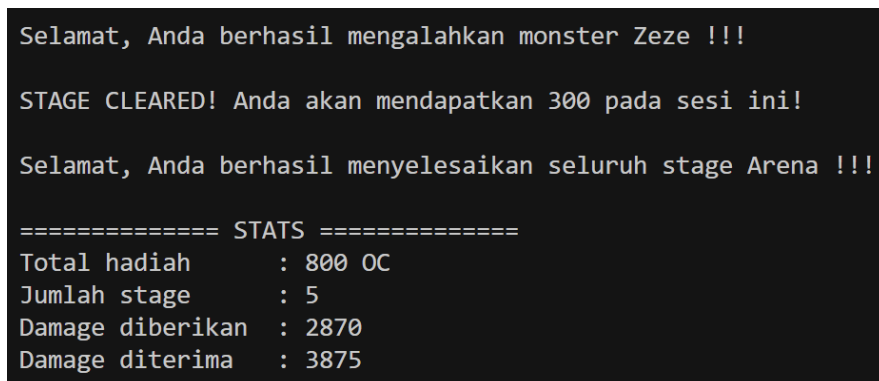
**Gambar 8.5**

Kondisi 6 : jika agent kalah



**Gambar 8.6**

Kondisi 7 : jika menang stage 5



**Gambar 8.7**

Kondisi 8 : memilih quit

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

Pilih perintah: 3

GAME OVER! Anda mengakhiri sesi latihan!!

===== STATS =====
Total hadiah      : 0 OC
Jumlah stage      : 0
Damage diberikan  : 0
Damage diterima   : 0
```

**Gambar 8.8**

## 8. F10 - Shop & Currency

Kondisi 1 : masuk shop

```
Masukkan Command :
>>> shop

=====

  SHOP

=====

Selamat datang di Mr. Yanto SHOP !!!

>>> Pilih aksi (LIHAT/BELI/KELUAR): █
```

**Gambar 9.1**

Kondisi 2 : melihat monster

```
>>> Pilih aksi (LIHAT/BELI/KELUAR): lihat

>>> Mau lihat apa? (MONSTER/POTION): monster

ID | Type           | ATK Power | DEF Power | HP   | Stok | Harga
1  | Pikachuow      | 125       | 10        | 600  | 10   | 500
3  | Zeze           | 300       | 10        | 100  | 3    | 1000
5  | Chacha         | 80        | 30        | 700  | 7    | 600

>>> Pilih aksi (LIHAT/BELI/KELUAR): █
```



**Gambar 9.2**

Kondisi 3 : melihat potion

```
>>> Mau lihat apa? (MONSTER/POTION): potion

ID | Type                | Stok | Harga
1  | Strength Potion     | 10   | 50
```

**Gambar 9.3**

Kondisi 4 : membeli monster

```
>>> Pilih aksi (LIHAT/BELI/KELUAR): beli

Jumlah O.W.C.A. Coin-mu sekarang 875.

>>> Mau beli apa? (MONSTER/POTION): monster

ID | Type        | ATK Power | DEF Power | HP | Stok | Harga
1  | Pikachu    | 125       | 10        | 600 | 10   | 500
3  | Zeze       | 300       | 10        | 100 | 3    | 1000
5  | Chacha     | 80        | 30        | 700 | 7    | 600

>>> Masukkan id monster: 1

Berhasil membeli item: Pikachu. Item sudah masuk ke inventory-mu!
```

**Gambar 9.4**

Kondisi 5 : membeli potion

```
>>> Pilih aksi (LIHAT/BELI/KELUAR): beli

Jumlah O.W.C.A. Coin-mu sekarang 375.

>>> Mau beli apa? (MONSTER/POTION): potion

ID | Type                | Stok | Harga
1  | Strength Potion     | 10   | 50

>>> Masukkan id potion: 1

>>> Masukkan jumlah: 2

Berhasil membeli item: 2 strength Potion. Item sudah masuk ke inventory-mu!
```

**Gambar 9.5**

Kondisi 6 : membeli: gagal memvalidasi

```
>>> Pilih aksi (LIHAT/BELI/KELUAR): beli

Jumlah O.W.C.A. Coin-mu sekarang 275.

>>> Mau beli apa? (MONSTER/POTION): monster
```

ID	Type	ATK Power	DEF Power	HP	Stok	Harga
1	Pikachow	125	10	600	9	500
3	Zeze	300	10	100	3	1000
5	Chacha	80	30	700	7	600

```
>>> Masukkan id monster: 5

O.W.C.A Coin Anda tidak cukup untuk membeli Chacha.
```

**Gambar 9.6**

Kondisi 7 : membeli: gagal dalam validasi inventory

```
>>> Mau beli apa? (MONSTER/POTION): monster
```

ID	Type	ATK Power	DEF Power	HP	Stok	Harga
1	Pikachow	125	10	600	9	500
3	Zeze	300	10	100	3	1000
5	Chacha	80	30	700	7	600

```
>>> Masukkan id monster: 1

Monster Pikachow sudah ada dalam inventory-mu! Pembelian dibatalkan.
```

**Gambar 9.7**

Kondisi 8 : keluar

```
>>> Pilih aksi (LIHAT/BELI/KELUAR): keluar

Terima kasih telah berkunjung di Mr. Yanto SHOP, sering-sering ya belanja disini !!!
```

**Gambar 9.8**

## 9. F11 - Laboratory

Kondisi 1 : masuk laboratory

```

Masukkan Command :
>>> laboratory

=====
LABORATORY
=====

Selamat Datang di Laboratory O.W.C.A!

===== UPGRADE PRICE =====
1. Level 1 -> Level 2: 200 OC
2. Level 2 -> Level 3: 400 OC
3. Level 3 -> Level 4: 600 OC
4. Level 4 -> Level 5: 800 OC

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

Pilih monster: 

```

**Gambar 10.1**

Kondisi 2 : memilih monster untuk di-upgrade

```

Pilih monster: 1

Bulbu akan di-upgrade ke level 3.
Harga untuk melakukan upgrade Bulbu adalah 400 OC.

Lanjutkan upgrade (Y/N): y

Selamat, Bulbu berhasil di-upgrade ke level 3 !

```

**Gambar 10.2**

Kondisi 3 : meng-upgrade monster level 5

```

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

Pilih monster: 1

Maaf, monster yang Anda pilih sudah memiliki level maksimum

```

**Gambar 10.3**

## 10. F12 - Shop Management

Kondisi 1: Memanggil command shop

```
Masukkan Command :
>>> shop

=====
SHOP
=====

Selamat datang di Mr. Yanto SHOP !!!

>>> Pilih aksi (LIHAT/BELI/KELUAR):
```

Gambar 11.1

Kondisi 2.1: Melihat monster

```
>>> Pilih aksi (LIHAT/BELI/KELUAR): lihat

>>> Mau lihat apa? (MONSTER/POTION): monster

ID | Type           | ATK Power | DEF Power | HP   | Stok | Harga
1  | Pikachuow      | 125       | 10        | 600  | 10   | 500
2  | Bulbu          | 50        | 50        | 1200 | 4    | 700
3  | Zeze           | 300       | 10        | 100  | 3    | 1000
4  | Zuko           | 100       | 25        | 800  | 8    | 550
5  | Chacha         | 80        | 30        | 700  | 7    | 600
```

Gambar 11.2

Kondisi 2.2: Membeli monster namun sudah ada di inventory

```
>>> Pilih aksi (LIHAT/BELI/KELUAR): beli

Jumlah O.W.C.A. Coin-mu sekarang 9999.

>>> Mau beli apa? (MONSTER/POTION): monster

ID | Type           | ATK Power | DEF Power | HP   | Stok | Harga
1  | Pikachuow      | 125       | 10        | 600  | 10   | 500
2  | Bulbu          | 50        | 50        | 1200 | 4    | 700
3  | Zeze           | 300       | 10        | 100  | 3    | 1000
4  | Zuko           | 100       | 25        | 800  | 8    | 550
5  | Chacha         | 80        | 30        | 700  | 7    | 600

>>> Masukkan id monster: 1

Monster Pikachuow sudah ada dalam inventory-mu! Pembelian dibatalkan.
```

Gambar 11.3

Kondisi 2.3: Berhasil membeli monster

```
>>> Pilih aksi (LIHAT/BELI/KELUAR): beli

Jumlah O.W.C.A. Coin-mu sekarang 9999.

>>> Mau beli apa? (MONSTER/POTION): monster
```

ID	Type	ATK Power	DEF Power	HP	Stok	Harga
1	Pikachow	125	10	600	10	500
2	Bulbu	50	50	1200	4	700
3	Zeze	300	10	100	3	1000
4	Zuko	100	25	800	8	550
5	Chacha	80	30	700	7	600

```
>>> Masukkan id monster: 5

Berhasil membeli item: Chacha. Item sudah masuk ke inventory-mu!
```

**Gambar 11.4**

Kondisi 3.1: Melihat potion

```
>>> Pilih aksi (LIHAT/BELI/KELUAR): lihat

>>> Mau lihat apa? (MONSTER/POTION): potion
```

ID	Type	Stok	Harga
1	Strength Potion	10	50
2	Resilience Potion	5	30
3	Healing Potion	3	20

**Gambar 11.5**

Kondisi 3.2: Coin tidak cukup

```
>>> Pilih aksi (LIHAT/BELI/KELUAR): beli

Jumlah O.W.C.A. Coin-mu sekarang 8.

>>> Mau beli apa? (MONSTER/POTION): potion
```

ID	Type	Stok	Harga
1	Strength Potion	9	50
2	Resilience Potion	5	30
3	Healing Potion	3	20

```
>>> Masukkan id potion: 1

>>> Masukkan jumlah: 1

O.W.C.A Coin Anda tidak cukup untuk membeli 1 strength Potion.
```

**Gambar 11.6**

Kondisi 3.2: Berhasil membeli potion

```
>>> Pilih aksi (LIHAT/BELI/KELUAR): beli

Jumlah O.W.C.A. Coin-mu sekarang 9399.

>>> Mau beli apa? (MONSTER/POTION): potion

ID | Type                | Stok | Harga
1  | Strength Potion     | 10   | 50
2  | Resilience Potion  | 5    | 30
3  | Healing Potion      | 3    | 20

>>> Masukkan id potion: 1

>>> Masukkan jumlah: 1

Berhasil membeli item: 1 strength Potion. Item sudah masuk ke inventory-mu!
```

**Gambar 11.7**

Kondisi 4.1: Menambah monster sebagai admin

```
>>> Pilih aksi (LIHAT/TAMBAH/UBAH/HAPUS/KELUAR): tambah

>>> Mau nambahin apa? (MONSTER/POTION): monster

ID | Type      | ATK Power | DEF Power | HP
5  | Chacha    | 80        | 30        | 700

>>> Masukkan id monster: 5

>>> Masukkan stok awal: 20

>>> Masukkan harga: 600

Chacha telah berhasil ditambahkan ke dalam shop!
```

**Gambar 11.8**

Kondisi 4.2: Mengubah stok dan harga monster sebagai admin

```
>>> Pilih aksi (LIHAT/TAMBAH/UBAH/HAPUS/KELUAR): ubah

>>> Mau ubah apa? (MONSTER/POTION): monster

ID | Type      | ATK Power | DEF Power | HP | Stok | Harga
1  | Pikachu   | 125       | 10        | 600 | 10   | 500
2  | Bulbu     | 50        | 50        | 1200 | 4    | 700
3  | Zeze      | 300       | 10        | 100 | 3    | 1000
4  | Zuko      | 100       | 25        | 800 | 8    | 550
5  | Chacha    | 80        | 30        | 700 | 6    | 600

>>> Masukkan id monster: 3

>>> Masukkan stok baru: 5

>>> Masukkan harga baru: 700

Zeze telah berhasil diubah dengan stok baru sejumlah 5 dan dengan harga baru 700
```

**Gambar 11.9**

Kondisi 4.3: Menghapus monster sebagai admin

```
>>> Pilih aksi (LIHAT/TAMBAH/UBAH/HAPUS/KELUAR): hapus

>>> Mau hapus apa? (MONSTER/POTION): monster

ID | Type           | ATK Power | DEF Power | HP   | Stok | Harga
1  | Pikachu        | 125       | 10        | 600  | 10   | 500
2  | Bulbu          | 50        | 50        | 1200 | 4    | 700
3  | Zeze           | 300       | 10        | 100  | 5    | 700
4  | Zuko           | 100       | 25        | 800  | 8    | 550
5  | Chacha         | 80        | 30        | 700  | 6    | 600

>>> Masukkan id monster: 5

>>> Apakah anda yakin ingin menghapus Chacha dari shop (Y/N)? Y

Chacha telah berhasil dihapus dari shop!
```

**Gambar 11.10**

Kondisi 5.1: Menambah potion sebagai admin

```
>>> Pilih aksi (LIHAT/TAMBAH/UBAH/HAPUS/KELUAR): tambah

>>> Mau nambahin apa? (MONSTER/POTION): potion

ID | Type
3  | Resilience Potion

>>> Masukkan id potion: 3

>>> Masukkan stok awal: 5

>>> Masukkan harga: 30
Resilience Potion telah berhasil ditambahkan ke dalam shop!
```

**Gambar 11.11**

Kondisi 5.2: Mengubah potion sebagai admin

```
>>> Pilih aksi (LIHAT/TAMBAH/UBAH/HAPUS/KELUAR): ubah

>>> Mau ubah apa? (MONSTER/POTION): potion

ID | Type           | Stok | Harga
1  | Strength Potion | 6    | 50
2  | Resilience Potion | 4    | 30
3  | Healing Potion  | 3    | 20

>>> Masukkan id potion: 3

>>> Masukkan stok baru: 5

>>> Masukkan harga baru: 20

Healing Potion telah berhasil diubah dengan stok baru sejumlah 5 dan dengan harga baru 20
```

**Gambar 11.12**

Kondisi 5.3: Menghapus potion sebagai admin

```

>>> Pilih aksi (LIHAT/TAMBAH/UBAH/HAPUS/KELUAR): hapus

>>> Mau hapus apa? (MONSTER/POTION): potion

ID | Type                | Stok | Harga
1  | Strength Potion     | 6    | 50
2  | Resilience Potion  | 4    | 30
3  | Healing Potion      | 5    | 20

>>> Masukkan id potion: 2

>>> Apakah anda yakin ingin menghapus Resilience Potion dari shop (y/n)? y

Resilience Potion telah berhasil dihapus dari shop!

```

Gambar 11.13

## 11. F13 - Monster Management

Kondisi 1 : Menampilkan semua monster

```

Masukkan Command :
>>> monster

=====
MONSTER MANAGEMENT
=====

SELAMAT DATANG DI DATABASE PARA MONSTER Mr_Monogram !!!

1. Tampilkan Semua Monster
2. Tambah Monster baru
3. Keluar

>>> Pilih Aksi (1/2/3): 1

ID | Type      | ATK Power | DEF Power | HP |
1  | Pikachu  | 125       | 10        | 600 |
2  | Bulbu    | 50        | 50        | 1200 |
3  | Zeze     | 300       | 10        | 100 |
4  | Zuko     | 100       | 25        | 800 |
5  | Chacha   | 80        | 30        | 700 |

```

Gambar 12.1

Kondisi 2: Menambah monster baru beserta output jika input salah



```

>>> Pilih Aksi (1/2/3): 2

Memulai pembuatan monster baru

>>> Masukkan Type / Nama : zuko
Nama sudah terdaftar, coba lagi!

>>> Masukkan Type / Nama : coco

>>> Masukkan ATK Power : api
Masukkan input bertipe Integer, coba lagi!

>>> Masukkan ATK Power : 250

>>> Masukkan DEF Power (0-50) : 200
DEF Power harus bernilai 0-50, coba lagi!

>>> Masukkan DEF Power (0-50) : 25

>>> Masukkan HP : 200

Monster baru berhasil dibuat!
Type          : Coco
ATK Power     : 250
DEF Power     : 25
HP            : 200

>>> Tambahkan Monster ke database (Y/N) : Y

Monster baru telah ditambahkan!

```

**Gambar 12.2**

```

>>> Pilih Aksi (1/2/3): 1

```

ID	Type	ATK Power	DEF Power	HP
1	Pikachow	125	10	600
2	Bulbu	50	50	1200
3	Zeze	300	10	100
4	Zuko	100	25	800
5	Chacha	80	30	700
7	Coco	250	25	200

**Gambar 12.3**

```

Monster baru berhasil dibuat!
Type          : Coco
ATK Power     : 250
DEF Power     : 45
HP            : 0

>>> Tambahkan Monster ke database (Y/N) : n

```

**Gambar 12.4**

## 12. F14 - Load

Kondisi 1 : load sukses

```
PS C:\Users\ASUS\OneDrive\Documents\Tugas\Sem 2\if1210-2024-tubes-k05-d\if1210-2024-tubes-k05-d> py main.py Default
Loading...
Selamat datang di program OWCA!

=====
WELCOME TO OWCA!
=====

Masukkan Command :
>>> 
```

**Gambar 12.1**

Kondisi 2 : tidak diberi nama folder

```
PS C:\Users\ASUS\OneDrive\Documents\Tugas\Sem 2\if1210-2024-tubes-k05-d\if1210-2024-tubes-k05-d> py main.py

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

**Gambar 12.2**

Kondisi 3 : menginput folder palsu

```
PS C:\Users\ASUS\OneDrive\Documents\Tugas\Sem 2\if1210-2024-tubes-k05-d\if1210-2024-tubes-k05-d> py main.py folderpalsu

Folder "folderpalsu" tidak ditemukan.
```

**Gambar 12.3**

### 13. F15 - Save

Kondisi 1 : berhasil menyimpan file

```
Masukkan Command :
>>> save

Masukkan nama folder : 20/05/2024

Saving...

Membuat folder data/20/05/2024...
Berhasil menyimpan data di folder data/20/05/2024!
```

**Gambar 13.1**

Kondisi 2 : menyimpan di folder yang sudah ada

```
Masukkan Command :  
>>> save  
  
Masukkan nama folder : 20/05/2024  
Folder data/20/05/2024 sudah ada.  
  
Saving...  
  
Berhasil menyimpan data di folder data/20/05/2024!
```

**Gambar 13.2**

#### **14. F16 - Exit**

Kondisi 1 : input tidak valid

```
Masukkan Command :  
>>> exit  
  
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (Y/N): a  
  
Invalid input. Please enter Y or N.  
  
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (Y/N): y  
  
Saving...
```

**Gambar 14.1**

Kondisi 2 : tidak menyimpan file

```
Masukkan Command :  
>>> exit  
  
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (Y/N): n  
  
Yahh progress kamu hilang, selamat tinggal :(
```

**Gambar 14.2**

## LAMPIRAN

 Form MoM Asistensi 1 K-05-D.docx

 Form MoM Asistensi 2 K-05-D.docx