

**Laporan Tugas Besar  
IF1210 Dasar Pemrograman  
K10-D  
2023/2024**



Kelompok D:

Kesha Mufrih Ramadhan (16523009)  
Anak Agung Gede Aditya (16523039)  
Nisrina Azzah Atiqoh (16523229)  
Haegen Quinston (19623009)  
Muhammad Rayhan Farrukh (19623119)

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

## Pernyataan Kelompok

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

Yang mengeluarkan pernyataan,

Kesha Mufrih Ramadhan (16523009)

Anak Agung Gede Aditya (16523039)

Nisrina Azzah Atiqoh (16523229)

Haegen Quinston (19623009)

Muhammad Rayhan Farrukh (19623119)

## Daftar Isi

<b>Pernyataan Kelompok.....</b>	<b>1</b>
<b>Daftar Isi.....</b>	<b>2</b>
<b>Daftar Tabel.....</b>	<b>4</b>
<b>Daftar Gambar.....</b>	<b>5</b>
<b>I. Deskripsi Persoalan.....</b>	<b>6</b>
<b>II. Daftar Pembagian Kerja.....</b>	<b>9</b>
<b>III. Checklist Hasil Kerja.....</b>	<b>11</b>
<b>IV. Spesifikasi, Desain Kamus Data, dan Desain Dekomposisi Fungsi Program.....</b>	<b>13</b>
a. Main.....	13
b. F00 - Random Number Generator.....	15
c. F01 - Register.....	16
d. F02 - Login.....	17
e. F03 - Logout.....	18
f. F04 - Menu & Help.....	19
g. F05 - Monster.....	20
h. F06 - Potion.....	21
i. F07 - Inventory.....	23
j. F08 - Battle.....	23
k. F09 - Arena.....	28
l. F10 - Shop & Currency (Agent).....	29
m. F11 - Monster Laboratory.....	32
n. F12 - Shop (Admin).....	34
o. F13 - Monster Management.....	47
p. F14 - Load.....	51
q. F15 - Save.....	53
r. F16 - Exit.....	55
<b>V. Hasil Testing Program.....</b>	<b>57</b>
<b>Lampiran.....</b>	<b>70</b>

## Daftar Tabel

Tabel 1. Struktur data pada file user.csv.....	6
Tabel 2. Struktur data pada file monster.csv.....	7
Tabel 3. Struktur data pada file item_inventory.csv.....	7
Tabel 4. Struktur data pada file monster_inventory.csv.....	7
Tabel 6. Struktur data pada file monster_shop.csv.....	8
Tabel 7. Pembagian Kerja.....	9
Tabel 8. Checklist Hasil Kerja.....	11

## Daftar Gambar

Gambar V.1 Register tidak valid.....	57
Gambar V.2 Register valid.....	57
Gambar V.3 Tampilan berhasil login.....	58
Gambar V.4 Prosedur logout.....	58
Gambar V.5 Tampilan menu untuk agent.....	58
Gambar V.6 Tampilan menu untuk admin.....	59
Gambar V.7 Tampilan potion saat tidak ada stok.....	59
Gambar V.8 Tampilan Potion ketika digunakan pada battle.....	59
Gambar V.9 Tampilan awal inventory.....	60
Gambar V.10 Fitur inspect inventory.....	60
Gambar V.11 Input tidak valid dan keluar dari inventory.....	60
Gambar V.12 Tampilan battle.....	61
Gambar V.13 Keluar battle.....	61
Gambar V.14 Tampilan awal arena.....	62
Gambar V.15 Tampilan Stage Cleared.....	62
Gambar V.16 Statistik arena.....	62
Gambar V.17 Tampilan shop awal (agent).....	63
Gambar V.18 Melihat monster pada shop.....	63
Gambar V.19 Melihat item pada shop.....	64
Gambar V.20 Membeli dengan O.W.C.A Coin yang tidak cukup.....	64
Gambar V.21 Keluar dari shop.....	64
Gambar V.22 Tampilan awal laboratory.....	64
Gambar V.23 Mengupgrade dengan O.W.C.A Coin tidak cukup.....	65
Gambar V.24 Keluar dari laboratory.....	65
Gambar V.25 Memilih aksi di shop management.....	65
Gambar V.26 Melihat monster di shop management.....	66
Gambar V.27 Menambahkan stok item suatu monster.....	66
Gambar V.28 Mengubah stok item di shop.....	67
Gambar V.29 Menghapus monster.....	67
Gambar V.30 Menu monster management.....	67
Gambar V.31 Aksi melihat monster.....	68
Gambar V.32 Membuat monster baru.....	68
Gambar V.33 Keluar dari monster.....	68
Gambar V.34 Load yang berhasil mencetak kata kata selamat datang.....	69
Gambar V.35 Load gagal karena salah run.....	69
Gambar V.36 Prosedur penyimpanan save.....	69
Gambar V.37 Keluar dari program (prosedur).....	69

## I. Deskripsi Persoalan

Tugas besar ini meminta untuk membuat program sebuah permainan untuk mengalahkan Dr. Asep Spakbor, seorang ilmuwan jahat yang telah menciptakan monster-monster mengerikan yang mengancam keamanan kota Danville. Purry si Platypus, dikenal juga sebagai Agent P, membutuhkan bantuan untuk melawan monster buatan Dr. Asep Spakbor.

Agent P membutuhkan kita untuk melatih monster-monster yang ada di hutan. Kita diberikan satu monster untuk dilatih sampai level maksimal 5. Monster dilatih dengan melawan monster lain yang muncul menggunakan algoritma Random Number Generator (RNG). Monster yang dilatih dapat diberi potion untuk meningkatkan kemampuan bertarung. Setiap agent yang melatih monster memiliki O.W.C.A Coin untuk membeli potion, agent mendapatkan tambahan O.W.C.A Coin jika berhasil mengalahkan monster lain.

Program yang dibuat harus sesuai dengan spesifikasi yang diberikan pada [IF1210 Spesifikasi Tugas Besar](#) dan dikolaborasikan dengan kreativitas masing-masing.

Data-data yang digunakan dalam permainan berupa file-file csv yang berisi data user yang sudah tersedia, data monster, inventory atau penyimpanan dari dunia game tersebut, dan data shop. Untuk struktur data yang digunakan dalam program ini yaitu:

### 1. User.csv

Tabel 1. Struktur data pada file user.csv

id	Id berupa integer yang menunjukkan banyak akun yang ada
username	Nama dari akun yang telah dibuat untuk login
password	Password untuk verifikasi kepemilikan akun untuk login
role	Role sesuai dari yang dipilih pengguna untuk akses fitur yang dapat digunakan
oc (owca coin)	Merupakan currency yang dapat digunakan dalam game

### 2. Monster.csv

Tabel 2. Struktur data pada file monster.csv

id	Id berupa integer yang menunjukkan banyak monster yang ada dan jenis dari monster
type	Merupakan type dari monster
atk_power	Attack power dari monster tersebut
def_power	Defense power dari monster tersebut
hp	Merupakan hitpoint dari monster tersebut

### 3. Item\_inventory.csv

Tabel 3. Struktur data pada file item\_inventory.csv

user_id	User id merupakan id dari user yang sudah disediakan ataupun digenerate pada saat pembuatan akun
type	Type dari item yang kita miliki dalam game
quantity	Banyak dari item yang kita miliki

### 4. Monster\_inventory.csv

Tabel 4. Struktur data pada file monster\_inventory.csv

user_id	User id merupakan id dari user yang sudah disediakan ataupun digenerate pada saat pembuatan akun
monster_id	Monster id merupakan id dari jenis monster yang dimiliki ini bisa sudah disediakan jika ada akun yang tersedia ataupun akan digenerate pada saat akun tersebut memiliki monster
level	Level dari monster tersebut

### 5. Item\_shop.csv

Tabel 5. Struktur data pada file item\_shop.csv

type	Type dari item yang ada di dalam game
stock	Stock atau ketersediaan dari item yang ada
price	Harga dari item yang tersedia

## 6. Monster\_shop.csv

Tabel 6. Struktur data pada file monster\_shop.csv

monster_id	Monster id merupakan id dari jenis monster yang dimiliki ini bisa sudah disediakan jika ada akun yang tersedia ataupun akan digenerate pada saat akun tersebut memiliki monster
stock	Stock atau ketersediaan dari monster yang ada
price	Harga dari monster yang tersedia



## II. Daftar Pembagian Kerja

Daftar pembagian kerja anggota kelompok, berdasarkan fitur yang dibuat, meliputi desain, implementasi, dan *testing*.

Tabel 7. Pembagian Kerja

Tugas	Desainer	Coder	Tester
Main	Haegen ▾	Haegen ▾	Kesha ▾
F00	Kesha ▾	Kesha ▾	Kesha ▾ - ▾
F01	Nisrina ▾ - ▾	Nisrina ▾ - ▾	Nisrina ▾ - ▾
F02	Nisrina ▾ - ▾	Nisrina ▾ - ▾	Nisrina ▾ - ▾
F03	Nisrina ▾ - ▾	Nisrina ▾ - ▾	Nisrina ▾ - ▾
F04	Haegen ▾ - ▾	Haegen ▾ - ▾	Haegen ▾ - ▾
F05	Haegen ▾ - ▾	Haegen ▾ - ▾	Haegen ▾ - ▾
F06	Haegen ▾ - ▾	Haegen ▾ - ▾	Haegen ▾ - ▾
F07	Haegen ▾ - ▾	Haegen ▾ - ▾	Haegen ▾ - ▾
F08	Kesha ▾ - ▾	Kesha ▾ - ▾	Haegen ▾ - ▾
F09	Kesha ▾ - ▾	Haegen ▾ - ▾	Haegen ▾ - ▾
F10	Rayhan ▾ - ▾	Rayhan ▾ - ▾	Rayhan ▾ - ▾
F11	Rayhan ▾ - ▾	Rayhan ▾ - ▾	Rayhan ▾ - ▾
F12	Rayhan ▾ - ▾	Rayhan ▾ - ▾	Rayhan ▾ - ▾
F13	Rayhan ▾ - ▾	Rayhan ▾ - ▾	Rayhan ▾ - ▾
F14	Aditya ▾ - ▾	Aditya ▾ - ▾	Aditya ▾ - ▾
F15	Aditya ▾ - ▾	Aditya ▾ - ▾	Aditya ▾ - ▾
F16	Aditya ▾ - ▾	Aditya ▾ - ▾	Aditya ▾ - ▾

Tugas	Desainer	Coder	Tester
B01	Kesha ▾ - ▾	Kesha ▾ - ▾	Kesha ▾ - ▾
B02	Kesha ▾ - ▾	Kesha ▾ - ▾	Kesha ▾ - ▾
B03	- ▾ - ▾	- ▾ - ▾	- ▾ - ▾
B04	- ▾ - ▾	- ▾ - ▾	- ▾ - ▾
B05	- ▾ - ▾	- ▾ - ▾	- ▾ - ▾

### III. Checklist Hasil Kerja

Checklist hasil rancangan, implementasi, dan testing setiap primitif.

Tabel 8. Checklist Hasil Kerja

Fitur	Desain	Implementasi	Testing
Main	✓	✓	✓
F00	✓	✓	✓
F01 - Register	✓	✓	✓
F02 - Login	✓	✓	✓
F03 - Logout	✓	✓	✓
F04 - Menu & Help	✓	✓	✓
F05 - Monster	✓	✓	✓
F06 - Potion	✓	✓	✓
F07 - Inventory	✓	✓	✓
F08 - Battle	✓	✓	✓
F09 - Arena	✓	✓	✓
F10 - Shop & Currency	✓	✓	✓
F11 - Laboratory	✓	✓	✓
F12 _ Shop Management	✓	✓	✓

Fitur	Desain	Implementasi	Testing
F13 - Monster Management	✓	✓	✓
F14 - Load	✓	✓	✓
F15 - Save	✓	✓	✓
F16 - Exit	✓	✓	✓
B01 - Git Best Practice	✓	✓	✓
B02 - Typing	✓	✓	✓
B03			
B04			
B05			

## IV. Spesifikasi, Desain Kamus Data, dan Desain Dekomposisi Fungsi Program

### a. Main

#### KAMUS

actionAwal : string

Action2 : string

#### ALGORITMA

```
repeat
    input (actionAwal)
    if (actionAwal = "register") then
        user_id, user_db, monsterInventory_db,
status_login <- register.register(user_db, monster_db,
monsterInventory_db)
    else if (actionAwal = "login") then
        output()
        user_id, status_login <- login.login(user_db)
    else if (actionAwal == "help") then
        helpMenu <- menu.help(status_login)
        output(helpMenu)
    else if (actionAwal = 'keluar') then
        exit
    else
        output("Command tidak valid")

owca_coin <- int(user_db[user_id-1][4])
repeat
    monster_user <-
functions.inventoryUser(monsterInventory_db, str(user_id))
    inventory_user <-
functions.inventoryUser(itemInventory_db, str(user_id))
    action2 <- str.lower(input(functions.clr('>>>
', 'cyan', '')))
    if (action2 = "logout") then
        status_login <- logout.logout(status_login)
        if (status_login = "") then
            break
    else if (action2 = "inventory") then
        if (status_login = "agent") then
            inventory.inventory(monster_user,
inventory_user, monster_db, owca_coin, user_id)
        else
```

```

        output("Anda tidak memiliki akses ke
inventory!")
    else if (action2 = "battle") then
        if (status_login = "agent") then
            owca_coin, inventory_user <-
battle.battle(owca_coin, monster_user, monster_db,
inventory_user)
        else
            output("Anda tidak memiliki akses ke
battle!")
    else if (action2 = "help") then
        helpMenu <- menu.help(status_login)
        output(helpMenu)
    else if (action2 = "arena") then
        if (status_login = "agent") then
            owca_coin, inventory_user <-
arena.arena(owca_coin, monster_user, monster_db,
inventory_user)
        else
            output("Anda tidak memiliki akses ke arena!")
    else if (action2 = "laboratory" then
        if status_login == "agent" then
            monsterInventory_db, owca_coin <-
laboratory.laboratory(monster_db, monsterInventory_db,
user_db, user_id, owca_coin)
        else
            output("Anda tidak memiliki akses ke
laboratory!")
    else if action2 = "shop" then
        if status_login == "agent" then
            monster_db, itemInventory_db, itemShop_db,
monsterInventory_db, monsterShop_db, owca_coin <-
shop.shop(user_db, monster_db, itemInventory_db, itemShop_db,
monsterInventory_db, monsterShop_db, user_id, owca_coin)
        else
            monster_db, monsterShop_db, itemShop_db <-
shop_management.shopManagement(monster_db, monsterShop_db,
itemShop_db)
    else if action2 == "monster" then
        if status_login == "admin" then
            monster_db, monsterShop_db <-
monster_management.monsterManagement(monster_db,
monsterShop_db)
        else

```

```

        output("Anda tidak memiliki akses ke monster
management!")
    else if action2 == "save" then
        user_db[user_id-1][4] <- owca_coin
        save.save(user_db, monster_db, monsterShop_db,
monsterInventory_db, itemShop_db, itemInventory_db)
    else if action2 == "exit" then
        game_status <- exit_.exit_(user_db, monster_db,
monsterShop_db, monsterInventory_db, itemShop_db,
itemInventory_db)
        if not game_status then
            output('\\nKeluar dari program...\\n')
            exit()
    else
        output("Input tidak valid, silakan ulangi!\\n")

```

#### b. F00 - Random Number Generator

##### **KAMUS LOKAL**

a, c, m, seed, x\_prev, x0 : integer  
numRange = list

**function** formula (a : integer, c : integer, m : integer, seed : integer, x0 : integer, x\_prev : integer) -> integer

**function** randomNumberGenerator (a : integer, c : integer, m : integer, x0 : integer, x\_prev : integer, numRange = list) -> integer

{I.S List range dari angka yang akan keluar secara acak}

{F.S Mengeluarkan angka "pilihan" dari list secara acak}

{Perubahan :

    x\_prev -> Angka acak dari numRange atau x\_prev sendiri  
jika range kosong}

{Sebuah rumus yang akan memproses x\_prev dengan konstanta a, c, m. Setelah diproses oleh rumus pertama, maka akan diproses oleh rumus kedua yang menggunakan variabel x\_prev lagi dengan m, elemen awal numRange, dan elemen akhir numRange untuk mendapatkan angka "acak" yang diinginkan}

##### **ALGORITMA**

**function** formula(a, c, m, seed)  
    x0 <- seed  
    x\_prev <- (a \* x0 + c) % m  
    -> x\_prev

```

function randomNumberGenerator(x_prev, numRange)
  a <- 4871
  c <- 0
  m <- 2^31-1
  x_prev <- (a * x_prev + c) % m
  if numRange is None then
    -> x_prev
  else
    -> int((x_prev / m) * (numRange[1] - numRange[0]) +
numRange[0])

```

#### c. F01-Register

##### **KAMUS LOKAL**

username, password : string

df\_user, df\_monster, df\_monster\_inventory : list

**function** cekusername (username : string) -> boolean

```

function cekusernameall (username : string, df_user : list)
-> boolean

```

**function** cekPassword (password : string) -> boolean

```

function register (df_user : list, df_monster : list,
df_monster_inventory : list) -> tuple(list)

```

```

function register (df_user : list, df_monster : list,
df_monster_inventory : list) -> list

```

{I.S List-list database dalam keadaan semula}

{F.S Menambahkan data baru kedalam list-list database berdasarkan aksi dalam modul}

{Perubahan :

df\_user -> username baru

df\_monster -> data diambil untuk monster awal pengguna

df\_monster\_inventory -> user id, monster id, dan level monster dari user baru}

{Sebuah menu untuk agent yang belum memiliki akun akan membuat akun terlebih dahulu dengan input username dan password agar bisa memulai perjalanan}

##### **ALGORITMA**

**function** cekusername(username : string) -> boolean

stat <- True

i traversal [0..(username]



```

        if not (i >= a and i <= z) or (i >= A and i <= Z) or
(i = '_' ) or (i = '-' ) or i not in ['0', '1', '2', '3', '4',
'5', '6', '7', '8', '9']) then
            stat <- False
    -> stat

```

```

function cekusernameall(username, df_user) -> bool
    stat <- True
    stat <- cekusername(username)
    i traversal [0...(df_user)]
        if i[0] = username then
            stat <- False
    -> stat

```

```

function cekPassword(password) -> boolean
    stat <- True
    if len(password) = 0 then
        stat <- False
    char traversal [0...(password)]
        if char = ';' then
            stat <- False
    -> stat

```

```

function register (df_user, df_monster, df_monster_inventory)
-> list
    belumRegister <- True
    while belumRegister do
        input(username)
    passBelumValid <- True
    while (passBelumValid) do
        input (password)

    monster traversal [0...(df_monster)]
        output(monster[0]). {monster[1]}

    repeat
        input(monsterPilihan)
        if monsterPilihan not in indeksPilihanMonster then
            ValueError
        else
            break
    -> idUserBaru, df_user, df_monster_inventory

```

d. F02-Login  
**KAMUS LOKAL**

```

df_user : array of array of string
username, password, notLogin, role : string
valid : boolean
i, indexBaris, len_user : integer
function login (df_user : list) -> list

{I.S List database dalam keadaan semula}
{F.S Mengambil data dalam list database}
{Perubahan :
    df_user -> akun user yang telah dipilih}
{Sebuah menu untuk agent yang sudah memiliki akun akan
diinstruksikan untuk memasukkan username dan password agar
dapat memulai/melanjutkan perjalanan menggunakan akun
tersebut}

```

#### **ALGORITMA**

```

function login(df_user : list) -> list
    input(username)
    input(password)
    notLogin <- ""
    while notLogin = "" do
        i <- 0
        valid <- False
        i traversal [0..(len(df_user))]
            if df_user[i][1] = username then
                valid <- True
                indexBaris <- i
                if df_user[i][3] = "admin" then
                    notLogin <- "admin"
                else
                    notLogin <- "agent"
                continue
    -> (indexBaris + 1), notLogin

```

#### e. F03-Logout

##### **KAMUS LOKAL**

```

statusLogin : boolean
function logout (statusLogin : boolean) -> boolean

{I.S Status yang menunjukkan kondisi awal akun pengguna}
{F.S Status akhir yang menunjukkan kondisi terkini akun
pengguna}
{Perubahan :
    statusLogin -> status akun terkini}

```

{Sebuah memo untuk agen yang ingin keluar dari akun yang sudah mereka gunakan dalam perjalanan}

#### **ALGORITMA**

```
function logout(statusLogin : string) -> string
    if statusLogin != "" then
        statusLogin <- ""
    -> statusLogin
```

#### f. F04 - Menu & Help

##### **KAMUS LOKAL**

status\_login : string

**function** help (status\_login : string) -> string

{I.S Memilih command yang akan dijalankan, status Agent harus sudah login}

{F.S Menjalankan command yang dipilih}

{Perubahan :

status\_login -> command}

{Sebuah command untuk mengingatkan pengguna untuk menuntun pengguna dan mengingatkan untuk melakukan validasi input dalam bentuk footnote}

#### **ALGORITMA**

```
function help(status_login : string) -> string
    if status_login = "admin" then
        helpMenu <- "
===== HELP =====
```

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

1. Logout: Keluar dari akun yang sedang digunakan
2. Shop Management: Melakukan manajemen pada SHOP sebagai tempat jual beli peralatan Agent
3. Monster Management: Melakukan manajemen pada data & statistik monster
4. Save: Melakukan aksi penyimpanan data untuk digunakan di kemudian hari
5. Exit Game

Footnote:

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

```

    else if status_login = "agent" then
        helpMenu <- "
===== HELP =====

```

Halo Agent! 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: Melakukan pertarungan dengan monster untuk mendapatkan OC
4. Arena: Melakukan set berbagai battle secara berturut-turut
5. Shop: Toko dimana berbagai monster dan potion bisa dibeli menggunakan OC
6. Laboratory: Melakukan upgrade/level up terhadap monster dengan OC
7. Save: Melakukan aksi penyimpanan data untuk digunakan di kemudian hari
8. Exit Game

Footnote:

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

```

    else
        helpMenu <- "
===== HELP =====

```

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

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

Footnote:

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

```

-> helpMenu

```

g. F05 - Monster

Penjelasan: Fitur F05-Monster hanya sebuah entitas, diimplementasikan di fungsi lain, dan tidak berdiri sendiri.

h. F06 - Potion

**KAMUS LOKAL**

monsterAgent, inventory\_user, id\_potion: list

HPAgent : integer

**function** potion (monsterAgent: list, inventory\_user: list,  
id\_potion: boolean, HPAgent: integer ) -> tuple(list)

{I.S Command untuk memilih potion}

{F.S Menggunakan potion yang telah dipilih}

{Sebuah menu untuk agent dapat memiliki dan menggunakan  
potion yang dapat membantu mereka dalam battle.

Jenis potion yang tersedia beserta efeknya.

1. Strength Potion: Meningkatkan ATK Power sebanyak 5% dari  
ATK Power.

2. Resilience Potion: Meningkatkan DEF Power sebanyak 5% dari  
DEF Power.

3. Healing Potion: Mengisi darah sebanyak 25% dari Base HP.  
HP tidak melebihi dari maksimal HP.

ATK Power, DEF Power, dan HP mengikuti level Monster saat  
ini.

Masing-masing potion hanya dapat digunakan sekali dalam 1  
battle dan efeknya hanya berlaku hingga battle itu selesai. }

**ALGORITMA**

**function** potion (monsterAgent, inventory\_user, id\_potion,  
HPAgent) -> tuple

monster <- monsterAgent[1] # Mengambil nama monster

b <- [0,0,0]

i <- 0

while (i < len(inventory\_user)) do

if (inventory\_user[i][1] = "strength") then

        b[0] <- int(inventory\_user[i][2])

else if (inventory\_user[i][1] = "resilience") then

        b[1] <- int(inventory\_user[i][2])

else

        b[2] <- int(inventory\_user[i][2])

    i <- i + 1

if b = [0,0,0] then

output("Tidak memiliki Potion")

```

        -> (monsterAgent, inventory_user, id_potion, HPAgent)
    else
        while True do
            output("""
===== POTION LIST =====
1. Strength Potion (Qty: {b[0]}) - Increases ATK Power
2. Resilience Potion (Qty: {b[1]}) - Increases DEF Power
3. Healing Potion (Qty: {b[2]}) - Restores Health
4. Cancel""")
            input(a)
            if a = "4" then
                -> (monsterAgent, inventory_user, id_potion,
HPAgent)
            else if a = "1" or a = "2" or a = "3" then
                if b[int(a)-1] = 0 then # Selama tidak ada
potion, harus mengulang
                    else
                        break
                else
                    output("Input tidak valid! Ulangi.")

            if (int(a) - 1 = 0) then
                if (id_potion[0] = True) then
                    i <- 0
                    while (i < len(inventory_user)) do
                        if inventory_user[i][1] = "strength" then
                            inventory_user[i][2] <- str(b[0] - 1
                            i <- i + 1
                            monsterAgent[2] <- ((monsterAgent[2])*21/20)
                            id_potion[0] <- False
                        else
                            output("{monster} menolak ramuan ini")
                    else if (int(a) - 1 = 1) then
                        if (id_potion[1] = True) then
                            i <- 0
                            while (i < len(inventory_user)) do
                                if inventory_user[i][1] = "resilience"
then
                                    inventory_user[i][2] <- str(b[1] - 1)
                                    i <- i + 1
                                    monsterAgent[3] <- ((monsterAgent[3])*21/20)
                                    id_potion[1] <- False
                                else if (int(a) - 1 = 2) then
                                    if (id_potion[2] = True) then
                                        i <- 0

```

```

        while (i < len(inventory_user)) do
            if (inventory_user[i][1] = "healing")
then
                inventory_user[i][2] <- str(b[2] - 1)
                i <- i + 1
                HPAgent <- ((HPAgent)*5/4)
                if HPAgent > monsterAgent[4] then
                    HPAgent <- monsterAgent[4]
                id_potion[2] <- False
            -> (monsterAgent, inventory_user, id_potion, HPAgent

```

#### i. F07-Inventory

##### **KAMUS LOKAL**

```

monster_inventory_df,      item_inventory_df,      monster_df,
userAccount : array [1..Neff] of string
owcacoins : string
procedure      inventory      (monster_inventory_df:      list,
item_inventory_df: list, monster_df: list, owcacoins: string,
userAccount: list)

```

```

{I.S ID untuk command tampilan detail}
{F.S Menampilkan detail setiap item di inventory}
{Sebuah menu untuk memunculkan detail setiap item yang
disimpan pada inventory. Untuk potion, akan ditampilkan
kegunaan (type) dan kuantitasnya. Untuk monster, akan
ditampilkan atribut-atribut yang dimiliki oleh monster
tersebut (type/name, atk power, def power, hp, level). Menu
ini juga menampilkan O.W.C.A. coin yang dimiliki user saat
ini.}

```

##### **ALGORITMA**

```

procedure      inventory      (monster_inventory_df:      list,
item_inventory_df: list, monster_df: list, owcacoins: string,
userAccount: list)
    i <- 0
    while (i < len(monster_user)) do
        output      (f"{i+1}.      Monster      (Name:
{monster_df[int(monster_user[i][1])-1][1]},      Lvl:
{monster_user[i][2]},      HP:
{monster_df[int(monster_user[i][1])-1][4]})")
        i <- i + 1
    j <- 0
    while (j < len(inventory_user)) do

```

```

        output(f"{len(monster_user)+ j+1}. Potion (Type:
{functions.convertItemNames((inventory_user[j][1]),'name')},
Amount: {inventory_user[j][2]})")
        j <- j + 1

```

#### j. F08 - Battle

##### **KAMUS LOKAL**

```

monster_user, monster_db, inventory_user, monsterAgent: array
[1..Neff] of string
levelMusuh : integer
function hitungDamage(atkPower: integer, defPower:
integer)-> tuple (integer, integer)
function hitungAtribut(baseAtk: integer, baseDef: integer,
baseHP: integer, level: integer)-> tuple(integer,integer,
integer)
function konversi_hp_ke_integer(monster_db: list)-> integer
function seleksimonster (monster_user: list,monster_db:
list)-> array
function startBattle(monster_db: list, inventory_user: list,
monsterAgent: list, levelMusuh: integer)-> int, int, list,
int
function battle (owca_coin: string, monster_user: list,
monster_db: list, inventory_user: list)-> int, list

```

{I.S List-list database dalam keadaan semula }

{F.S List-list database berubah sesuai fungsi yang jalan pada modul}

{Perubahan:

user\_db -> }

{Sebuah menu yang membuat monster pilihan agent akan bertarung dengan monster musuh yang muncul secara acak}

##### **ALGORITMA**

```

function hitungDamage(atkPower: integer, defPower:
integer)-> tuple (integer, integer)
atkPower <- strToInt(atkPower)
defPower <- strToInt(defPower)
minAtk <- atkPower - (atkPower * 0.3)
maxAtk <- atkPower + (atkPower * 0.3)
seed <- randomNumberGenerator_formula(a=4871, c=0, m=2^31-1,
seed=None)
attack <- randomNumberGenerator_randomNumberGenerator(seed,
numRange=[minAtk, maxAtk])
damage <- round(attack * ((100 - defPower) / 100))
-> damage

```



```

function hitungAtribut(baseAtk: integer, baseDef: integer,
baseHP: integer, level: integer)-> tuple(integer,integer,
integer)
baseAtk <- strToInt(baseAtk)
baseDef <- strToInt(baseDef)
multipilier <- 1 + ((strToInt(level) - 1) * 0.1)
-> (baseAtk * multipilier, baseDef * multipilier, baseHP *
multipilier)

```

```

function konversi_hp_ke_integer(monster_db: list)-> integer
repeat (length(monster_db)) times
    monster_db[i][5] <- strToInt(monster_db[i][5])
-> monster_db

```

```

function seleksiMonster(monster_user: list, monster_db:
list)-> tuple
output("<=== CHOOSE YOUR MONSTER ===>")
num <- 0
for each row in monster_user do
    for each mow in monster_db do
        if row[2] = mow[1] then
            output(num+1, ". ", mow[2], " ", repeat(" ",
14-length(mow[2])), "| Level : ", row[3])
            num <- num + 1
output()
repeat
    chooseMonster <- input(functions_clr("Pilih monster anda : ",
'cyan', ''))
    if not functions_isInteger(chooseMonster) then
        output('Masukan berupa integer!\n')
        continue
    jumlahMonster <- functions_hitungJumlahBaris(monster_user)
    if 1 <= strToInt(chooseMonster) <= jumlahMonster then
        ids <- monster_user[strToInt(chooseMonster)][2]
        monsterAgent <- monster_db[strToInt(ids)]
        append(monsterAgent,
monster_user[strToInt(chooseMonster)][3])
        exit repeat
    else
        output("Pilihan monster tidak tersedia! Silahkan pilih
kembali!\n")
until false
-> monsterAgent

```

```

function startBattle(monster_db: list, inventory_user: list,
monsterAgent: list, levelMusuh: integer)-> int, int, list,
int
seed <- randomNumberGenerator_formula(4871, 0, 2^31-1, None)
jumlahMonster <- len(monster_db)
                                indeksMusuh                                <-
randomNumberGenerator_randomNumberGenerator(seed,                                [0,
jumlahMonster])
monsterMusuh <- None
i <- 0
for monster in monster_db do
    if i = indeksMusuh then
        monsterMusuh <- monster
    i <- i + 1
output("OMAGAAA monster ", monsterMusuh[1], " muncul !!!")
mulaiBattle <- true
HPAgent <- int(monsterAgent[4])
HPMusuh <- int(monsterMusuh[4])
hpm <- HPAgent
levelAgent <- int(monsterAgent[5])
turn <- 1
id_potion <- [true, true, true]
totalDamageDiberikan <- 0
totalDamageDiterima <- 0
while mulaiBattle do
    output("<===Turn ", turn, " (", monsterAgent[1],
")====>")
    output("1. Attack")
    output("2. Potion")
    output("3. Quit")
    action <- input("Pilih perintah : ")
    if not isInteger(action) then
        output("Input tidak valid, silahkan coba lagi")
        continue
    if action = "2" then
        monsterAgent, inventory_user, id_potion, HPAgent <-
potion(monsterAgent, inventory_user, id_potion, HPAgent)
    elseif action = "3" then
        output("Anda berhasil kabur dari pertarungan")
    elseif action = "1" then
        atkPower, defPower, _ <- hitungAtribut(monsterAgent[2],
monsterMusuh[3], HPAgent, levelAgent)
        damage <- hitungDamage(atkPower, defPower)
        HPMusuh <- HPMusuh - damage
        totalDamageDiberikan <- totalDamageDiberikan + damage

```

```

        output("BOOM ", monsterAgent[1], " menyerang ",
monsterMusuh[1], " dengan damage ", damage)
        if HPMusuh <= 0 then
            output("VERY NIICE ", monsterMusuh[1], " berhasil
dikalahkan !!!")
            mulaiBattle <- false
        else
            output("Sisa HP musuh ", monsterMusuh[1], " sebesar
", HPMusuh)
            output("RAWRRR Giliran ", monsterMusuh[1], "
menyerang!!!")
            atkMusuh, _, _ <- hitungAtribut(monsterMusuh[2],
monsterMusuh[3], HPMusuh, levelMusuh)
            damageMusuh <- hitungDamage(atkMusuh, monsterAgent[3])
            HPAgent <- HPAgent - damageMusuh
            totalDamageDiterima <- totalDamageDiterima +
damageMusuh
            output(monsterMusuh[1], " menyerang ", monsterAgent[1],
" dengan damage ", damageMusuh)
            if HPAgent <= 0 then
                mulaiBattle <- false
            else
                output("Sisa HP dari ", monsterAgent[1], " sebesar ",
HPAgent)
                turn <- turn + 1
            if HPMusuh <= 0 then
                win_stat <- "menang"
            elseif HPAgent <= 0 then
                win_stat <- "kalah"
            else
                win_stat <- "kabur"
            endif
            -> totalDamageDiberikan, totalDamageDiterima,
inventory_user, win_stat

```

```

function battle (owca_coin: string, monster_user: list,
monster_db: list, inventory_user: list)-> int, list
seed <- randomNumberGenerator_formula(4871, 0, 2^31-1, None)
                                levelMusuh <-
randomNumberGenerator_randomNumberGenerator(seed, [1, 5])
    monsterAgent <- seleksimonster(monster_user, monster_db)
    _, _, inventory_user, win_stat <- startBattle(monster_db,
inventory_user, monsterAgent, levelMusuh)
    if win_stat = "menang" then

```

```

                                coinDidapat      <-
randomNumberGenerator_randomNumberGenerator(seed, [20, 100])
    owca_coin <- str(int(owca_coin) + coinDidapat)
    output("Anda mendapatakan ", coinDidapat, " OC")
    elseif win_stat = "kalah" then
        output("hiks hiks ", monsterAgent[1], " telah dikalahkan,
jangan putus asa tetap semangat!")
    else
        output("Exiting battle...")
-> owca_coin, inventory_user

```

#### k. F09-Arena

##### **KAMUS LOKAL**

```

monster_user, inventory_user, monster_db: array [1..Neff] of
string
owcacoin : string
function arena (owca_coin: string, monster_user: list,
monster_db: list, inventory_user: list)-> tuple (string,
list)

```

{I.S List-list database dalam keadaan semula }

{F.S List-list database berubah sesuai fungsi yang jalan pada modul}

{Perubahan:

owca\_coin -> penambahan jumlah owca\_coin tergantung kemenangan atau tidaknya agent

Inventory\_user -> perubahan inventory berdasarkan menang atau tidaknya agent }

{Sebuah menu yang akan menjalankan fungsi battle yang akan diulang selama 5 kali sesuai dengan urutan stage dari 1 sampai 5}

##### **ALGORITMA**

```
totalDamageDiterima <- 0
```

```
totalDamageDiberikan <- 0
```

```
hadiah <- [30, 50, 100, 150, 200]
```

```
totalhadiah <- [0, 30, 80, 180, 330, 530]
```

```
output("Selamat datang di arena!")
```

```
monsterAgent      <-      battle.seleksimonster(monster_user,
monster_db)
```

```
stage <- 1
```

```
while stage < 6 do
```

```
    levelMusuh <- stage
```

```

    output("==== Stage " + stage + " ====")
    (damageDiberikan, damageDiterima, inventory_user, stat)
<-      battle.startBattle(monster_db,      inventory_user,
monsterAgent, levelMusuh)
        totalDamageDiterima <- totalDamageDiterima +
damageDiterima
        totalDamageDiberikan <- totalDamageDiberikan +
damageDiberikan

    if stat = "kalah" then
        output("Womp womp agent kalah dalam arena")
    else if stat = "kabur" then
        output("Tidak seru! Rupanya agent kabur dari arena!!!
:-b")

    else
        if stage < 5 then
            output("STAGE CLEARED! Anda mendapatkan " +
hadiah[stage - 1] + " OC pada stage ini!\n")
            owca_coin <- str_to_int(owca_coin) + hadiah[stage
- 1]
            stage <- stage + 1
            input("Tekan enter untuk melanjutkan ke stage
selanjutnya!")
        else
            output("Selamat, Agent berhasil menyelesaikan
seluruh stage Arena !!!")
            owca_coin <- str_to_int(owca_coin) + hadiah[stage
- 1]
            stage <- stage + 1

stats <- ""
==== STATS ====
Total hadiah      : " + totalhadiah[stage-1] + " OC
Jumlah Stage      : " + (stage-1) + "
Damage diberikan  : " + totalDamageDiberikan + "
Damage diterima   : " + totalDamageDiterima + "
""
output(stats)

->(owca_coin, inventory_user)

```

# I. F10 - Shop & Currency (Agent)

## **KAMUS LOKAL**

```

monsterDalamShop,itemDalamShop,monsterShopList = list[list]

aksiUtama, cekLihat, yesNo : array [1..Neff] of string
shopMenu : boolean
function shop_lihat_beli ( cekLihat : list, aksi_shop :
string, owca : integer) -> string

function f_lihat (tipeList : string, combMonsterList : list,
itemList : list) -> boolean

function beliShop (monsterDalamShop: list, itemDalamShop :
list, monsterShopList : list, monsterShop_db:
list,itemShop_db : list, monsterInventory_db : list,
itemInventory_db : list, OC : integer, tipeList : string,
idUser : string) -> tuple(list(list))

function shop (user_db : list, monster_db : list,
itemInventory_db : list, itemShop_db : list,
monsterInventory_db : list, monsterShop_db : list, idUser :
string, owca : integer) -> tuple(list)

{I.S List-list database dalam keadaan semula}
{F.S List-list database berubah berdasarkan aksi dalam modul}
{Perubahan :
    user_db -> O.W.C.A Coin
    itemInventory_db -> quantity item
    monsterInventory_db -> menambah monster
    itemShop_db -> Stok item
    monsterShop_db -> Stok monster}
{Sebuah menu shop saat Agent dapat melihat dan membeli stok
monster atau potion yang ada pada shop}

```

#### **ALGORITMA**

```

function shop_lihat_beli(cekLihat, aksi_shop, owca):
    if aksi_shop = "beli" then
        output(current OC)
    repeat
        output(action options)
        aksi_lihat <- input lowercased action

        if aksi_lihat in cekLihat then
            break
        else
            output(invalid option message)
    -> aksi_lihat

```

```

function f_lihat(tipeList, combMonsterList, itemList):
    if (tipeList = "monster") then
        output(monster list with header)
    else if (tipeList = "item") then
        output(item list with header)
    -> true

function beliShop (monsterDalamShop, itemDalamShop,
monsterShopList, monsterShop_db, itemShop_db,
monsterInventory_db, itemInventory_db, OC, tipeList, idUser):
    if (tipeList = "monster") then
        repeat
            idBeli <- input monster ID
            until idBeli is valid integer and exists in
monsterDalamShop

            if (monster is in user inventory) then
                output(already in inventory message)
            else
                hargaMonster <- get price from monsterShop_db
                stokMonster <- get stock from monsterShop_db
                if (OC >= hargaMonster) then
                    yakin <- input confirmation
                    if (yakin = "y") then
                        add monster to user inventory
                        decrement monster stock in shop
                        decrement OC by hargaMonster
                        monsterShop_db <- update monsterShop_db
with new stock
                        monsterShopList <- update monsterShopList
with new stock
                    output(successful purchase message)
                else
                    output(insufficient OC message)
        else if (tipeList = "item") then
            repeat
                idBeli <- input item ID
                until idBeli is valid integer and exists in
itemDalamShop
            repeat
                jumlah <- input item quantity
                until quantity is valid and stock and OC are
sufficient
            shopMenu <- true

```

```
-> monsterShopList, shopMenu, OC, monsterShop_db,
itemShop_db, monsterInventory_db, itemInventory_db
```

```
function shop (user_db, monster_db, itemInventory_db,
itemShop_db, monsterInventory_db, monsterShop_db, idUser,
owca)
aksiUtama <- ['lihat','beli','keluar','data']
cekLihat <- ['monster','item','batal']
owca <- (owca)
shopMenu <- True
```

```
  while shopMenu do
    shopMenu <- False
    update shop lists
    repeat
      aksi_shop <- input main action
    until aksi_shop in aksiUtama

    if (aksi_shop = "data") then
      mana <- input (data type)
      if (mana = "inventory") then
        output(user inventory)
      else if (mana = "shop") then
        output(shop data)
      else if (mana = "user") then
        output(user data)
      -> true
    else if (aksi_shop != "keluar") then
      aksi_shop_2 <- shop_lihat_beli (cekLihat,
      aksi_shop, owca)
      if (aksi_shop = "lihat") then
        -> f_lihat(aksi_shop_2, monsterDalamShop,
itemDalamShop)
      else if (aksi_shop = "beli") then
        -> beliShop(monsterDalamShop, itemDalamShop,
monsterShopList, monsterShop_db, itemShop_db,
monsterInventory_db, itemInventory_db, owca, aksi_shop_2,
idUser)
        -> monster_db, itemInventory_db, itemShop_db,
monsterInventory_db, monsterShop_db, owca
```

m. FII - Monster Laboratory

**KAMUS LOKAL**



```

menuLab, lanjut : boolean
jumlahMonster : integer
Inventory : list[list]

```

```

function panggilBerdasarkanID (namaList : list, indexID :
string) -> integer

```

```

function laboratory (monster_db : list, monsterInventory_db :
list, user_db : list, idUser : string, owca : integer) ->
tuple(list)

```

```

{I.S List-list database dalam keadaan semula}
{F.S List-list database berubah berdasarkan aksi dalam modul}
{Perubahan :
    monsterInventory_db -> level monster
    user_db -> O.W.C.A Coin}
{Sebuah menu saat para Agent dapat menaikkan level
monster-monster yang dimiliki Agent}

```

#### ALGORITMA

```

function panggilBerdasarkanID (namaList, indexID)

```

```

    idx <- 0
    line traversal [0...(namaList)]
    if line[0] = indexID then
        idx <- idx + 1
    -> idx

```

```

function laboratory (monster_db, monsterInventory_db,
user_db, idUser, owca)

```

```

    menuLab <- true
    lanjut <- true
    jumlahMonster <- fn.hitungJumlahBaris(inventory)
    while menuLab do
        menuLab <- false
        if jumlahMonster > 0 then
            i traversal [0...(jumlahMonster)]
                idx <- panggilBerdasarkanID(monster_db,
inventory[i][1])
                namaMonster <- monster_db[idx][1]
                levelMonster <- inventory[i][2]
            lanjut <- true
        else
            lanjut <- false

    if (lanjut) then
        repeat

```

```

(aksi)
if (aksi = 'upgrade') then
  repeat
    (nomorMonster)
    if nomorMonster = 'keluar' then
      menuLab <- true

      levelMonster <-
inventory[(nomorMonster)-1][2]
      if (levelMonster = '1') then
        harga <- 300
      else if (levelMonster = '2') then
        harga <- 600
      else if (levelMonster = '3') then
        harga <- 1000
      else if (levelMonster = '4') then
        harga <- 1500

      if not menuLab:
        idMonsterAsli <-
inventory[(nomorMonster)-1][1]
        panggilBerdasarkanID (monster_db,
idMonsterAsli)
        namaMonster <- monster_db[idx-1][1]

        repeat
          (yesNo)
          if (yesNo = 'y') then
            output(congrats message)
            owca <- str(int(owca) -
harga)

inventory[(nomorMonster)-1][2] <- str(int(levelMonster) + 1)
            output(owca)
            menuLab <- true
          else if (yesNo = 'n') then
            menuLab <- true

      else if (aksi = 'keluar') then
        output(goodbye message)

-> monsterInventory_db, owca

```

n. F12 - Shop (Admin)

**KAMUS LOKAL**

admin\_main\_aksi, lihat\_valid : list

shopMenu : boolean

**function** database (monster\_db : list, monsterDalamShop : list, itemDalamShop : list, tipeList : string) -> tuple(list)

**function** menuPertama(lihat\_valid: list, aksi\_shopmng: string) -> string

**function** fnTambah(monsterDalamShop : list, itemDalamShop : list, shopMenu : bool, tipeList : str)

**function** fnUbah(shopMenu : bool, tipeList : str) -> bool:

**function** fnHapus (monsterDalamShop : list, itemDalamShop : list, monsterShop\_db : list, itemShop\_db : list, tipeList: str) -> bool

**function** shopManagement(monster\_db : list, monsterShop\_db : list, itemShop\_db : list) -> tuple (list)

{I.S List-list database dalam keadaan semula}

{F.S List-list database berubah berdasarkan aksi dalam modul}

{Perubahan:

monsterShop\_db -> Stok dan harga monster}

itemShop\_db -> Stok dan harga item

{Sebuah menu dimana para Agent dapat menaikkan level monster-monster yang dimiliki Agent}

**ALGORITMA**

**function** database (monster\_db : list, monsterDalamShop : list, itemDalamShop : list, tipeList : string) -> tuple(list)

indexDalamShop, indexDatabase <- database(monster\_db, monsterDalamShop, itemDalamShop, itemShop\_db, tipeList)

if (len(indexDatabase) > 0) then

output('> Ketik "BATAL" untuk keluar')

repeat

id <- input('>>> Masukkan id ' + tipeList + ': ')

if (id = 'batal') then

shopMenu <- true

exit repeat

if not (fn\_isInteger(id)) then

output('Masukan berupa integer!')

else

if (tipeList = 'monster') then

```

        tipe <- 'Monster'
    else if (tipeList = 'item') then
        tipe <- 'Item'
        id <- fn_convertItemNames(id, 'id')
        if id not in indexDatabase then
            if id in indexDalamShop then
                output(tipe + ' sudah ada di shop!')
            else
                output(tipe + ' tidak ada di database!')
        else
            exit repeat
until false
repeat
    if shopMenu then
        exit repeat
    stok <- input('>>> Masukkan stok awal: ')
    if fn_isInteger(stok) then
        if int(stok) <= 0 then
            output('Stok harus lebih dari 0')
        else
            exit repeat
    else
        output('Masukan berupa integer!')
until false

repeat
    if shopMenu then
        exit repeat
    harga <- input('>>> Masukkan harga: ')
    if (fn_isInteger(harga)) then
        if (int(harga) < 0) then
            output('Harga tidak boleh negatif!')
        else
            exit repeat
    endif
    else
        output('Masukan berupa integer!')
until false
if (tipeList = 'monster') then
    listDatabase <- monster_db
    listUnitShop <- monsterShop_db
    indexKolom <- 1
else if tipeList = 'item' then
    listDatabase <- itemShop_db
    listUnitShop <- itemShop_db

```

```

        indexKolom <- 0
        if not shopMenu then
            for line in listDatabase do
                if line[0] = id then
                    namaUnit <- line[indexKolom]
            for line in listUnitShop do
                if line[0] = id then
                    line[1] <- stok
                    line[2] <- harga
            namaUnit <- fn_convertItemNames(namaUnit, 'name')
            output(tipe + ' telah berhasil ditambahkan!')
            output('Nama : ' + namaUnit)
            output('Stok : ' + stok)
            output('Harga : ' + harga)
        shopMenu <- true
    -> shopMenu, monsterShop_db, itemShop_db

```

```

function menuPertama(lihat_valid: list, aksi_shopmng: string)
-> string

```

```

while (True) do
    output('=> AKSI <=&n> Monster&n> Item&n> Batal')
    output('-'*55)
    monster_potion <- str.lower(input(fn.clr('>>> Mau ' +
aksi_shopmng + ' apa?: ', 'cyan', '')))
    {Validasi Input}
    if (monster_potion in lihat_valid) then
        exit
    else
        output('Pilih opsi yang valid!&n')
    -> monster_potion

```

```

function fnTambah(monsterDalamShop : list, itemDalamShop :
list, shopMenu : bool, tipeList : str)

```

```

(indexDalamShop, indexDatabase) <- database(monster_db,
monsterDalamShop, itemDalamShop, itemShop_db, tipeList)

```

```

if (len(indexDatabase) > 0) then
    output('&n> Ketik "BATAL" untuk keluar&n')
    repeat
        id <- str.lower(input(fn.clr('>>> Masukkan id ' +
tipeList + ': ', 'cyan', '')))

```

```

if (id = 'batal') then
  shopMenu <- true
  exit

if (not fn.isInteger(id)) then
  output('Masukan berupa integer!')
  continue

if (tipeList = 'monster') then
  tipe <- 'Monster'
else if (tipeList = 'item') then
  tipe <- 'Item'
  id <- fn.convertItemNames(id, 'id')

if (not (id in indexDatabase)) then
  if (id in indexDalamShop) then
    output(tipe + ' sudah ada di shop!\n')
  else
    output(tipe + ' tidak ada di database!\n')
  continue
else
  exit
until (false)

repeat
  if (shopMenu) then
    exit
  stok <- str.lower(input(fn.clr('>>> Masukkan stok awal:
', 'cyan', ')))
  if (fn.isInteger(stok)) then
    if (int(stok) <= 0) then
      output('Stok harus lebih dari 0\n')
      continue
    else
      exit
  else
    output('Masukan berupa integer!\n')
    continue
until (false)

repeat

```

```

    if (shopMenu) then
      exit
      harga <- str.lower(input(fn.clr('>>> Masukkan harga: ',
'cyan', '')))
      if (fn.isInteger(harga)) then
        if (int(harga) < 0) then
          output('Harga tidak boleh negatif!\n')
          continue
        else
          exit
      else
        output('Masukan berupa integer!\n')
        continue
until (false)

```

```

if (tipeList = 'monster') then

  listDatabase <- monster_db

  listUnitShop <- monsterShop_db

  indexKolom <- 1 {indexKolom Nama}
else if (tipeList = 'item') then

  listDatabase <- itemShop_db

  listUnitShop <- itemShop_db

  indexKolom <- 0

```

```

if (not shopMenu) then

  foreach line in listDatabase do

    if (line[0] = id) then

      namaUnit <- line[indexKolom]

  foreach line in listUnitShop do

    if (line[0] = id) then

      line[1] <- stok

```

```

        line[2] <- harga

        namaUnit <- fn.convertItemNames(namaUnit, 'name')
        output('\n' + tipe + ' telah berhasil ditambahkan!')
        output('Nama : ' + namaUnit)
        output('Stok : ' + stok)
        output('Harga : ' + harga)
        {manipulasi list of monster shop}

shopMenu <- true {Kembali ke shop menu}
-> (shopMenu, monsterShop_db, itemShop_db)

function fnUbah(shopMenu : bool, tipeList : str) -> bool:
if (tipeList = 'monster') then
    fn.printData(fn.appendHeader(monsterDalamShop, tipeList))
    listUnitShop <- monsterDalamShop
    indexList <- fn.indexDalamList(monsterDalamShop)
elseif (tipeList = 'item') then
    fn.printData(fn.appendHeader(itemDalamShop, tipeList))
    listDatabase <- itemDalamShop
    listUnitShop <- itemDalamShop
    indexList <- ['1', '2', '3']

output('\n')
repeat
    id <- str.lower(input(fn.clr('>>> Masukkan id ' + tipeList
+ ': ', 'cyan', '')))
    if (id in indexList) then

```



```

        exit
    else
        if (fn.isInteger(id)) then
            if (tipeList = 'monster') then
                if (int(id) > fn.hitungJumlahBaris(monster_db)) then
                    output('Monster tidak ada dalam database!\n')
                else
                    output('Monster tidak ada dalam shop\n')
            else if (tipeList = 'item') then
                if (int(id) > 3) then
                    output('Item tidak ada dalam database!\n')
                else
                    output('Item tidak ada dalam shop\n')
            else
                output('Masukan berupa integer!\n')
        until (false)

    repeat
        stok <- str.lower(input(fn.clr('>>> Masukkan stok baru: ',
        'cyan', '')))
        if (not fn.isInteger(stok)) then
            output('Masukan berupa integer!\n')
        else
            exit
    until (false)

    repeat
        harga <- str.lower(input(fn.clr('>>> Masukkan harga baru:
        ', 'cyan', '')))

```

```

    if (not fn.isInteger(harga)) then
        output('Masukan berupa integer!\n')
    else
        exit
until (false)

```

```

if (tipeList = 'monster') then
    tipe <- 'Monster'
    listDatabase <- monster_db
    listUnitShop <- monsterShop_db
    indexKolom <- 1 {indexKolom Nama}
elseif (tipeList = 'item') then
    tipe <- 'Item'
    listDatabase <- itemShop_db
    listUnitShop <- itemShop_db
    indexKolom <- 0
    id <- fn.convertItemNames(id, 'id')
endif

```

```

if (not shopMenu) then
    foreach line in listDatabase do
        if (line[0] = id) then
            namaUnit <- line[indexKolom]

    for each line in listUnitShop do
        if (line[0] = id) then
            line[1] <- stok
            line[2] <- harga

```

```

namaUnit <- fn.convertItemNames(namaUnit, 'name')

output('\n')

if (stok == '' and harga == '') then
  output(tipe + ' tidak jadi diubah')
else
  output(tipe + ' telah berhasil diubah!')
  output('Nama : ' + namaUnit)
  if (stok != '') then
    output('Stok baru : ' + stok)
  if (harga != '') then
    output('Harga baru: ' + harga)
{manipulasi list of monster shop}

shopMenu <- true {Kembali ke shop menu}
-> (shopMenu, monsterShop_db, itemShop_db)

function fnHapus (monsterDalamShop : list, itemDalamShop :
list, monsterShop_db : list, itemShop_db : list, tipeList:
str) -> bool

if (tipeList == 'monster') then

  tipe <- 'Monster'

  fn.printData(fn.appendHeader(monsterDalamShop, tipeList))

  listDatabase <- monster_db

  listUnitShop <- monsterShop_db

  indexKolom <- 1 {indexKolom Nama untuk pemanggilan}
elseif (tipeList == 'item') then

  tipe <- 'Item'

  fn.printData(fn.appendHeader(itemDalamShop, tipeList))

```

```

listDatabase <- itemShop_db

listUnitShop <- itemShop_db

indexKolom <- 0

id <- str.lower(input(fn.clr('>>> Masukkan id ' + tipeList +
': ', 'cyan', '')))

if (tipeList = 'item') then

  id <- fn.convertItemNames(id, 'id')

{Manipulasi list}

for each line in listDatabase do

  if (line[0] = id) then

    namaUnit <- line[indexKolom]

for each line in listUnitShop do

  if (line[0] = id) then

    line[1] <- 0

namaUnit <- fn.convertItemNames(namaUnit, 'name')

output(tipe + ' ' + namaUnit + ' telah berhasil dihapus dari
shop!')

shopMenu <- true {Kembali ke shop menu}

-> (shopMenu, monsterShop_db, itemShop_db)

function shopManagement(monster_db : list, monsterShop_db :
list, itemShop_db : list) -> tuple (list)

admin_main_aksi <- ['lihat','tambah','ubah','hapus','keluar']

```

```

lihat_valid <- ['monster','item','batal']

shopMenu <- benar

{Menampilkan header}

output('='*22 + fn.clr(' SHOP MANAGEMENT ', 'yellow', 'bold') +
'='*22, '\n')

output('      \(\^▽^\)/ Admin, Welcome to the
SHOP!!!', '\n')

{Loop utama menu shop}

while shopMenu do

    shopMenu <- salah

    itemDalamShop <- fn.updateListShop(itemShop_db, 'item')

    monsterShopList <-
fn.combineList(monster_db, monsterShop_db)

    monsterDalamShop <-
fn.updateListShop(monsterShopList, 'monster')

{Menu shop}

while benar do

    output()

    output('-'*55)

    output('=> AKSI <=\n'

        '> Lihat\n'

        '> Tambah\n'

        '> Ubah\n'

        '> Hapus\n'

        '> Keluar\n')

```

```

        output('-'*55)

        aksi_shopmng <- input(fn.clr('>>> Pilih aksi:
', 'cyan', ''))

        if aksi_shopmng ada dalam admin_main_aksi then

            keluar loop

        else

            output('Pilih aksi yang valid!\n')

        {Proses berdasarkan aksi yang dipilih}

        if aksi_shopmng ≠ 'keluar' then

            monster_potion <- menuPertama(lihat_valid,
aksi_shopmng)

            if monster_potion = 'batal' then

                shopMenu <- benar

                lanjutkan loop

            endif

            if aksi_shopmng = 'lihat' then

                output('-'*55)

                shopMenu <-
shop.f_lihat(monster_potion, monsterDalamShop, itemDalamShop)

            elseif aksi_shopmng = 'tambah' then

                output('-'*55)

                (shopMenu, monsterShop_db, itemShop_db) <-
fnTambah(monster_db, monsterDalamShop,
monsterShop_db, itemDalamShop, itemShop_db, shopMenu,
monster_potion)

            elseif aksi_shopmng = 'ubah' then

                output('-'*55)

```

```

        (shopMenu,monsterShop_db,itemShop_db) <-
fnUbah(monsterShop_db,itemShop_db,monster_db,
monsterDalamShop, itemDalamShop,shopMenu,monster_potion)

        elseif aksi_shopmng = 'hapus' then

            output('-'*55)

            (shopMenu, monsterShop_db,itemShop_db) <-
fnHapus(monster_db, monsterDalamShop,
itemDalamShop,monsterShop_db,itemShop_db,monster_potion)

        else

            {Keluar dari menu shop}

            output()

            output('  ✖(•ω•`)o Selamat Tinggal!!! ')

            output()

        {Mengembalikan database yang mungkin telah diubah}
-> monster_db, monsterShop_db, itemShop_db

```

#### o. F13 - Monster Management

**function** inputDanCekMons(monster\_db : list, monsterShop\_db : list) -> tuple(list)

**function** monsterManagement (monster\_db : list, monsterShop\_db : list) -> tuple(list)

{I.S List-list database dalam keadaan semula}

{F.S List-list database berubah berdasarkan aksi dalam modul}

{Perubahan:

monster\_db -> Nama dan atribut monster}

monsterShop\_db -> Monster baru dengan stok 0}

{Sebuah menu dimana para Agent dapat menaikkan level monster-monster yang dimiliki Agent}

#### **ALGORITMA**

**function** inputDanCekMons(monster\_db : list, monsterShop\_db : list) -> tuple(list)

brk <- salah

```

{Loop untuk memasukkan nama monster}

while benar do

    name <- input('>>> Masukkan Type / Nama Monster: ')

    found <- fn.cekUdahAda(name, monster_db, 1)

    if found then

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

    else

        keluar loop


{Loop untuk memasukkan ATK Power}

while benar do

    atkPower <- input(fn.clr('>>> Masukkan ATK Power:
','cyan',''))

    if fn.isInteger(atkPower) then

        if int(atkPower) ≥ 0 then

            keluar loop

        else

            output('ATK Power harus bernilai positif, coba
lagi!','\n')

        else

            output('Masukkan input bertipe Integer, Coba
lagi!','\n')


{Loop untuk memasukkan DEF Power}

while benar do

    defense <- input(fn.clr('>>> Masukkan DEF Power:
','cyan',''))

    if fn.isInteger(defense) then

```



```

        if int(defense) ≤ 50 dan int(defense) ≥ 0 then
            keluar loop
        else
            output('DEF Power harus bernilai 0-50, coba
lagi!','\n')
        endif
    else
        output('Masukkan input bertipe Integer, Coba
lagi!','\n')

{Loop untuk memasukkan HP}
while benar do
    hp <- input(fn.clr('>>> Masukkan HP: ','cyan',''))
    if fn.isInteger(hp) then
        if int(hp) > 0 then
            keluar loop
        else
            output('HP harus lebih besar dari 0, coba
lagi!','\n')
        endif
    else
        output('Masukkan harus Integer, Coba lagi!','\n')

{Mencari ID terbesar di list dan menambahkan monster baru}
for line dalam monster_db do
    idLast <- line[0]
monster_db.append([str(int(idLast)+1), name, atkPower,
defense, hp])

```

```

monsterShop_db.append([str(int(idLast)+1), 0, 0])

monsMenu <- benar

-> monsMenu, monster_db, monsterShop_db

function monsterManagement (monster_db : list, monsterShop_db
: list) -> tuple(list)

monsMenu <- benar

tipe <- 'monster'

{Menampilkan header}

output('='*26 + ' OWCA-DEX ' + '='*26,'\n')

output('      ♪ •` ω •´ ♪ SELAMAT DATANG DI OWCA-DEX!!!')

{Loop utama menu monster}

while monsMenu do

    monsMenu <- salah

    output('-'*55)

    output('Menu')

    output('> Lihat')

    output('> Tambah')

    output('> Keluar')

    {Loop untuk memilih aksi}

    while benar do

        output('-'*55)

```

```

        mon_aksi <- input(fn.clr('>>> Pilih Aksi:
', 'cyan', ''))

        if mon_aksi = '1' atau mon_aksi = '2' atau mon_aksi =
'3' then

            keluar loop

        else

            output('Pilih aksi yang valid!!')

{Proses berdasarkan aksi yang dipilih}
if mon_aksi = '1' then
    output('-'*55)

    fn.printData(fn.appendHeader(monster_db, 'monster'))

    output()

    monsMenu <- benar
else if mon_aksi = '2' then
    output('-'*55)

    output('Memulai pembuatan Monster baru')

    (monsMenu, monster_db, monsterShop_db) <-
inputDanCekMons(monster_db, monsterShop_db)

else

    {Keluar dari menu monster}

    output()

    output('    ✖(•ω•`)o Selamat Tinggal!!! ')

    output()

-> monster_db, monsterShop_db

```

p. F14-Load

**KAMUS LOKAL**

arr : list

path, file : string

db : list

**function** cekArrayKosong (arr : list) -> list

**function** afkhFolderAda (path : string) -> boolean

**function** separasiArray (db : list) -> list

**function** bacaFile (file : string) -> string

**function** load() -> list[list]

{I.S Belum ada list yang akan digunakan pada program}

{F.S Membuat list data yang akan dipakai}

{Perubahan :

Tidak ada data -> user\_db, monster\_db, itemInventory\_db,  
itemShop\_db, monsterInventory\_db, monsterShop\_db}

{Sebuah fungsi yang mengambil data-data dalam bentuk .csv  
menjadi bentuk array yang akan digunakan oleh agent ataupun  
admin}

**ALGORITMA**

**function** cekArrayKosong(arr)

data <- []

i traversal [arr]

if not i = ' ' or i = "\n" then

data <- data + i

**function** afkhFolderAda(path)

isPathAda = os.path.exists(path)

-> isPathAda

**function** separasiArray(db)

data <- []

adlhHeader <- True

line traversal [(db)]

if adlhHeader <- False then  
continue

row <- []

Temp <- ""

skipLine <- False

char traversal [(line)]

if char = ';' then  
row.append(temp)  
temp <- ' '

else if char = "\n" then

```

        skipLine <- True
        else if char = " " and not temp then
            continue
        else
            if skipLine and temp then
                row.append(temp)
                temp <- ""
                skipLine <- False
            temp <- temp + char
        if temp then
            row.append(temp)
            Temp <- ""
        if row then
            data.append(row)
    -> data
function bacaFile(file)
    lines <-[]
    line traversal[(file)]
        lines.append(line)
    -> lines

function load()
    Perser <- argparse.ArgumentParser(usage <- "python main.py
<nama_folder>")
    parser.add_argument("x")
    stat <- True

    if len(sys.argv) = 1 then
        sys.exit(1) #folder tidak ada
    args = parser.parse.args()

    if afkhFolderAda(args.x) then
        with open(f"{args.x}/user.csv", 'r') as file
            user_db <-
separasiArray(cekArrayKosong(bacaFile(file)))
        with open(f"{args.x}/monster.csv", 'r') as file
            monster_db <-
separasiArray(cekArrayKosong(bacaFile(file)))
        with open(f"{args.x}/item_inventory.csv", 'r') as
file
            itemInventory_db <-
separasiArray(cekArrayKosong(bacaFile(file)))
        with open(f"{args.x}/item_shop.csv", 'r') as file
            itemShop_db <-
separasiArray(cekArrayKosong(bacaFile(file)))
        with open(f"{args.x}/monster_inventory.csv", 'r') as
file
            monsterInventory_db <-
separasiArray(cekArrayKosong(bacaFile(file)))
        with open(f"{args.x}/monster_shop.csv", 'r') as file

```

```

        monsterShop_db <-
separasiArray(cekArrayKosong(bacaFile(file)))

        return user_db, monster_db, itemInventory_db,
itemShop_db, monsterInventory_db, monsterShop_db, stat
    else
        stat <- False
    -> [], [], [], [], [], [], [], stat

```

#### q. F15 - Save

##### **KAMUS LOKAL**

```

df_user, df_monster, df_monsterShop, df_monsterInventory,
df_itemShop, df_itemInventory : list
folder, parentFolder, fullPath : string

```

**procedure** save (input df\_user : list, df\_monster : list,  
df\_monsterShop : list, df\_monsterInventory : list,  
df\_itemShop : list, df\_itemInventory : list, output file)

{I.S List-list database dalam keadaan semula}

{F.S List-list database akan disimpan kedalam sebuah folder}

{Perubahan :

Membuat folder parent "data\_save" dan folder di dalamnya  
yang berisi data baru}

{Sebuah prosedur yang membuat folder parent dan folder  
penyimpanan lagi didalamnya yang diisi oleh data-data dalam  
bentuk .csv sesuai dengan kondisi terakhir sebelum prosedur  
save dijalankan}

##### **ALGORITMA**

output("Masukkan nama folder : ")

input(folder)

parentFolder <- "data\_save"

fullPath <- os.path.join(parentFolder, folder)

output("Saving...")

delay(0.75)

if not(os.path.isdir(parentFolder)) then

assign(parentFolder, )

output("Membuat folder parent " + parentFolder + "...")

if not(os.path.isdir(fullPath)) then

os.mkdir(fullPath)

output("Membuat folder " + fullPath)

```

{Save data user}
assign(f, "user.csv")
rewrite(f)
line traversal [0...(df_user)]
    text <- line[0] + ";" + line[1] + ";" + line[2] + ";" +
line[3] + ";" + line[4] + "\n"
    write(f, text)
close(f)

{Save data monster}
assign(f, "/monster.csv")
rewrite(f)
line traversal [0...(df_monster)]
    text <- line[0] + ";" + line[1] + ";" + line[2] + ";" +
line[3] + ";" + line[4] + "\n"
    write(f, text)
close(f)

{Save data monster shop}
assign(f, "/monster_shop.csv")
rewrite(f)
line traversal [0...(df_monsterShop)]
    text <- line[0] + ";" + line[1] + ";" + line[2] + "\n"
    write(f, text)
close(f)

{Save data monster inventory}
assign(f, "/monster_inventory.csv")
rewrite(f)
line traversal [0...(df_monsterInventory)]
    text <- line[0] + ";" + line[1] + ";" + line[2] + "\n"
    write(f, text)
close(f)

{Save data item shop}
assign(f, "/item_shop.csv")
rewrite(f)
line traversal [0...(df_itemShop)]
    text <- line[0] + ";" + line[1] + ";" + line[2] + "\n"
    write(f, text)
close(f)

{Save data item inventory}
assign(f, "/item_inventory.csv")
rewrite(f)

```

```

line traversal [0...(df_itemInventory)]
  text <- line[0] + ";" + line[1] + ";" + line[2]
  write(f, text)
close(f)

output(fullPath)

```

r. F16 - Exit

**KAMUS LOKAL**

```

df_user, df_monster, df_monsterShop, df_monsterInventory,
df_itemShop, df_itemInventory : list
inginExit : string
gameMasihJalan : bool

```

```

function exit_(df_user : list, df_monster : list,
df_monsterShop : list, df_monsterInventory : list,
df_itemShop : list, df_itemInventory : list) -> bool

```

```

{I.S List-list database dalam keadaan semula}
{F.S Menutup program secara keseluruhan}
{Perubahan :
  gameMasihJalan -> False}
{Sebuah fungsi yang mengembalikan status apakah program masih
jalan atau tidak berdasarkan input dari pengguna}

```

**ALGORITMA**

```

gameMasihJalan <- true
while gameMasihJalan do
  output("Apakah Anda mau melakukan penyimpanan file
yang sudah diubah? (y/n) ")
  input(inginExit)
  inginExit <- str_lower(ininginExit)

  while (inginExit != "y") and (inginExit != "n") do
    output("Input Tidak Valid!")
    output("Apakah Anda mau melakukan penyimpanan
file yang sudah diubah? (y/n) ")
    input(inginExit)
    inginExit <- str_lower(ininginExit)

  if inginExit = "y" then
    save.save(df_user, df_monster, df_monsterShop,
df_monsterInventory, df_itemShop, df_itemInventory)

```



```
        gameMasihJalan <- false
    else
        gameMasihJalan <- false

-> gameMasihJalan
```

## V. Hasil Testing Program

Berikan tangkapan layar hasil pengujian program berdasarkan fitur-fitur pada spesifikasi. Pada setiap fitur, tangkapan layar minimal berisi: (1) data masukan, (2) data keluaran untuk *input* yang valid dan tidak valid jika terdapat validasi.

### a. F01 - Register

```
>>> register

Masukan username: !!!!!
Username !!!!! salah, silakan menggunakan username lain.

Masukan username: █
```

Gambar V.1 Register tidak valid

```
Masukan username: haegenhebat
Username haegenhebat telah berhasil register.

Masukan password: standbyme
Password Valid.

Silahkan pilih salah satu monster sebagai monster awalmu :)
1. Pikachow
2. Bulbu
3. Zeze
4. Zuko
5. Chacha

>>> CHOOSE YOUR MONSTER: █
```

Gambar V.2 Register valid

### b. F02 - Login

```

X==X==X==X==X==X==X==X==X==X==X==X==X==X==X==X==X==X==X==X==X==X
+ > + > + > Selamat datang di program OWCA ! < + < + < + < +
X==X==X==X==X==X==X==X==X==X==X==X==X==X==X==X==X==X==X==X==X==X

    Ketik REGISTER, LOGIN, atau HELP untuk memulai program

>>> login

====> SILAHKAN LOGIN <====
Username : Asep_Spakbor
Password : asepow123

<=====>
Selamat datang, agent Asep_Spakbor! Ayo kita kalahkan Dr. Asep Spakbor!!!
Masukkan command "HELP" untuk daftar command yang dapat kamu panggil.
>>> 

```

Gambar V.3 Tampilan berhasil login

### c. F03 - Logout

```

Keluar dari inventory.
>>> logout
Dadah, anda sudah logout dari akun

    Ketik REGISTER, LOGIN, atau HELP untuk memulai program

>>> 

```

Gambar V.4 Prosedur logout

### d. F04 - Menu & Help

```

===== HELP =====

Halo Agent! 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: Melakukan pertarungan dengan monster untuk mendapatkan OC
4. Arena: Melakukan set berbagai battle secara berturut-turut
5. Shop: Toko dimana berbagai monster dan potion bisa dibeli menggunakan OC
6. Laboratory: Melakukan upgrade/level up terhadap monster dengan OC
7. Save: Melakukan aksi penyimpanan data untuk digunakan di kemudian hari
8. Exit Game

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

>>> 

```

Gambar V.5 Tampilan menu untuk agent

```

>>> help
===== HELP =====

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

1. Logout: Keluar dari akun yang sedang digunakan
2. Shop Management: Melakukan manajemen pada SHOP sebagai tempat jual beli peralatan Agent
3. Monster Management: Melakukan manajemen pada data & statistik monster
4. Save: Melakukan aksi penyimpanan data untuk digunakan di kemudian hari
5. Exit Game

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

>>> █

```

Gambar V.6 Tampilan menu untuk admin

#### e. F05 - Monster

Penjelasan: Mekanisme fungsi ini telah diimplementasikan di fungsi lainnya sehingga fungsi ini tidak dapat di-testing.

#### f. F06 - Potion

```

<====Turn 2 (Zeze)====>
1. Attack
2. Potion
3. Quit
Pilih perintah : 2
Anda tidak memiliki Potion dalam inventory!

```

Gambar V.7 Tampilan potion saat tidak ada stok

```

<====Turn 1 (Bulbu)====>
1. Attack
2. Potion
3. Quit
Pilih perintah : 2

===== POTION LIST =====
1. Strength Potion (Qty: 0) - Increases ATK Power
2. Resilience Potion (Qty: 2) - Increases DEF Power
3. Healing Potion (Qty: 2) - Restores Health
4. Cancel
2
Sebuah gelembung tameng tiba-tiba muncul melingkupi Bulbu yang membuatnya semakin tangguh dan sulit dilukai.
DEF POWER + 5%
<====Turn 1 (Bulbu)====>
1. Attack
2. Potion
3. Quit
Pilih perintah : █

```

Gambar V.8 Tampilan Potion ketika digunakan pada battle

#### g. F07 - Inventory

```

Masukkan command "HELP" untuk daftar command yang dapat kamu panggil.
>>> inventory
===== INVENTORY LIST (User ID: 2) =====
Jumlah O.W.C.A. Coin-mu sekarang 9999.
1. Monster      (Name: Pikachow, Lvl: 1, HP: 600)
2. Potion       (Type: Strength Potion, Amount: 5)
3. Potion       (Type: Resilience Potion, Amount: 3)

Ketikkan id untuk menampilkan detail item: █

```

Gambar V.9 Tampilan awal inventory

```

Jumlah O.W.C.A. Coin-mu sekarang 9999.
1. Monster      (Name: Pikachow, Lvl: 1, HP: 600)
2. Potion       (Type: Strength Potion, Amount: 5)
3. Potion       (Type: Resilience Potion, Amount: 3)

Ketikkan id untuk menampilkan detail item: 1

MONSTER
Name: Pikachow
ATK Power: 125
DEF Power: 10
HP: 600
Level: 1

Ketikkan id untuk menampilkan detail item: 2

POTION
Type: strength
Quantity: 5

Ketikkan id untuk menampilkan detail item: 3

POTION
Type: resilience
Quantity: 3

```

Gambar V.10 Fitur inspect inventory

```

Ketikkan id untuk menampilkan detail item: 4
Input tidak valid, coba lagi!

Ketikkan id untuk menampilkan detail item: keluar
Keluar dari inventory.
>>> █

```

Gambar V.11 Input tidak valid dan keluar dari inventory

#### h. F08 - Battle

```

>>> battle
<=== CHOOSE YOUR MONSTER ===>
1. Zeze          | Level : 1

Pilih monster anda : 1

OMAGAAA monster Bulbu muncul !!!

<===Battle Start===>
Name : Bulbu
ATK Power : 50
DEF Power : 50
HP : 1200
Level : 2

<====Turn 1 (Zeze)====>
1. Attack
2. Potion
3. Quit
Pilih perintah : 1
BOOM Zeze menyerang Bulbu dengan damage 142
Sisa HP musuh Bulbu sebesar 1058

RAWRRR Giliran Bulbu menyerang!!!
Bulbu menyerang Zeze dengan damage 47
Sisa HP dari Zeze sebesar 253

<====Turn 2 (Zeze)====>
1. Attack
2. Potion
3. Quit
Pilih perintah : █

```

Gambar V.12 Tampilan battle

```

<====Turn 2 (Zeze)====>
1. Attack
2. Potion
3. Quit
Pilih perintah : 3
Anda berhasil kabur dari pertarungan
Exiting battle...
>>> █

```

Gambar V.13 Keluar battle

#### i. F09 - Arena

```

>>> arena
Selamat datang di arena!
<=== CHOOSE YOUR MONSTER ===>
1. Zeze          | Level : 1

Pilih monster anda : 1
===== Stage 1 =====

OMAGAAA monster Bulbu muncul !!!

<===Battle Start===>
Name : Bulbu
ATK Power : 50
DEF Power : 50
HP : 1200
Level : 1

<===Turn 1 (Zeze)===>

```

Gambar V.14 Tampilan awal arena

```

<===Turn 27 (Bulbu)===>
1. Attack
2. Potion
3. Quit
Pilih perintah : 1
BOOM Bulbu menyerang Chacha dengan damage 27
VERY NIICE Chacha berhasil dikalahkan !!!
STAGE CLEARED! Anda mendapatkan 30 OC pada stage ini!

Tekan enter untuk melanjutkan ke stage selanjutnya1
===== Stage 2 =====

OMAGAAA monster Pikachow muncul !!!

```

Gambar V.15 Tampilan *Stage Cleared*

```

RAWRRR Giliran Zuko menyerang!!!
Zuko menyerang Bulbu dengan damage 82
Womp womp agent kalah dalam arena

===== STATS =====
Total hadiah      : 180 OC
Jumlah Stage      : 3
Damage diberikan  : 2793
Damage diterima   : 3879

```

Gambar V.16 Statistik arena

## j. F10 - Shop

```

>>> shop
===== SHOP =====

  \ ( ^ ▽ ^ ) /   Welcome to the SHOP!!!

_____

=> AKSI <=
> Lihat
> Beli
> Keluar

_____

>>> Pilih aksi: █

```

Gambar V.17 Tampilan shop awal (agent)

```

=> AKSI <=
> Lihat
> Beli
> Keluar

_____

>>> Pilih aksi: lihat

_____

=> AKSI <=
> Monster
> Item
> Batal

_____

>>> Mau lihat apa?: 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	300	3	1000
4	Zuko	100	25	800	8	550
5	Chacha	80	30	700	7	600

Gambar V.18 Melihat monster pada shop

```

>>> Pilih aksi: lihat

_____

=> AKSI <=
> Monster
> Item
> Batal

_____

>>> Mau lihat apa?: item

```

ID	Tipe	Stok	Harga
1	Strength Potion	10	50
2	Resilience Potion	5	30
3	Health Potion	3	20



Gambar V.19 Melihat item pada shop

```
>>> Pilih aksi: beli

Jumlah O.W.C.A. Coin-mu sekarang 0

=> AKSI <=
> Monster
> Item
> Batal

>>> Mau beli apa?: monster
>>> Masukkan id monster: 1

OC-mu tidak cukup! Membatalkan pembelian

Jumlah O.W.C.A. Coin-mu sekarang 0
```

Gambar V.20 Membeli dengan O.W.C.A Coin yang tidak cukup

```
=> AKSI <=  
> Lihat  
> Beli  
> Keluar
```

---

```
>>> Pilih aksi: keluar  
  
    ^*(•w•`)o Selamat Tinggal!!!  
  
>>> 
```

Gambar V.21 Keluar dari shop

**k. F11 - Laboratory**

```
>>> laboratory  
=====+ LABORATORY +=====+  
  
      ( ◡◡◡ ) ^Welcome to the LAB!  
  
=>=>=>=>=>=>=>=>=> MONSTER LIST <=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<=<  
  
1. Bulbu (Level: 2)  
  
>>> Pilih aksi (upgrade/keluar):
```

Gambar V.22 Tampilan awal laboratory

```
>>> Pilih aksi (upgrade/keluar): lol  
Input tidak valid!  
  
>>> Pilih aksi (upgrade/keluar): upgrade  
  
<->->->->->->->->->-> UPGRADE PRICE <-<-<-<-<-<-<-<-<-<-<->  
Level 1 -> Level 2: 300 OC  
Level 2 -> Level 3: 600 OC  
Level 3 -> Level 4: 1000 OC  
Level 4 -> Level 5: 1500 OC  
  
O.W.C.A Coin-mu: 290

---


```

```
>>> Pilih Monster: 1  
OC-mu tidak cukup untuk meng-upgrade monster!
```

```
>>> Pilih Monster:
```

Gambar V.23 Mengupgrade dengan O.W.C.A Coin tidak cukup

```
1. Bulbu (Level: 1)

>>> Pilih aksi (upgrade/keluar): keluar

      ^ (•w•)o Selamat Tinggal!!!

>>> █
```

Gambar V.24 Keluar dari laboratory

## I. F12 - Shop Management

```
>>> Pilih aksi: 1
Pilih aksi yang valid!

=> AKSI <=
> Lihat
> Tambah
> Ubah
> Hapus
> Keluar

>>> Pilih aksi: lihat
=> AKSI <=
> Monster
> Item
> Batal
```

Gambar V.25 Memilih aksi di shop management

```
>>> Mau lihat apa?: 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	300	3	1000
4	Zuko	100	25	800	8	550
5	Chacha	80	30	700	7	600

---

```
=> AKSI <=
> Lihat
> Tambah
> Ubah
> Hapus
> Keluar
```

Gambar V.26 Melihat monster di shop management

```
>>> Pilih aksi: tambah
=> AKSI <=
> Monster
> Item
> Batal
```

---

```
>>> Mau tambah apa?: xxx
Pilih opsi yang valid!

=> AKSI <=
> Monster
> Item
> Batal
```

---

```
>>> Mau tambah apa?: bulbu
Pilih opsi yang valid!

=> AKSI <=
> Monster
> Item
> Batal
```

---

```
>>> Mau tambah apa?: monster
```

---

ID	Type	ATK Power	DEF Power	HP
6	Spakbor Monster	3000	50	12000

```
> Ketik "BATAL" untuk keluar
```

Gambar V.27 Menambahkan stok item suatu monster

```

>>> Pilih aksi: ubah
=> AKSI <=
> Monster
> Item
> Batal

>>> Mau ubah apa?: item

ID || Tipe           || Stok || Harga ||
1  || Strength Potion || 10   || 50    ||
2  || Resilience Potion || 5    || 30    ||
3  || Health Potion    || 3    || 20    ||

>>> Masukkan id item: 2
>>> Masukkan stok baru: 7
>>> Masukkan harga baru: 30

Item telah berhasil diubah!
Nama : Resilience Potion
Stok baru : 7
Harga baru: 30

```

Gambar V.28 Mengubah stok item di shop

```

>>> Pilih aksi: hapus
=> AKSI <=
> Monster
> Item
> Batal

>>> Mau hapus apa?: 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        || 300  || 3    || 1000  ||
4  || Zuko     || 100       || 25        || 800  || 8    || 550   ||
5  || Chacha   || 80        || 30        || 700  || 7    || 600   ||

>>> Masukkan id monster: 

```

Gambar V.29 Menghapus monster

### m. F13 - Monster Management

```

>>> monster
===== OWCA-DEX =====

  ♪ •`ω •´♪ SELAMAT DATANG DI OWCA-DEX!!!

Menu
> Lihat
> Tambah
> Keluar

>>> Pilih Aksi: 

```

Gambar V.30 Menu monster management

```

>>> Pilih Aksi: lihat
Pilih aksi yang valid!!

>>> Pilih Aksi: 1

```

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

```

Menu
> Lihat
> Tambah
> Keluar

```

Gambar V.31 Aksi melihat monster

```

>>> Pilih Aksi: 2

Memulai pembuatan Monster baru
>>> Masukkan Type / Nama Monster: Nadil
>>> Masukkan ATK Power: 200
>>> Masukkan DEF Power: 30
>>> Masukkan HP: 900

Menu
> Lihat
> Tambah
> Keluar

>>> Pilih Aksi: 

```

Gambar V.32 Membuat monster baru

```

>>> Pilih Aksi: 3

  ^ (•w•`)o Selamat Tinggal!!!

>>> 

```

Gambar V.33 Keluar dari monster

n. F14 - Load



## Lampiran

### Form MoM Asistensi Tugas Besar IF1210/Dasar Pemrograman Sem. 2 2023/2024

Nomor Asistensi : 1  
No. Kelompok/Kelas : D/K10  
Tanggal asistensi : 3 Mei 2024

Anggota kelompok	NIM / Nama (Hanya yang Hadir)	
	1	16523009/Kesha Mufrih Ramadhan
	2	16523229/Nisrina Azzah Atiqoh
	3	19623009/Haegen Quinston
	4	19623119/Muhammad Rayhan Farrukh
	5	16523039/Anak Agung Gede Aditya Virgadinata
Asisten pembimbing	6	
	NIM / Nama	
	13521024/Ahmad Nadil	

#### Catatan Asistensi:

<b>Rangkuman Diskusi</b>
Diskusi dilakukan di ITB - Korea Cyber Security R&D Center (KOICA) ITB Jatinangor pukul 17.00-17.40 WIB. <ul style="list-style-type: none"><li>- Load - csv diubah ke array/dictionary</li><li>- Save - folder baru isi csv hasil dari load</li><li>- Password tidak perlu dicek, asal tidak kosong</li><li>- Menu - fungsinya sesuai role, ke main</li><li>- F5 &amp; F6 (monster &amp; potion) hanya entitas, diimplementasikan di fungsi lain</li><li>- Database tidak berubah, level 5 kekuatannya tetap sama, semua atribut naik di arraynya</li><li>- Arena - battle loop 5 kali, tiap battle ada hadiah/reward</li><li>- Ada stat</li><li>- Monster management - pastikan nama monster unik dan ngga ada di database</li></ul>
<b>Tindak Lanjut</b>
<ul style="list-style-type: none"><li>- F05 Monster hanya entitas, bukan sebuah program fungsi sendiri, kami memasukkan mekanisme monster ke dalam program battle.</li><li>- F06 Potion hanya entitas, bukan sebuah fungsi sendiri, kami menyimpan array potion akan di main.</li></ul>
<b>Dokumentasi</b>





**Form MoM Asistensi Tugas Besar**  
**IF1210/Dasar Pemrograman**  
**Sem. 2 2023/2024**

Nomor Asistensi : 2  
 No. Kelompok/Kelas : D/K10  
 Tanggal asistensi : 14 Mei 2024

Anggota kelompok	NIM / Nama (Hanya yang Hadir)	
	1	16523009/Kesha Mufrih Ramadhan
	2	16523229/Nisrina Azzah Atiqoh
	3	19623009/Haegen Quinston
	4	
	5	
	6	
Asisten pembimbing	NIM / Nama	
	13521024/Ahmad Nadil	

Catatan Asistensi:

<b>Rangkuman Diskusi</b>
<p>Diskusi dilakukan di ITB - Korea Cyber Security R&amp;D Center (KOICA) ITB Jatinangor pukul 16.00-16.35 WIB.</p> <p>19623119/Muhammad Rayhan Farrukh dan 16523039/Anak Agung Gede Aditya Virgadinata tidak mengikuti asistensi 2 karena sakit.</p> <ul style="list-style-type: none"> <li>- Arg parse : input semua nama folder csv (lebih lengkap di QnA no. 48: skema load save)</li> <li>- Tidak boleh pake global</li> <li>- Pakai array harusnya aman untuk semuanya</li> <li>- Laboratory : belum bikin “terupgrade”, baru print doang</li> <li>- Shop management: bukan ambil dari csv, harusnya dari variabel</li> <li>- Beberapa fungsi masih load langsung, belum pake argumen</li> <li>- Buat notasi algoritmik sendiri tapi tidak perlu terlalu detail. Contoh: output (nama_variabel). Yang penting semua ada notalnya tapi tidak perlu sempurna.</li> <li>- Saran bonus : Jackpot</li> <li>- Kami belum membuat program utama, prioritas MAIN -&gt; save &amp; edit</li> <li>- Setelah register, agent tidak langsung login, login dipanggil secara terpisah di main</li> </ul>
<b>Tindak Lanjut</b>
<ul style="list-style-type: none"> <li>- Finalisasi fungsi load sebelum main, pakai argparse</li> <li>- Membuat fungsi main (utama), baru membuat fungsi lain seperti save dan edit yang belum dipegang</li> <li>- Bugs fixes</li> <li>- Menyelesaikan laporan</li> </ul>
<b>Dokumentasi</b>

