

**LAPORAN TUGAS BESAR  
IF1210 DASAR PEMROGRAMAN**

**K05-B**



Abdullah Farhan	19623305
Ahmad Wicaksono	19623245
Shannon Aurellius A. Lie	19623105
B. Alfin Geraldine Baya	16523175
Bryan R. Ryanto Purba	16523045

**Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung**

**2024**

## HALAMAN KEJUJURAN

### 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 2022/2023.”*

*Abdullah Farhan (19623305)*

*Ahmad Wicaksono (19623245)*

*Shannon Aurellius A. Lie (19623105)*

*B. Alfin Geraldine Baya (16523175)*

*Bryan R. Ryanto Purba (16523045)*

## DAFTAR ISI

<b>HALAMAN KEJUJURAN</b>	<b>4</b>
<b>DAFTAR ISI</b>	<b>5</b>
<b>DAFTAR TABEL</b>	<b>7</b>
<b>DAFTAR GAMBAR</b>	<b>8</b>
<b>DESKRIPSI PERSOALAN</b>	<b>9</b>
<b>PEMBAGIAN KERJA ANGGOTA KELOMPOK</b>	<b>10</b>
Tabel 1. Pembagian Kerja Anggota Kelompok	10
<b>CHECKLIST HASIL RANCANGAN, IMPLEMENTASI, DAN TESTING</b>	<b>11</b>
Tabel 2. Checklist Hasil Rancangan, Implementasi, dan Testing	11
<b>DESAIN COMMAND</b>	<b>12</b>
1. F00 - Random Number Generator (RNG)	12
2. F01 - Register	12
3. F02 - Login	12
4. F03 - Logout	12
5. F04 - Menu & Help	12
6. F05 - Monster	13
7. F06 - Potion	13
8. F07 - Inventory	14
9. F08 - Battle	14
10. F09 - Arena	14
11. F10 - Shop & Currency (Potion/ Monster)	14
12. F11 - Laboratory	14
13. F12 - Shop Management	15
14. F13 - Monster Management	15
15. F14 - Load	15
16. F15 - Save	15
17. F16 - Exit	15
<b>DESAIN KAMUS DATA</b>	<b>17</b>
F00 - Random Number Generator	17
F01 - Register	17
F02 - Login	18
F03 - Logout	18
F04 - Menu & Help	18
F05 - Monster	18
F06 - Potion	19
F07 - Inventory	19

F08 - Battle	20
F09 - Arena	20
F10 - Shop & Currency (Potion/ Monster)	21
F11 - Laboratory	22
F12 - Shop Management	22
F13 - Monster Management	23
F14 - Load	23
F15 - Save	24
F16 - Exit	24
<b>DESAIN DEKOMPOSISI ALGORITMIK &amp; FUNGSIONAL PROGRAM</b>	<b>28</b>
<b>SPESIFIKASI TIAP FUNGSI</b>	<b>34</b>
1. F00 - Random Number Generator (RNG)	34
2. F01 - Register	35
3. F02 - Login	36
4. F03 - Logout	39
5. F04 - Menu & Help	39
6. F05 - Monster	41
7. F06 - Potion	45
8. F07 - Inventory	46
9. F08 - Battle	48
10. F09 - Arena	49
11. F10 - Shop & Currency (Potion/ Monster)	51
12. F11 - Laboratory	51
13. F12 - Shop Management	54
14. F13 - Monster Management	56
15. F14 - Load	58
16. F15 - Save	60
17. F16 - Exit	62
<b>PENGUJIAN PROGRAM</b>	<b>63</b>
Gambar 1. Memulai Program	63
Gambar 2. Tampilan Awal	63
Gambar 2. Tampilan Awal	63
Gambar 3. Tampilan Help	63
Gambar 4. Tampilan Register	63
Gambar 5. Tampilan Login	64
Gambar 6. Tampilan Help(setelah Login)	64
Gambar 7. Tampilan Inventory	64
Gambar 8(a). Tampilan Battle	65
Gambar 8(b). Tampilan Battle	67
Gambar 8(c). Tampilan Battle	67

Gambar 9(a). Tampilan Arena	68
Gambar 9(b). Tampilan Arena	70
Gambar 9(c). Tampilan Arena	72
Gambar 9(d). Tampilan Arena	73
Gambar 10. Tampilan Shop_currency	74
Gambar 11(a). Tampilan Shop_management	74
Gambar 11(b). Tampilan Shop_management	76
Gambar 12. Tampilan Laboratory	76
Gambar 13. Tampilan Monster_management	77
Gambar 14. Tampilan Jackpot	78
Gambar 15. Tampilan Load	78
Gambar 16(a). Tampilan Save (Before)	79
Gambar 16(b). Tampilan Save (After)	79
Gambar 17. Tampilan Logout	79
Gambar 18. Tampilan Exit	79
<b>LAMPIRAN</b>	<b>80</b>

## DAFTAR TABEL

<b>Tabel 1.</b> Pembagian Kerja Anggota Kelompok	8
<b>Tabel 2.</b> Checklist Hasil Rancangan, Implementasi, dan Testing	9

## DAFTAR GAMBAR

<b>Gambar 1.</b> Memulai Program	66
<b>Gambar 2.</b> Tampilan Awal	66
<b>Gambar 3.</b> Tampilan Help	66
<b>Gambar 4.</b> Tampilan Register	66
<b>Gambar 5.</b> Tampilan List Monster	67
<b>Gambar 6.</b> Tampilan setelah register	67
<b>Gambar 7.</b> Tampilan Login	67
<b>Gambar 8.</b> Tampilan Help	68
<b>Gambar 9.</b> Tampilan Inventory	68
<b>Gambar 10.</b> Tampilan Inventory	70
<b>Gambar 11.</b> Tampilan Battle	71
<b>Gambar 12.</b> Tampilan Battle	73
<b>Gambar 13.</b> Tampilan Battle	73
<b>Gambar 14.</b> Tampilan Arena	74
<b>Gambar 15.</b> Tampilan Arena	76
<b>Gambar 16.</b> Tampilan Arena	76
<b>Gambar 17.</b> Tampilan Shop	78
<b>Gambar 18.</b> Tampilan Shop	78
<b>Gambar 19.</b> Tampilan Laboratory	78
<b>Gambar 20.</b> Tampilan Jackpot	79
<b>Gambar 21.</b> Tampilan Load	79
<b>Gambar 22.</b> Tampilan Save	79
<b>Gambar 23.</b> Tampilan Logout	80
<b>Gambar 24.</b> Tampilan Logout	80
<b>Gambar 25.</b> Tampilan Cheat Admin	80
<b>Gambar 26.</b> Tampilan Shop Admin	80
<b>Gambar 27.</b> Tampilan Shop Admin (Lihat)	81
<b>Gambar 28.</b> Tampilan ShopAdmin (Tambah Monster)	81
<b>Gambar 29.</b> Tampilan ShopAdmin (Tambah Potion)	81
<b>Gambar 30.</b> Tampilan Shop Admin (Ubah Monster)	83
<b>Gambar 31.</b> Tampilan Shop Admin (Hapus Monster)	83
<b>Gambar 32.</b> Tampilan Cheat Agent	83
<b>Gambar 33.</b> Tampilan Cheat tambahoc	85
<b>Gambar 34.</b> Tampilan kurangoc	85

## DESKRIPSI PERSOALAN

Tugas ini meminta dibuatkan program permainan berdasarkan kisah Phineas and Ferb di mana Purry si Platypus (Agent P) ingin menghancurkan monster-monster mengerikan ciptaan Dr. Asep Spakbor, ilmuwan jahat. Monster terbaru Dr. Asep Spakbor terlalu kuat, sehingga Purry membutuhkan bantuan dari agen-agen lainnya. Purry bergegas ke markas rahasia O.W.C.A. (Organisasi Warga Cool Abiez) dan bertemu dengan agen-agen lainnya, yaitu kalian. Purry mengatakan bahwa untuk mengalahkan Dr. Asep Spakbor, kalian harus bekerja sama dan merencanakan strategi yang matang untuk mengalahkan monster-monster kuat tersebut. Kalian siap untuk memulai misi pencarian monster di hutan terpencil yang diyakini sebagai tempat tinggal banyak jenis monster, menghadapi segala tantangan yang muncul demi keselamatan kota Danville.

Pada tugas ini kita diminta untuk membuat sebuah program permainan dengan tema monster. Dalam permainan ini, ada beberapa fitur yang harus dibuat, yaitu:

1. Help: Memberikan informasi tentang semua perintah yang tersedia di dalam permainan.
2. Registrasi dan Login: Membuat akun baru dan masuk ke akun yang sudah ada.
3. Battle: Bertarung melawan monster.
4. Arena: Tempat berlatih untuk meningkatkan kemampuan pemain dan monster
5. Shop\_Currency: Tempat Agent membeli monster dan potion
6. Shop\_management: Tempat Admin mengatur Monster Shop dan Item Shop
7. Monster\_management: Tempat Admin membuat monster baru
8. Laboratory: Meningkatkan kemampuan monster yang dimiliki.
9. Jackpot: Kesempatan untuk mendapatkan koin tambahan jika beruntung.
10. Load: Memuat data permainan yang telah disimpan sebelumnya.
11. Save: Menyimpan data permainan setelah bermain.
12. Logout: Keluar dari akun yang sedang digunakan.
13. Exit: Keluar dari program permainan.

Jadi, program ini mencakup berbagai aspek seperti bertarung, berlatih, menyelesaikan misi, dan mengelola akun serta data permainan.



## PEMBAGIAN KERJA ANGGOTA KELOMPOK

**Tabel 1.** Pembagian Kerja Anggota Kelompok

<b>Fitur</b>	<b>Implementasi</b>	<b>NIM Desainer</b>	<b>NIM Coder</b>	<b>NIM Tester</b>
F00 - Random Number Generator	Fungsi random_number	19623245	19623305	19623245 19623305
F01 - Register	Fungsi register	19623105	19623105	19623245 19623305
F02 - Login	Fungsi login	19623105	19623105	19623245 19623305
F03 - Logout	Fungsi logout	16523175	16523175	19623245 19623305
F04 - Menu & Help	Fungsi menu Fungsi help	19623105	19623105	19623245 19623305
F05 - Monster	Fungsi monster	19623305	19623305	19623245 19623305
F06 - Potion	Fungsi potion	16523045	16523045	19623245 19623305
F07 - Inventory	Fungsi inventory	19623305	19623305	19623245 19623305
F08 - Battle	Fungsi battle	19623305	19623305	19623245 19623305
F09 - Arena	Fungsi Arena	19623305	19623305	19623245 19623305
F10 - Shop & Currency	Fungsi shop Fungsi currency	19623305	19623305	19623245 19623305
F11 - Laboratory	Fungsi laboratory	19623245	19623245	19623245 19623305
F12 - Shop Management	Fungsi shop_management	19623305	19623305	19623245 19623305
F13 - Monster Management	Fungsi monster_management	19623245	19623245	19623245 19623305
F14 - Load	Fungsi load	19623245	19623245	19623245

				19623305
F15 - Save	Fungsi save	19623245	19623245	19623245 19623305
F16 - Exit	Fungsi exit	16523045	16523045	19623245 19623305
B03 - Jackpot	Fungsi jackpot	19623245	19623245	19623245 19623305

## CHECKLIST HASIL RANCANGAN, IMPLEMENTASI, DAN TESTING

**Tabel 2.** Checklist Hasil Rancangan, Implementasi, dan Testing

silakan klo mau diedit lagi, ini aku buatin defaultnya

<b>Fitur</b>	<b>Desain</b>	<b>Implementasi</b>	<b>Testing</b>
F00 - Random Number Generator	✓	✓	✓
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	✓	✓	✓
F13 - Monster Management	✓	✓	✓
F14 - Load	✓	✓	✓

F15 - Save	✓	✓	✓
F16 - Exit	✓	✓	✓
Penggabungan code	✓	✓	✓
B04 - JACKPOT!	✓	✓	✓

**Keterangan:** ✓ : sudah selesai dikerjakan;  
X : dikerjakan, tapi belum selesai;  
- : tidak dikerjakan sama sekali

## DESAIN COMMAND

### 1. F00 - Random Number Generator (RNG)

#### **random\_number**

input: (bilangan acak dalam rentang tertentu)

output:

Akan menghasilkan angka acak pada sistem.

### 2. F01 - Register

input:

users (list), monsters (list), monster\_inventory (list), user\_login (list)

output:

Memulai proses registrasi pengguna baru. Pengguna diminta untuk memasukkan username dan password. Sistem akan melakukan validasi username untuk memastikan hanya alfabet, angka, underscore, dan strip yang diterima. Jika username telah terpakai, pengguna diminta untuk memilih username lain. Selanjutnya, pengguna akan diminta untuk memilih monster awal dari daftar monster yang tersedia. Setelah memilih, sistem akan melakukan registrasi dengan menambahkan data pengguna baru ke dalam database pengguna, termasuk role sebagai 'Agent'. Sistem juga akan membuat entri baru dalam inventory monster untuk pengguna tersebut. Setelah registrasi berhasil, pengguna akan diberitahu untuk melakukan login.

### 3. F02 - Login

#### **login**

input:

username → pengguna diminta untuk mengisi username

password → pengguna diminta untuk mengisi password

output:

Akan menampilkan pesan berhasil login serta mengeluarkan data username, dan role atau pesan gagal login.

### 4. F03 - Logout

#### **logout**

input: (tidak ada)

output:

Akan menampilkan pesan berhasil atau gagal logout dari sistem.

### 5. F04 - Menu & Help

#### **Help**

input: (tidak ada)

output:

Berdasarkan status role yang ada pada sistem, help akan memberikan pesan berupa perintah apa saja yang dapat dilakukan. Untuk setiap role, terdapat perintah uniknya masing-masing.

#### 6. F05 - Monster

Desain Command edit\_att\_m:

Input:

- monster\_inv: list{Inventaris monster pengguna}
- monster: list{Data monster}
- user\_login: list{Data login pengguna}

Output:

- edit\_att: dict{Data monster yang akan diedit atributnya}

Desain Command atk:

Input:

- attacker: list{Data penyerang}
- defender: list{Data yang diserang}

Output:

- updated\_defender: list{Data yang diserang setelah diserang}

Desain Command show\_monster:

Input:

- player\_monster: list{Data monster}
- lvl: int{Level monster}

Output:

- None (Output berupa tampilan ke layar)

Desain Command edit\_att\_r\_m:

Input:

- monster: list{Data monster}
- level: int{Level baru}

Output:

- updated\_monster: list{Data monster yang atributnya sudah diperbarui}

#### 7. F06 - Potion

input: (item\_inv: list, user\_login: list)

output:

Mengambil data potion untuk pengguna tertentu dari inventaris. Akan mengembalikan daftar potion beserta kuantitasnya yang dimiliki oleh pengguna.

#### 8. F07 - Inventory

input: (user\_login: list, monster: list, potion: list, monster\_inventory: list)

output:

Menampilkan inventaris pengguna yang mencakup informasi tentang koin OC, monster, dan potion yang dimiliki. Pengguna dapat memilih untuk melihat detail item tertentu.

#### 9. F08 - Battle

Input:

- user\_login: list{Data login pengguna}
- monster: list{Data monster}
- item\_inv: list{Inventaris item pengguna}
- monster\_inv: list{Inventaris monster pengguna}

Output:

- updated\_user\_login: list{Data login pengguna yang diperbarui setelah pertempuran}
- updated\_item\_inv: list{Inventaris item pengguna yang diperbarui setelah pertempuran}

#### 10. F09 - Arena

input: user\_login (list), data\_monster (list), item\_inv (list), monster\_inv (list)

output:

Memulai pertempuran antara pengguna dengan monster di arena. Mempertaruhkan lima monster dengan tingkat kesulitan bertambah setiap kali. Setiap kemenangan akan memberikan pengguna hadiah dalam bentuk OC (O.W.C.A. Coin), dan total damage yang diberikan dan diterima akan dicatat. Pada akhir pertempuran, statistik pertempuran akan ditampilkan.

#### 11. F10 - Shop & Currency (Potion/ Monster)

input: monster\_shop (list), item\_shop (list), monster\_inventory (list), item\_inventory (list), user\_login (list), monster (list)

output:

Menampilkan opsi untuk berbelanja di toko. Pengguna dapat memilih untuk melihat daftar monster atau potion yang tersedia, membeli item, atau keluar dari toko. Saat memilih untuk membeli, sistem akan memvalidasi jumlah O.W.C.A. Coin pengguna dan mengurangi stok dari barang yang dibeli. Jika pembelian berhasil, data pembelian akan ditambahkan ke inventaris pengguna. Jika pengguna memilih untuk keluar, pesan terima kasih akan ditampilkan.

#### 12. F11 - Laboratory

input: user\_login (list), monster (list), monster\_inventory (list)

output:

Memulai proses untuk meningkatkan level monster yang dimiliki pengguna. Pengguna akan disajikan dengan daftar monster yang dimilikinya beserta levelnya.

Pengguna kemudian dapat memilih monster yang ingin ditingkatkan levelnya. Setelah memilih, sistem akan menampilkan harga upgrade untuk setiap level. Pengguna diminta untuk memasukkan nomor monster yang ingin di-upgrade dan memberikan konfirmasi. Jika pengguna setuju, sistem akan melakukan upgrade dan mengurangi koin OC yang dimiliki pengguna sesuai dengan harga upgrade. Jika monster sudah mencapai level

**13. F12 - Shop Management**

maksimum (level 5), maka tidak dapat di-upgrade lagi. Jika pengguna memilih untuk keluar, program akan berhenti.

**14. F13 - Monster Management**

input: monster (list)

output:

Menampilkan opsi pengelolaan data monster. Pengguna dapat memilih untuk melihat daftar semua monster yang tersedia atau menambahkan monster baru ke dalam database. Setelah memilih tindakan, pengguna diminta untuk memasukkan nomor aksi yang diinginkan. Jika pengguna memilih untuk menambahkan monster baru, mereka akan diminta untuk memasukkan informasi monster seperti jenis, ATK Power, DEF Power, dan HP-nya. Setelah memasukkan informasi tersebut, sistem akan menampilkan detail monster yang dibuat dan meminta konfirmasi untuk menambahkannya ke dalam database. Jika pengguna memilih untuk keluar, program akan berhenti.

**15. F14 - Load**

input: (folder\_name: str)

output:

Melakukan pengambilan data dari folder yang ditentukan di dalam direktori "data". Folder tersebut berisi file-file CSV yang berisi data-data pengguna, monster, inventaris monster, inventaris item, dan lainnya. Setelah data berhasil dimuat, akan dikembalikan dalam bentuk tuple yang berisi data-data yang diambil dari file CSV. Jika folder tidak ditemukan, akan memberikan pesan error.

**16. F15 - Save**

input: user (list), user\_login (list), monster (list), monster\_shop (list), monster\_inventory (list), item\_shop (list), item\_inventory (list)

output:

Meminta pengguna untuk memasukkan nama folder untuk menyimpan data. Jika folder sudah ada, data akan disimpan di dalamnya. Jika folder belum ada, folder baru akan dibuat. Setelah data berhasil disimpan, akan memberikan pesan berhasil menyimpan data di folder yang sesuai.

**17. F16 - Exit**

input: user (list), user\_login (list), monster (list), monster\_shop (list), monster\_inventory (list), item\_inventory (list), item\_shop (list)

output:

Meminta pengguna untuk mengkonfirmasi apakah ingin menyimpan perubahan yang telah dilakukan sebelum keluar. Jika pengguna menyetujui penyimpanan, perubahan pada data akan disimpan dan program akan keluar. Jika pengguna tidak menyetujui penyimpanan, program akan langsung keluar.



## DESAIN KAMUS DATA

### F00 - Random Number Generator

#### FUNGSI PEMBANTU - generate\_random

##### KAMUS LOKAL

```
constant modulus = 2147483647 # Bilangan prima terbesar
yang dapat diwakili dalam bilangan 32-bit
constant multiplier = 16807 # Multiplier yang relatif
prima dengan modulus
constant increment = 0 # Increment dapat diabaikan
karena tidak terlalu mempengaruhi kualitas bilangan acak
```

#### FUNGSI PEMBANTU - random\_number\_arr

##### KAMUS LOKAL

```
constant modulus = 2147483647 # Bilangan prima terbesar
yang dapat diwakili dalam bilangan 32-bit
result = []
```

#### F00 - function random\_number

##### KAMUS LOKAL

```
modulus : integer
next_value : integer
```

### F01 - Register

```
users: list{Daftar pengguna}
monsters: list{Daftar monster}
monster_inventory: list{Inventory monster}
user_login: list{Data login pengguna}
username: string{Nama pengguna yang dimasukkan}
password: string{Kata sandi yang dimasukkan}
new_agent_id: integer{ID baru untuk pengguna}
new_agent: list{Data pengguna baru yang akan ditambahkan}
data_role: string{Peran pengguna baru}
data_username: string{Nama pengguna untuk data login}
data_id: string{ID pengguna untuk data login}
data_oc: string{Jumlah O.W.C.A. Coin untuk data login}
for_monster_inventory: list{Data inventory monster baru untuk
pengguna}
monster_choice: integer{Pilihan monster awal oleh pengguna}
```

### **F02 - Login**

```
type user_login : < id : string, username : string, role :  
string, oc : string, data_status_login : string >  
account_found : boolean  
user : user_login  
data_username : string
```

### **F03 - Logout**

```
user: list{Daftar pengguna}  
user_login: list{Data login pengguna}  
defaultkan_data: function{Mengembalikan data login ke nilai  
default}  
save_oc_user_login: function{Menyimpan data login ke dalam  
database}  
logout: function{Proses logout pengguna}  
user_login[1][0]: string{ID pengguna dalam data login}  
user_login[1][1]: string{Nama pengguna dalam data login}  
user_login[1][2]: string{Peran pengguna dalam data login}  
user_login[1][3]: string{Jumlah O.W.C.A. Coin dalam data login}  
user_login[1][4]: string{Status login dalam data login}
```

### **F04 - Menu & Help**

#### **F04 - procedure Help**

##### **KAMUS LOKAL**

```
data_status_login : boolean  
data_role : string
```

### **F05 - Monster**

- RNG: module{Modul untuk menghasilkan angka acak}
- math: module{Modul untuk operasi matematika}
- colorizer: module{Modul untuk memberikan warna pada teks}
- random\_number: function{Fungsi untuk menghasilkan angka acak}
- edit\_att\_m: function{Fungsi untuk mengedit atribut monster}
  - monster\_inv: list{Inventaris monster}
  - monster: list{Daftar monster}
  - user\_login: list{Data login pengguna}
- atk: function{Fungsi untuk menyerang monster}
  - attacker: list{Pengirim serangan}
  - defender: list{Penerima serangan}

- show\_monster: function{Fungsi untuk menampilkan detail monster}
  - player\_monster: list{Data monster}
  - lvl: int{Level monster}
- edit\_att\_r\_m: function{Fungsi untuk mengedit atribut monster berdasarkan level}
  - monster: list{Data monster}
  - level: int{Level monster}
- monst\_att: function{Fungsi untuk menghitung nilai atribut monster}
  - level: int{Level monster}
  - nilai: str{Nilai atribut monster}
- monster\_art\_user: function{Fungsi untuk menampilkan seni monster pemain}
- monster\_art\_musuh: function{Fungsi untuk menampilkan seni monster musuh}

#### F06 - Potion

item\_inv: List dari data inventory item.

user\_login: Data pengguna yang sedang login.

data\_id: ID data pengguna yang sedang login.

data\_p: List yang akan berisi item yang dimiliki oleh pengguna yang sedang login setelah di-load.

#### F07 - Inventory

#### F08 - Inventory

- inventory: function{Fungsi untuk menampilkan inventory pengguna}
  - user\_login: list{Data login pengguna}
  - monster: list{Data monster}
  - potion: list{Data potion}
  - monster\_inventory: list{Data inventaris monster}
- data\_oc: str{Jumlah O.W.C.A. Coin pengguna}
- inventory: dict{Data inventaris}
  - user\_id\_inventory: list{ID pengguna inventaris}
  - type: list{Jenis item}
  - quantity: list{Kuantitas item}
  - user\_id\_monster: list{ID pengguna monster}
  - monster\_id: list{ID monster}
  - monster\_level: list{Level monster}
  - monsters: list{Daftar monster}
  - hp\_monst: list{HP monster}

- atk\_monst: list{ATK Power monster}
- def\_monst: list{DEF Power monster}

#### F08 - Battle

- battle: function{Fungsi untuk memulai pertempuran}
  - user\_login: list{Data login pengguna}
  - monster: list{Data monster}
  - item\_inv: list{Inventaris item pengguna}
  - monster\_inv: list{Inventaris monster pengguna}
- id\_monster: int{ID monster yang dipilih untuk pertempuran}
- monster\_lvl: int{Level monster yang dipilih untuk pertempuran}
- random\_monster: list{Informasi monster yang dipilih untuk pertempuran}
- result: dict{Hasil pertempuran}
- reward: int{Jumlah OC yang didapatkan setelah memenangkan pertempuran}
- fight: function{Fungsi untuk mengeksekusi pertempuran}
  - monster\_lvl: int{Level monster yang dipilih untuk pertempuran}
  - random\_monster: list{Informasi monster yang dipilih untuk pertempuran}
  - user\_login: list{Data login pengguna}
  - item\_inv: list{Inventaris item pengguna}
  - monster\_inv: list{Inventaris monster pengguna}
  - monster: list{Data monster}
- user\_hp: int{HP pengguna selama pertempuran}
- monster\_hp: int{HP monster selama pertempuran}
- user\_atk: int{ATK pengguna selama pertempuran}
- monster\_atk: int{ATK monster selama pertempuran}
- user\_def: int{DEF pengguna selama pertempuran}
- monster\_def: int{DEF monster selama pertempuran}
- turn: int{Giliran pertempuran}
- battle\_result: dict{Hasil pertempuran}

#### F09 - Arena

```

user_login: list{Data login pengguna}
data_monster: list{Daftar data monster}
item_inv: list{Daftar inventaris item}

```

```

monster_inv: list{Daftar inventaris monster}
id_monster: list{ID monster yang dipilih}
stat: dict{Statistik pertarungan}
result: dict{Hasil pertarungan}
reward: integer{Hadiah yang diperoleh}
monster: list{Data monster}
monster_level: integer{Level monster}
show_arena_stat: function{Menampilkan statistik arena}
arena: function{Simulasi pertarungan arena}

```

#### F10 - Shop & Currency (Potion/ Monster)

- shop\_currency: function{Fungsi untuk mengelola transaksi di toko}
  - monster\_shop: list{Data toko monster}
  - item\_shop: list{Data toko item}
  - monster\_inventory: list{Inventaris monster pengguna}
  - item\_inventory: list{Inventaris item pengguna}
  - user\_login: list{Data login pengguna}
  - monster: list{Data monster}
- bubble\_sort\_id\_m: function{Fungsi untuk mengurutkan ID monster}
  - arr: list{Array ID monster}
  - dict: dict{Dictionary data monster}
- change\_type\_id: function{Fungsi untuk mengubah tipe data ID}
  - arr: list{Array ID}
- change\_type\_str: function{Fungsi untuk mengubah tipe data string}
  - arr: list{Array}
- lihat\_monster: function{Fungsi untuk menampilkan daftar monster di toko}
  - monster\_shop: list{Data toko monster}
  - monster: list{Data monster}
- lihat\_potion: function{Fungsi untuk menampilkan daftar potion di toko}
  - item\_shop: list{Data toko item}
- beli\_monster: function{Fungsi untuk membeli monster dari toko}
- id\_m: int{ID monster yang ingin dibeli}
  - monster\_shop: list{Data toko monster}
  - monster\_inventory: list{Inventaris monster pengguna}
  - user\_login: list{Data login pengguna}
  - monster: list{Data monster}
- beli\_potion: function{Fungsi untuk membeli potion dari toko}

- id\_p: int{ID potion yang ingin dibeli}
- qty: int{Jumlah potion yang ingin dibeli}
- item\_inventory: list{Inventaris item pengguna}
- item\_shop: list{Data toko item}
- user\_login: list{Data login pengguna}

#### F11 - Laboratory

- laboratory: function{Fungsi untuk mengelola proses laboratorium}
  - user\_login: list{Data login pengguna}
  - monster: list{Data monster}
  - monster\_inventory: list{Data inventaris monster}
- load\_data\_m: function{Fungsi untuk memuat data monster}
  - user\_login: list{Data login pengguna}
  - monster: list{Data monster}
  - monster\_inv: list{Data inventaris monster}
- lihat: function{Fungsi untuk menampilkan data monster dan harga upgrade}
  - data\_m: dict{Data monster}
- upgrade\_m: function{Fungsi untuk meng-upgrade monster}
  - lvl\_now: int{Level monster saat ini}
  - pilih: int{Pilihan monster yang ingin di-upgrade}
  - data\_m: list{Data monster}
  - monster\_inv: list{Data inventaris monster}
  - user\_login: list{Data login pengguna}

#### F12 - Shop Management

- shop\_management: function{Fungsi untuk mengelola data toko}
  - monster\_shop: list{Data toko monster}
  - item\_shop: list{Data toko item}
  - monster: list{Data monster}
- bubble\_sort: function{Fungsi untuk mengurutkan data}
  - arr: list{List yang ingin diurutkan}
- cari\_yg\_gk\_ada: function{Fungsi untuk mencari data yang tidak ada di dalam list}
  - a: list{List pertama}
  - b: list{List kedua}
- check\_m\_in\_shop: function{Fungsi untuk memeriksa keberadaan monster dalam toko}
  - shop\_m: dict{Data toko monster}
  - id: int{ID monster yang ingin diperiksa}

- `check_p_in_shop: function{Fungsi untuk memeriksa keberadaan potion dalam toko}`
  - `shop_p: dict{Data toko potion}`
  - `type: str{Jenis potion yang ingin diperiksa}`
- `tambah_m: function{Fungsi untuk menambah data monster ke toko}`
  - `monster: list{Data monster}`
  - `monster_shop: list{Data toko monster}`
- `tambah_p: function{Fungsi untuk menambah data potion ke toko}`
  - `item_shop: list{Data toko item}`
- `ubah_m: function{Fungsi untuk mengubah data monster dalam toko}`
  - `monster_shop: list{Data toko monster}`
  - `monster: list{Data monster}`
- `ubah_p: function{Fungsi untuk mengubah data potion dalam toko}`
  - `item_shop: list{Data toko item}`
- `hapus_m: function{Fungsi untuk menghapus data monster dari toko}`
  - `monster_shop: list{Data toko monster}`
  - `monster: list{Data monster}`
- `hapus_p: function{Fungsi untuk menghapus data potion dari toko}`
  - `item_shop: list{Data toko item}`

### F13 - Monster Management

- `monster_management: function{Fungsi untuk mengelola data monster}`
  - `monster: list{Data monster}`
- `check_m: function{Fungsi untuk memeriksa keberadaan monster berdasarkan jenis}`
  - `monster: list{Data monster}`
  - `type_m: str{Jenis monster yang ingin diperiksa}`
  - `found: bool{Status keberadaan monster}`
- `lihat_m: function{Fungsi untuk menampilkan data monster}`
  - `monster: list{Data monster}`
- `buat_m: function{Fungsi untuk membuat monster baru}`
  - `monster: list{Data monster}`

### F14 - Load

`argparse: module{Modul untuk membuat alat baris perintah}`

```

os: module{Modul untuk interaksi dengan sistem operasi}
sys: module{Modul yang memberikan akses ke beberapa variabel yang
digunakan atau dipelihara oleh interpreter dan untuk berinteraksi
dengan interpreter itu sendiri}
time: module{Modul untuk manajemen waktu}
parser: ArgumentParser{Objek parser argumen}
args: namespace{Nama ruang untuk menyimpan argumen baris perintah
yang di-parse}
folder_name: string{Nama folder yang akan dimuat}
folder_path: string{Path lengkap folder yang akan dimuat}
user: list{Data pengguna yang dimuat dari file CSV}
user_login: list{Data login pengguna yang dimuat dari file CSV}
monster: list{Data monster yang dimuat dari file CSV}
monster_inventory: list{Inventaris monster yang dimuat dari file
CSV}
monster_shop: list{Data toko monster yang dimuat dari file CSV}
item_inventory: list{Inventaris item yang dimuat dari file CSV}
item_shop: list{Data toko item yang dimuat dari file CSV}
read_csv_folder: function{Membaca file CSV di dalam folder
tertentu}
load: function{Memuat data dari folder tertentu}

```

#### F15 - Save

```

user: list{Data pengguna}
user_login: list{Data login pengguna}
monster: list{Data monster}
monster_shop: list{Data toko monster}
monster_inventory: list{Inventaris monster}
item_shop: list{Data toko item}
item_inventory: list{Inventaris item}
folder_name: string{Nama folder untuk penyimpanan data}
os: module{Modul untuk interaksi dengan sistem operasi}
time: module{Modul untuk manajemen waktu}
save: function{Menyimpan data ke dalam folder}
new_save: function{Membuat folder baru dan menyimpan data}

```

#### F16 - Exit

##### F06 - Exit

```

- Save: module{Modul untuk menyimpan data}
- save_oc_user_login: function{Fungsi untuk menyimpan data OC
pengguna ke dalam database}
- user: list{Daftar pengguna}

```



- user\_login: list{Data login pengguna}
- Exit: function{Fungsi untuk keluar dari program}
- user: list{Daftar pengguna}
- user\_login: list{Data login pengguna}
- monster: list{Daftar monster}
- monster\_shop: list{Data toko monster}
- monster\_inventory: list{Inventaris monster}
- item\_inventory: list{Inventaris item}
- item\_shop: list{Data toko item}
- konfirmasi: string{Variabel untuk menyimpan input konfirmasi}
- konfirmasi: string{Variabel untuk menyimpan input konfirmasi}

## **FUNGSI PEMBANTU**

### **1. buat\_shop.py**

function load\_data\_m\_shop\_buat\_shop\_c

function load\_data\_p\_shop\_buat\_shop\_c

### **2. enkripsi.py**

function manual\_index

**KAMUS LOKAL**

n: Karakter yang dicari di dalam array a.

a: Array yang akan dicek untuk mencari karakter n.

function encode

**KAMUS LOKAL**

alphabet\_lower: String

alphabet\_upper: String yang berisi alfabet huruf besar berulang dua kali.

cipher: Pesan yang akan dienkripsi.

function decode

**KAMUS LOKAL**

alphabet\_lower: String yang berisi alfabet huruf kecil berulang dua kali.

alphabet\_upper: String yang berisi alfabet huruf besar berulang dua kali.

message: Pesan hasil dekripsi.

### **3. Jackpot.py**

J1 - Jackpot

- jackpot: function{Fungsi untuk bermain jackpot}
- user\_login: list{Data login pengguna}
- monster: list{Data monster}
- monster\_inventory: list{Inventaris monster pengguna}
- data\_oc: int{Jumlah OC (Overworld Coin) pengguna}

- data\_id: str{ID pengguna}
- items: dict{Daftar item dalam jackpot}
- mulai: str{Pilihan untuk memulai permainan}
- result: list{Hasil acak dari permainan jackpot}
- random\_id: int{ID acak untuk monster yang didapat dalam jackpot}
- found: bool{Status pengecekan keberadaan monster dalam inventaris}

#### J2 - Check ID Monster

- check\_id\_m: function{Fungsi untuk memeriksa keberadaan monster dalam inventaris}
  - monster: list{Data monster}
  - id\_m: str{ID monster yang akan diperiksa}
  - data\_id: str{ID pengguna}
- found: bool{Status keberadaan monster dalam inventaris}

#### J3 - Daftar Item Jackpot

- items: dict{Daftar item dalam jackpot}
  - 0: list{Item Beras}
    - Nama: str{"Beras"}
    - Harga: int{50 OC}
  - 1: list{Item Telur}
    - Nama: str{"Telur"}
    - Harga: int{100 OC}
  - 2: list{Item Kopi}
    - Nama: str{"Kopi"}
    - Harga: int{200 OC}
  - 3: list{Item Baju}
    - Nama: str{"Baju"}
    - Harga: int{300 OC}
  - 4: list{Item Koin}
    - Nama: str{"Koin"}
    - Harga: int{400 OC}

#### J4 - Eksekusi Jackpot

- Eksekusi berbagai kemungkinan hasil permainan jackpot, termasuk pengurangan OC, penambahan OC, dan penambahan monster ke inventaris pengguna.

### 4. parseran.py

P1 - Baca CSV Folder

```
- read_csv_folder: function{Membaca file CSV dalam suatu folder
dan mengembalikan isi file sebagai list}
  - file: str{Nama file CSV}
  - folder_path: str{Path folder tempat file CSV berada}
- values: list[list[any]]{Isi file CSV yang dibaca}
```

#### P2 - Pisah Baris CSV

```
- split: function{Memisahkan nilai-nilai dalam satu baris CSV dan
mengembalikan sebagai list}
  - line: str{Baris CSV yang akan dipisahkan}
- out: list[any]{List hasil pemisahan nilai dalam baris CSV}
```

#### P3 - Hapus Matrix

```
- matrix_pop: function{Menghapus item tertentu dari sebuah
matriks}
  - matrix: list{Matriks yang akan dihapus itemnya}
  - item: list{Item yang akan dihapus dari matriks}
- new_matrix: list{Matriks baru setelah item dihapus}
```

#### P4 - Gabung Manual

```
- manual_join: function{Menggabungkan elemen-elemen list dengan
separator tertentu}
  - separator: str{Pemisah antar elemen}
  - items: list{List elemen yang akan digabungkan}
- result: str{String hasil gabungan elemen list}
```

#### P5 - Tulis CSV ke Folder

```
- write_csv_to_folder: function{Menulis data matriks ke file CSV
dalam suatu folder}
  - foldername: str{Nama folder tempat file CSV akan disimpan}
  - csv: str{Nama file CSV}
  - matrix: list{Matriks data yang akan ditulis ke file CSV}
```

#### P6 - Salin Array

```
- copy: function{Menghasilkan salinan array}
  - array: list{Array yang akan disalin}
- arr_copy: list{Salinan array}
```

## DESAIN DEKOMPOSISI ALGORITMIK & FUNGSIONAL PROGRAM

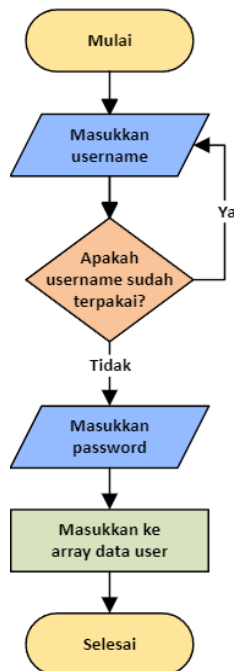
flowchart (Note: tidak sempat selesai dikarenakan anggota kami ada yang sakit, dan menurut asisten kelompok lain flowchart tidak wajib dibuat)

1. F00 - Random Number Generator (RNG)

Gambar 1 : F00 - Random Number Generator (RNG)

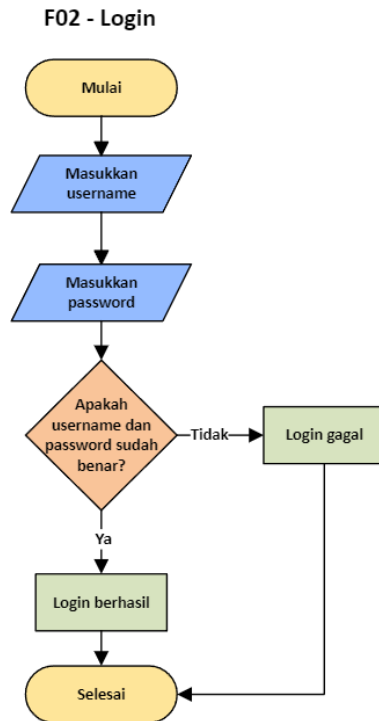
2. F01 - Register

F01 - Register



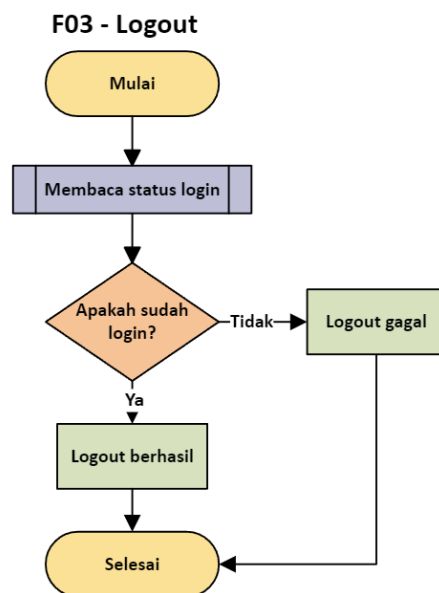
Gambar 2 : F01 - Register

3. F02 - Login



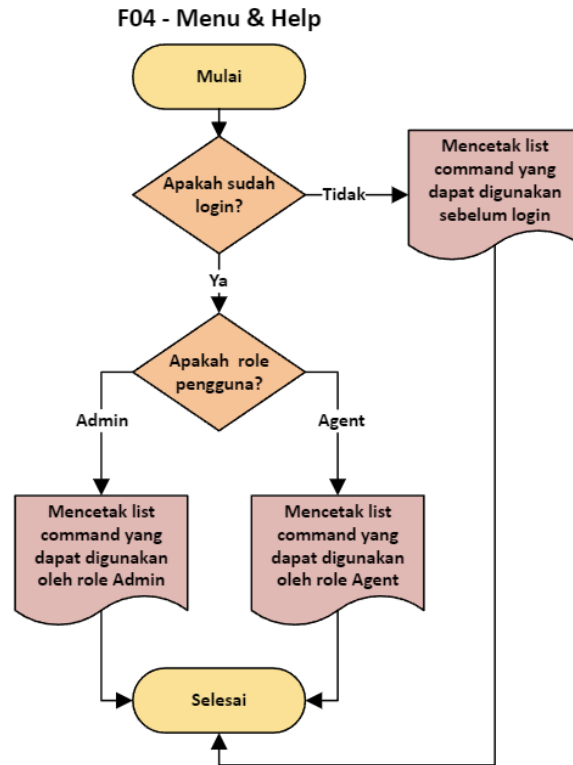
Gambar 3 : F02 - Login

#### 4. F03 - Logout



Gambar 4 : F03 - Logout

#### 5. F04 - Menu & Help



Gambar 5 : F04 - Menu & Help

6. F05 - Monster

Gambar 6 : F05 - Monster

7. F06 - Potion

Gambar 7 : F06 - Potion

8. F07 - Inventory

Gambar 8 : F07 - Inventory

9. F08 - Battle

Gambar 9 : F08 - Battle

10. F09 - Arena

Gambar 10 : F09 - Arena

11. F10 - Shop & Currency (Potion/ Monster)

Gambar 11 : F10 - Shop & Currency (Potion/ Monster)

12. F11 - Laboratory

Gambar 12 : F11 - Laboratory

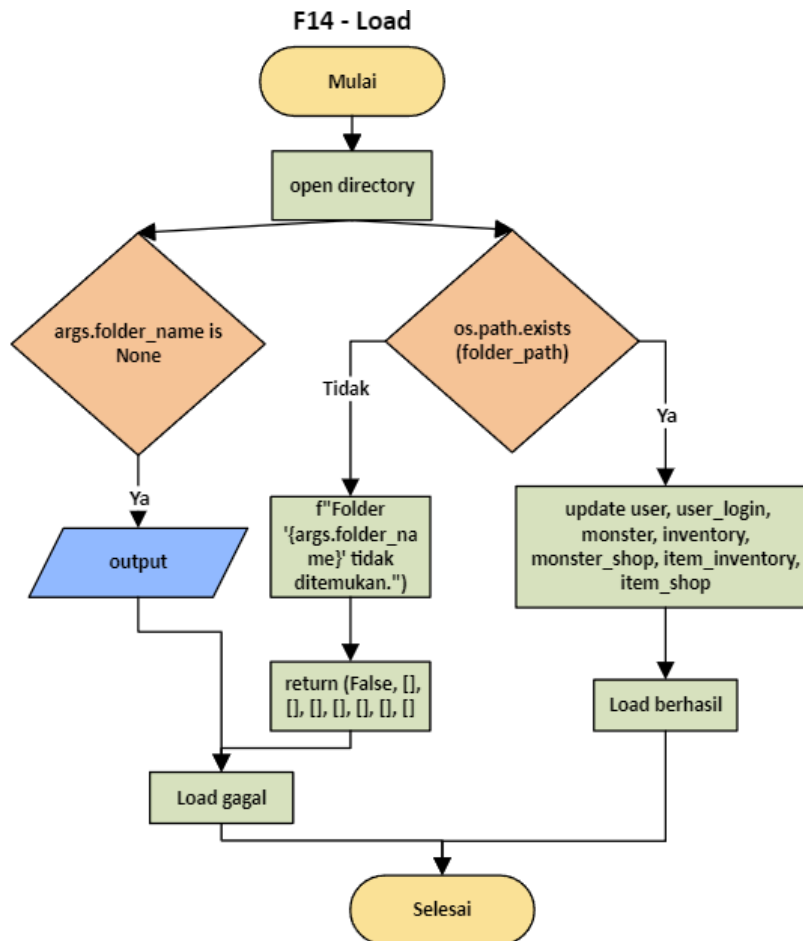
13. F12 - Shop Management

Gambar 13 : F12 - Shop Management

14. F13 - Monster Management

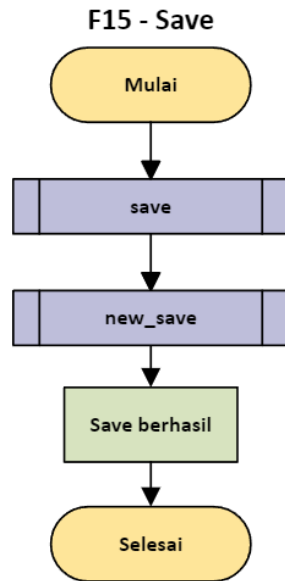
Gambar 14 : F13 - Monster Management

15. F14 - Load



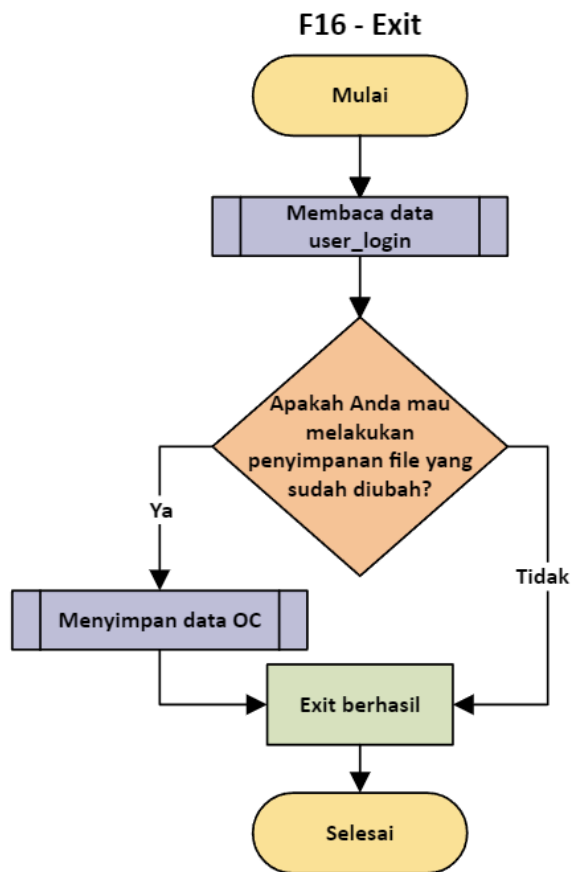
Gambar 15 : F14 - Load

16. F15 - Save



Gambar 16 : F15 - Save

17. F16 - Exit



Gambar 17 : F16 - Exit





## SPESIFIKASI TIAP FUNGSI

### 1. F00 - Random Number Generator (RNG)

#### **FUNGSI - generate\_random**

{Fitur ini berfungsi membuat fungsi generator bilangan acak kongruensial linier (LCG)}  
{I.S seed=None}  
{F.S menghasilkan bilangan acak dalam rentang tertentu}

#### **KAMUS LOKAL**

modulus: integer  
multiplier: integer  
increment: integer

#### **ALGORITMA**

```
function generate_random(seed = None) -> integer
    modulus <- 2147483647
    multiplier <- 16807
    increment <- 0
    if seed is None then
        seed <- datetime.now().microsecond + os.getpid()
    next_value <- (multiplier * seed + increment) % modulus
    → next_value
```

#### **FUNGSI - random\_number**

{Fitur ini berfungsi menghasilkan angka acak pada sistem}  
{I.S range\_of\_number terdefinisi}  
{F.S menghasilkan bilangan acak dalam rentang tertentu}

#### **KAMUS LOKAL**

next\_value : integer  
modulus : integer

#### **ALGORITMA**

```
from datetime import datetime
import os
function random_number(range_of_random=None)-> integer
    modulus <- 2147483647
    next_value <- generate_random()
    if range_of_random is not None then
        min_range, max_range <- range_of_random
```

```

        next_value ← (next_value / modulus) * (max_range -
        min_range) + min_range
    → int(next_value)

```

#### **FUNGSI - random\_number\_arr**

{Fitur ini berfungsi menghasilkan angka acak pada sistem}  
 {I.S range\_of\_number terdefinisi}  
 {F.S menghasilkan array bilangan acak dalam rentang tertentu}

#### **KAMUS LOKAL**

```

    modulus: integer
    result: []

```

#### **ALGORITMA**

```

function    random_number_arr(range_of_random=None,    n=1)    ->
integer
    rng ← generate_random()
    if range_of_random is not None then
        min_range, max_range ← range_of_random
        i traversal [0..N]
            next_value ← (next(rng) / modulus) * (max_range
            - min_range) + min_range
            result.append(int(next_value))
    else
        i traversal [0..N]
            result.append(next(rng))
    → result

```

## **2. F01 - Register**

#### **FUNGSI - register**

{Fitur ini berfungsi memanggil fungsi register\_user}  
 {I.S user\_login terdefinisi}  
 {F.S berhasil register}

#### **KAMUS LOKAL**

```

    users : list

```

#### **ALGORITMA**

```

function    register_user(users:list,    monsters:list,
    monster_inventory:list, user_login:list)->(list,list)
    if user_login[1][4] == "True" then

```

```

        output("Register gagal! Anda telah login dengan
username", user_login[1][1], "silahkan lakukan 'LOGOUT' sebelum
melakukan register'")
        -> users, monster_inventory
    else
        input(username)
        input(password)
        if not all (char in
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz")
            output("Username hanya boleh berisi alfabet,
angka, underscore, dan strip!")
            -> users, monster_inventory
        output("Silahkan pilih salah satu monster sebagai monster
awalmu.")
        i traversal [1..len(monster)]
            output(i, ".", monster[i][1])
        input(monster_choice)
        while monster_choice < 1 or monster_choice > len(monster)
        do
            output("Pilihan monster tidak valid")
            input(monster_choice)
        new_agent_id <- len(users) if users else 1
        new_agent <- [new_agent_id, username, encode(password),
"Agent", 0]
        users.append(new_agent)
        data_role <- users[new_agent_id][3]
        data_username <- username
        data_id <- str(new_agent_id)
        data_oc = "0"
        for_monster_inventory = [str(new_agent_id),
str(monster_choice), "1"]
        monster_inventory.append(for_monster_inventory)
        output("Selamat datang Agent", username, ". Mari kita
mengalahkan Dr. Asep Spakbor dengan",
monsters[monster_choice][1], "!")
        user_login <- login.save_data_login(data_id,
data_username, data_role, data_oc, user_login)
        -> users, monster_inventory

```

### 3. F02 - Login

#### FUNGSI PEMBANTU - save\_data\_login

```

{Fitur ini berfungsi inisialisasi penyimpanan data login}
{I.S. user_login terdefinisi}
{F.S. terinisialisasi list penyimpanan data login}

```

#### KAMUS LOKAL

```
type user_login : < id : string, username : string, role :  
string, oc : string, data_status_login : string >
```

#### ALGORITMA

```
user_login[1][0] ← id  
user_login[1][1] ← username  
user_login[1][2] ← role  
user_login[1][3] ← oc  
user_login[1][4] ← 'True'  
→ user_login
```

#### FUNGSI - login

```
{Fitur ini berfungsi masuk ke dalam akun yang sudah terdaftar}  
{I.S. user_login terdefinisi}  
{F.S. tercetak 'Anda telah login dengan username',  
user_login[1][1], ' silahkan lakukan "LOGOUT" sebelum melakukan  
login kembali apabila sudah login. Kemudian tercetak Selamat  
datang," data_role, username,"! Masukkan command 'help' untuk  
daftar command yang dapat kamu panggil." apabila belum login  
sebelumnya, username, dan password benar. Akan tercetak  
"Password salah!" apabila belum login sebelumnya tetapi  
memasukkan username dan password yang salah. Akan tercetak  
"Username tidak terdaftar" apabila username salah.}
```

#### KAMUS LOKAL

```
type user_login : < id : string, username : string, role :  
string, oc : string, data_status_login : string >  
account_found : boolean  
data_username : string
```

#### ALGORITMA

```
if user_login[1][4] = "True" then  
    output('Anda telah login dengan username',  
user_login[1][1], ' silahkan lakukan "LOGOUT" sebelum  
melakukan login kembali.')
```

else

```
    input(username)  
    input(password)  
    repeat(len(user)) times  
        if username = user[i][1] then
```

```

        if password == decode(user[i][2])then
            account_found ← True
            data_username ← username
            data_role ← user[i][3]
            data_id ← user[i][0]
            data_oc ← user[i][4]
            user_login ←
            save_data_login(data_id,data_username,data
            _role,data_oc,user_login)
            output("Selamat datang,", data_role ,
username,"!")

            output("Masukkan command 'help' untuk
daftar command yang dapat kamu panggil.")
            break
        else
            account_found ← True
            output("Password salah!")
            break
    else
        account_found ← False

    if not account_found then
        output("Username tidak terdaftar")
{REALISASI FUNGSI/PROSEDUR}
    function decode(cipher:str) -> str:
        alphabet_lower <-
"abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz"
        alphabet_upper <-
"ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ"
        message <-d ""
        letter traversal cipher:
            if letter in alphabet_lower then
                index <- manual_index(letter, alphabet_lower)
                message <- message + alphabet_lower[index - 13]
            elif letter in alphabet_upper:
                index = manual_index(letter, alphabet_upper)
                message <- message + alphabet_upper[index - 13]
            else
                message <- message + letter
        -> message

    function save_data_login
(id:str,username:str,role:str,oc:str,user_login:list) -> list
        user_login[1][0] ← id
        user_login[1][1] ← username

```

```

user_login[1][2] ← role
user_login[1][3] ← oc
user_login[1][4] ← 'True'
-> user_login

```

#### 4. F03 - Logout

##### **FUNGSI PEMBANTU - defaultkan\_data**

```

{Fitur ini berfungsi untuk menjadikan data umum sistem}
{I.S user_login terdefinisi}
{F.S. menghasilkan list data umum sistem}

```

##### **KAMUS LOKAL**

```

type user_login : < id : string, username : string, role :
string, oc : string, data_status_login : string >

```

##### **ALGORITMA**

```

user_login[1][0] ← ''
user_login[1][1] ← ''
user_login[1][2] ← ''
user_login[1][3] ← ''
user_login[1][4] ← 'False'
→ user_login

```

##### **FUNGSI - logout**

```

{Fitur ini berfungsi untuk melakukan logout pada sistem }
{I.S user_login terdefinisi}
{F.S. tercetak "Anda telah logout" apabila telah login
sebelumnya dan tercetak 'Anda belum login, silahkan login
terlebih dahulu sebelum melakukan logout' apabila belum login
sebelumnya.}

```

##### **KAMUS LOKAL**

```

type user_login : < id : string, username : string, role :
string, oc : string, data_status_login : string >
user : list
user_login : list

```

##### **ALGORITMA**

```

if user_login[1][4]='True' then
    user_login ← defaultkan_data(user_login)
    user ← save_oc_user_login(user,user_login)
    output("Anda telah logout")
else
    output('Logout gagal!')

```

```

        output('Anda belum login, silahkan login terlebih
        dahulu sebelum melakukan logout')
{REALISASI FUNGSI/PROSEDUR}
save_oc_user_login

```

## 5. F04 - Menu & Help

### FUNGSI - Help

```
{ Fitur ini berfungsi memunculkan panduan penggunaan sistem }
```

### KAMUS LOKAL

```

data_status_login : boolean
data_role : string

```

### ALGORITMA

```

data_role ← user_login[1][2]
data_username ← user_login[1][1]
data_status_login ← user_login[1][4]
depend on (data_status_login, data_role)

```

```

data_status_login = "False":
    output("===== HELP =====")
    output("Kamu belum login sebagai role apapun.
    Silahkan login terlebih dahulu.")
    output("1. login: Masuk ke dalam akun yang
    sudah terdaftar")
    output("2. register: Membuat akun baru")
    output("Footnote:")
    output("1. Untuk menggunakan aplikasi, silahkan
    masukkan nama fungsi yang terdaftar")
    output("2. Jangan lupa untuk memasukkan input
    yang valid")

```

```

data_role = "Agent":
    output(f"Halo Agent {data_username}. Kamu
    memanggil command HELP. Kamu memilih jalan yang
    benar, semoga kamu tidak sesat kemudian.")
    output("===== HELP =====")
    output("1. login - Untuk melakukan login ke
    dalam sistem")
    output("2. logout - Untuk melakukan logout pada
    sistem")
    output("3. inventory - Untuk melihat jumlah
    O.W.C.A Coin, monster, dan potion yang dimiliki")
    output("4. battle - Untuk bertarung melawan
    monster secara random")

```



```

        output("5. arena - Untuk melatih monster yang
dimiliki dan mendapatkan hadiah berupa O.W.C.A Coin")
        output("6. shop_currency - Untuk membeli
monster dan potion pada toko")
        output("7. laboratory - Untuk melakukan upgrade
monster yang dimiliki di inventory")
        output("8. save - Untuk melakukan penyimpanan
data")
        output("9. exit - Untuk keluar dari program")
        output("\\n")
        output("Footnote:")
        output('1. Untuk menggunakan aplikasi, silahkan
masukkan nama fungsi yang terdaftar')
        output('2. Jangan lupa untuk memasukkan input
yang valid')

data_role = "Admin":
        output('Selamat datang, Admin. Berikut adalah
hal-hal yang dapat kamu lakukan:')
        output("==== HELP =====")
        output("1. register - Untuk melakukan
registrasi user baru")
        output("2. login - Untuk melakukan login ke
dalam sistem")
        output("3. logout - Untuk melakukan logout pada
sistem")
        output("4. shop_management - Untuk melakukan
penambahan, perubahan, dan penghapusan Item yang
dijual dari Database, pengaturan harga dan jumlah
dari tiap item pada toko")
        output("5. monster_management - Untuk mengatur
dan menambahkan data monster dalam database")
        output("6. save - Untuk melakukan penyimpanan
data")
        output("7. exit - Untuk keluar dari program")
        output("\\n")
        output("Footnote:")
        output("1. Untuk menggunakan aplikasi, silahkan
masukkan nama fungsi yang terdaftar")
        output("2. Jangan lupa untuk memasukkan input
yang valid")

```

## 6. F05 - Monster

### Fungsi - Edit Atribut Monster

```

function edit_att_m(monster_inv: list, monster: list,
user_login: list) → dict
{I.S. : monster_inv, monster, user_login terdefinisi}
{F.S. : Mengembalikan dictionary (dict) yang berisi atribut
monster yang telah diedit}

```

#### KAMUS LOKAL

```

data_id: string
edit_att: dict
m, mo: array of string
lvl: integer

```

type dict

```

edit_att {
    "id": array of string,
    "type": array of string,
    "atk": array of integer,
    "def": array of integer,
    "hp": array of integer,
    "lvl": array of integer
}

```

#### ALGORITMA

```

data_id <- user_login[1][0]
edit_att <- {
    "id": [],
    "type": [],
    "atk": [],
    "def": [],
    "hp": [],
    "lvl": [],
}
i traversal [1..len(monster_inv)]
    if monster_inv[i-1][0] == data_id and
monster_inv[i-1][1] not in edit_att["id"] then
    edit_att["id"].append(monster_inv[i-1][1])
    edit_att["lvl"].append(int(monster_inv[i-1][2]))
    lvl <- int(monster_inv[i-1][2])
    j transversal [1..len(monster)]
        if monster[j-1][0] == monster_inv[i-1][1] then
            edit_att["type"].append(monster[j-1][1])
            edit_att["atk"].append(monst_att(lvl,
monster[j-1][2]))
            if monst_att(lvl, monster[j-1][3]) <= 50
then

```

```

                                edit_att["def"].append(monst_att(lvl,
monster[j-1][3]))
                                else
                                    edit_att["def"].append(50)
                                edit_att["hp"].append(monst_att(lvl,
monster[j-1][4]))
                                → edit_att

```

### **Fungsi - Serangan**

```

function atk(attacker: list, defender: list) → list
{I.S. : attacker dan defender terdefinisi}
{F.S. : Mengembalikan defender dengan HP yang telah dikurangi
oleh serangan attacker}

```

### **KAMUS LOKAL**

```

    rng_atk : integer
    atk_power : integer
    def_power : integer
    total_minus : integer
    hp_before : integer

```

### **ALGORITMA**

```

    rng_atk <- random_number([-30, 30])
    atk_power <- attacker[2] * (100 + rng_atk) / 100
    def_power <- random_number([0, defender[3]])
    total_minus <- atk_power * (100 - def_power) / 100
    hp_before <- defender[4]
    if math.floor(defender[4] - total_minus) > 0 then
        defender[4] <- math.floor(defender[4] - total_minus)
    else
        defender[4] <- 0
    print(clr.colored(f"{defender[1]}", 'red') + ' terkena
damage sebesar ' + clr.colored(f"{hp_before - defender[4]}",
'red'))
    → defender

```

### **Procedure - Tampilkan Monster**

```

procedure show_monster(player_monster: list, lvl: integer)
{I.S. : player_monster dan lvl terdefinisi}
{F.S. : Menampilkan informasi monster ke layar}

```

### **KAMUS LOKAL**

-

### **ALGORITMA**

```

    print(f"Name      : " + colored(f"{player_monster[1]}",
'red'))
    print(f"ATK Power : " + colored(f"{player_monster[2]}",
'red'))
    print(f"DEF Power : " + colored(f"{player_monster[3]}",
'red'))
    print(f"HP        : " + colored(f"{player_monster[4]}",
'red'))
    print(f"Level     : " + colored(f"{lvl}", 'red'))
    return

```

#### **Fungsi - Edit Atribut Monster Musuh**

function edit\_att\_r\_m(monster: list, level: integer) → list  
 {I.S. : monster dan level terdefinisi}  
 {F.S. : Mengembalikan monster dengan atribut yang telah diedit sesuai level}

#### **KAMUS LOKAL**

monster : list

#### **ALGORITMA**

```

monster[2] <- monst_att(level, monster[2])
monster[3] <- monst_att(level, monster[3])
if monster[3] > 50 then
    monster[3] <- 50
monster[4] <- monst_att(level, monster[4])
→ monster

```

#### **Fungsi - Hitung Atribut Monster**

function monst\_att(level: integer, nilai: string) → integer  
 {I.S. : level dan nilai terdefinisi}  
 {F.S. : Mengembalikan nilai atribut monster yang telah dihitung berdasarkan level}

#### **KAMUS LOKAL**

tidak ada

#### **ALGORITMA**

```

→ int(float(nilai) * (1 + (level - 1) / 10))

```

#### **Procedure - Seni Monster Pengguna**

procedure monster\_art\_user()  
 {I.S. : tidak ada}  
 {F.S. : Menampilkan seni ASCII monster pengguna}



```

array[0...M] of string) -> array[0...P] of array[0...Q] of
string
{ Fungsi untuk memuat data item pengguna berdasarkan item
inventory dan user login. }
{ I.S.: item_inv adalah daftar berisi item inventory. }
{ F.S.: Data item yang dimiliki pengguna dikembalikan
dalam bentuk daftar. }

```

#### KAMUS LOKAL

```

user_login : array[0...N] of array[0...M] of string
item_inv : array[0...N] of array[0...M] of string
data_id : string
data_p : array[0...P] of array[0...Q] of string
p : integer

```

#### ALGORITMA

```

user_login <- parseran.read_csv(user_login.csv)
item_inv <- parseran.read_csv(item_inventory.csv)
data_id <- user_login[1][0]
data_p : array[0...P] of array[0...Q] of string <-
array kosong
  p transversal [0, len(item_inv) - 1]
    if item_inv[p][0] = data_id then
      data_p.append([item_inv[p][1],
item_inv[p][2]])
    -> data_p

```

### 8. F07 - Inventory

{Fungsi ini memperlihatkan semua monster dan potion yang disimpan dari sebuah user}

#### KAMUS LOKAL

```

data_oc: integer
inventory: array[0...N] of array[0...M] of string

```

#### ALGORITMA

```

inventory <- {
  "user_id_inventory": []
  "type": []
  "quantity": []
  "user_id_monster": []
  "monster_id": []
  "monster_level": []
  "monsters": []
  "hp_monst": []
  "atk_monst": []
}

```

```

        "def_monst": []
    }

    u_id_invent <- inventory["user_id_inventory"]
    tipe_item <- inventory["type"]
    qty <- inventory["quantity"]
    u_id_monst <- inventory["user_id_monster"]
    monst_id <- inventory["monster_id"]
    monst_lvl <- inventory["monster_level"]
    monst <- inventory["monsters"]
    hp_monst <- inventory["hp_monst"]
    atk_monst <- inventory["atk_monst"]
    def_monst <- inventory["def_monst"]
    output("===== INVENTORY LIST (User ID:",
    user_login[1][0], "=====")
    output("Jumlah O.W.C.A Coin-mu sekarang", data_oc)
    x traversal[len(potion)]

    inventory["user_id_inventory"].append(potion[x][0])
        inventory["type"].append(potion[x][1])
        inventory["quantity"].append(potion[x][2])
    x traversal[len(monster_inventory)]

inventory["user_id_monster"].append(monster_inventory[x][0
])

    inventory["monster_id"].append(monster_inventory[x][1
])

    inventory["monster_level"].append(monster_inventory[x
][2])

    for x in monster
        if str(monster_inventory[x][1]) == str(m[0])
    then
        inventory["monsters"].append(m[1])
        inventory["hp_monst"].append(m[4])
        inventory["atk_monst"].append(m[2])
        inventory["def_monster"].append(m[3])
    output("=====ITEMS=====")
    save_id_p <- []
    x traversal [len(inventory["monsters"])]
        if u_id_monst[x+1] == user_login[1][0] then

```

```

        output(x, "Monster  ", monst[x],
"(Monster-ID: ", monst_id[x+1], ", Level:",
monst_lvl[x+1], ", HP:", hp_monst[x], ") ")
        save_id_p.append(x)
input(tampil)
while tampil != "keluar" do
    input(x)
    if tampil == "monster" then
        if x in save_id_m then
            output(x, "Monster  ", monst[x],
"(Monster-ID: ", monst_id[x+1], ", Level:",
monst_lvl[x+1], ", HP:", hp_monst[x], ") ")
        else
            output("Tidak ada data")
        elif tampil == "Potion" then
            if x in save_id_p then
                output(x, ". Potion      Type:",
tipe_item[x]", Quantity:", qty[x])
            else
                output("Tidak ada data")
        input(tampil)

```

## 9. F08 - Battle

```

function battle(user_login: list, monster: list, item_inv:
list, monster_inv: list) -> list
{I.S. : user_login, monster, item_inv, dan monster_inv
terdefinisi}
{F.S. : Mengembalikan user_login yang telah diperbarui}

```

### KAMUS LOKAL

```

id_monster: integer
monster_lvl: integer
random_monster: list
result: dict
reward: integer

```

### ALGORITMA

```

id_monster ← RNG.random_number([1, len(monster)])
monster_lvl ← RNG.random_number([1, 5])
random_monster ← []
for m in monster do
    if m[0] == str(id_monster) then
        random_monster.append(m[0])
        random_monster.append(m[1])

```



```

        random_monster.append(int(m[2]))
        random_monster.append(int(m[3]))
        random_monster.append(int(m[4]))
    random_monster.append(monster_lvl)
    result ← fight(monster_lvl, random_monster, user_login,
item_inv, monster_inv, monster)
    if result['win'] then
        reward ← RNG.random_number([5, 30])
        user_login[1][3] ← str(int(user_login[1][3]) +
reward)
        output(clr.colored(f"Selamat, Anda berhasil
mengalahkan monster {random_monster[1]}. Anda mendapatkan
{reward} OC.", 'yellow'))
        -> user_login
    else
        output(clr.colored(f"Yahhh, Anda dikalahkan monster
{random_monster[1]}. Jangan menyerah, coba lagi !!!",
'grey'))

```

## 10. F09 - Arena

```

function show_arena_stat(stat: dict) → None
{I.S. : stat berisi informasi statistik arena}
{F.S. : Statistik arena ditampilkan ke layar}

```

### KAMUS LOKAL

#### ALGORITMA

```

output(25*"=" + "STAT" + 25*"=")
output(f"""
Jumlah Hadiah           : {stat['oc']}
Jumlah Stage            : {stat['stage']}
Total Damage Diberikan  : {stat['total_damage']}
Total Damage Diterima   : {stat['damage_taken']}
""")

```

```

function arena(user_login: list, data_monster: list, item_inv:
list, monster_inv: list) → list
{I.S. : user_login, data_monster, item_inv, dan monster_inv
terdefinisi}
{F.S. : Mengembalikan user_login yang telah diperbarui}

```

### KAMUS LOKAL

```

status_login: string
id_monster: list of integer
stat: dict

```

```

monster: list
monster_level: integer
result: dict
reward: integer

```

#### ALGORITMA

```

status_login <- user_login[1][4]
if status_login == "False" then
  → Help.help()
else
  → id_monster <- RNG.random_number_arr([1,
len(data_monster)], 5)
  → stat <- {
    'total_damage': 0,
    'damage_taken': 0,
    'oc': 0,
    'stage': 0,
  }

  i transversal [0..5] do
    for m in data_monster do
      if m[0] == str(id_monster[i]) then
        → monster <- m
        → monster_level <- i + 1
        → monster <-
Monster_edt.edit_att_r_m(monster, monster_level)
        → result <- Battle.fight(monster_level,
monster, user_login, item_inv, monster_inv, data_monster)
        → stat['damage_taken'] +=
result['damage_taken']
        → stat['total_damage'] +=
result['total_damage']

        if result['win'] then
          → reward <- int(30 * (i + 1) ** 1.25)
          → user_login[1][3] <- str(int(user_login[1][3])
+ reward)

          → stat["stage"] += 1
          → stat["oc"] += reward
          output(f"Selamat, Anda berhasil mengalahkan
monster {monster[1]}. \nSTAGE CLEARED! Anda mendapatkan
{reward} OC pada stage ini.")
          time.sleep(0.75)
          if not (stat['stage'] == 5) then
            output("Memulai stage berikutnya...")

```

```

        else
            output(f"Yahhh, Anda dikalahkan monster
{monster[1]}. Jangan menyerah, coba lagi !!!")
            break

    if stat['stage'] == 5 then
        output("Selamat, Anda berhasil menyelesaikan
seluruh stage!")

    → show_arena_stat(stat)
    → user_login

```

## 11. F10 - Shop & Currency (Potion/ Monster)

### KAMUS LOKAL

```

    pilihan : string
    lihat_apa: string
    beli_apa: string

```

### ALGORITMA

```

    output("Irasshaimase! Selamat datang di SHOP!!")
    input(pilihan)
    while pilihan != "keluar" do
        if pilihan == "lihat" then
            input(lihat_apa)
            if lihat_apa == "monster" then
                lihat_monster(monster_shop, monster)
            elif lihat_apa == "potion" then
                lihat_potion(item_shop)
        elif pilihan == "beli" then
            output("Jumlah O.W.C.A koin mu:",
user_login[1][3])
            input(beli_apa)
            if beli_apa == "monster" then
                input(id_monst)
                (monster_inventory, user_login) <-
beli_monster(id_monst, monster_shop,
monster_inventory, user_login, monster)
            elif beli_apa == "potion" then
                input(id_pot)
                input(qty)
                (item_inventory, user_login) <-
beli_potion(id_pot, qty, item_inventory, item_shop,
user_login)
            input(pilihan)

```

```

        output("Mr. Yanto bilang makasih, belanja
lagi ya nanti :)")

```

## 12. F11 - Laboratory

### Fungsi Pembantu - Load Data Monster

```

function load_data_m(user_login: list, monster: list,
monster_inv: list) → dict
{I.S. : user_login, monster, dan monster_inv terdefinisi}
{F.S. : Mengembalikan data monster yang dimiliki pengguna}

```

#### KAMUS LOKAL

```

data_id: string
data_m: dict
i: integer

```

#### ALGORITMA

```

data_id <- user_login[1][0]
data_m <- {
    "id": [],
    "type": [],
    "lvl": [],
}

i traversal [0, len(monster_inv) - 1] do
    if monster_inv[i][0] == data_id then
        → data_m["id"].append(monster_inv[i][1])
        → data_m["lvl"].append(monster_inv[i][2])

        j traversal [0, len(monster) - 1] do
            if monster[j][0] == monster_inv[i][1] then
                → data_m["type"].append(monster[j][1])
        → data_m

```

### Fungsi Pembantu - Lihat Monster

```

procedure lihat(data_m: dict) → None
{I.S. : data_m terdefinisi}
{F.S. : Menampilkan daftar monster dan harga upgrade ke layar}

```

#### KAMUS LOKAL

```

i: integer

```

#### ALGORITMA

```

output('Selamat datang di Lab Dokter Asep !!!')

```

```

    output('===== MONSTER LIST =====')
    i traversal [0, len(data_m["id"]) - 1] do
        → output(f'{data_m["id"][i]}. {data_m["type"][i]}
(Level: {data_m["lvl"][i]})')

```

```

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

```

### Fungsi Pembantu - Upgrade Monster

```

function upgrade_m(lvl_now: int, pilih: int, data_m: dict,
monster_inv: list, user_login: list) → (list, list)
{I.S. : lvl_now, pilih, data_m, monster_inv, dan user_login
terdefinisi}
{F.S. : Mengembalikan user_login dan monster_inv yang
diperbarui setelah upgrade}

```

### KAMUS LOKAL

```

harga: list of integer
data_oc: string
data_id: string
i: integer

```

### ALGORITMA

```

harga <- [300, 500, 800, 1000]
data_oc <- user_login[1][3]
data_id <- user_login[1][0]
if int(data_oc) < harga[lvl_now + 1 - 2] then
    → output('Maaf, koin anda tidak cukup')
    → (user_login, monster_inv)

i traversal [0, len(monster_inv) - 1] do
    if monster_inv[i][0] == data_id and monster_inv[i][1]
== data_m["id"][pilih - 1] then
        → monster_inv[i][2] <- str(int(monster_inv[i][2]) +
1)

    data_oc <- str(int(data_oc) - harga[lvl_now + 1 - 2])
    output(f'Selamat, {data_m["type"][pilih - 1]} berhasil
diupgrade ke level {lvl_now + 1}')
    user_login[1][3] <- str(data_oc)
    → (user_login, monster_inv)

```

### Fungsi Utama - Laboratory

```
procedure laboratory(user_login: list, monster: list,  
monster_inventory: list) → None  
{I.S. : user_login, monster, dan monster_inventory terdefinisi}  
{F.S. : Menampilkan menu laboratory dan memproses input  
pengguna}
```

### KAMUS LOKAL

```
data_m: dict  
harga: list of integer  
pilih: string  
lvl_now: int  
lvl_tlh_up: int  
yakini_up: string
```

### ALGORITMA

```
data_m <- load_data_m(user_login, monster,  
monster_inventory)  
lihat(data_m)  
harga <- [300, 500, 800, 1000]  
  
while pilih != 'keluar' do  
    pilih <- input('>>> Pilih monster: ')  
    i <- int(pilih) - 1  
    if pilih in data_m["id"] then  
        if int(data_m["lvl"][i]) < 5 then  
            lvl_now <- int(data_m["lvl"][i])  
            lvl_tlh_up <- lvl_now + 1  
            output(f'{data_m["type"][i]} akan di-upgrade ke  
level {lvl_tlh_up}')  
            output(f'Harga untuk melakukan upgrade  
{harga[lvl_tlh_up - 2]}')  
            yakini_up <- input('>>> Lanjutkan upgrade (Y/N):  
' )  
            if yakini_up == 'Y' then  
                (user_login, monster_inventory) <-  
upgrade_m(lvl_now, i + 1, data_m, monster_inventory,  
user_login)  
                data_m <- load_data_m(user_login, monster,  
monster_inventory)  
                lvl_now <- int(data_m["lvl"][i - 1])  
                lvl_tlh_up <- lvl_now + 1  
            else  
                output(f'Woke, anda tidak jadi upgrade  
{data_m["type"][i - 1]}')
```

```

        else
            output('Maaf, monster yang Anda pilih sudah
memiliki level maksimum')
        else
            output('Monster Id Tidak ada di Inventory')
        pilih <- input('>>> Pilih monster: ')

```

### 13. F12 - Shop Management

```

function shop_management(monster_shop: list, item_shop: list,
monster: list) -> None
{I.S. : monster_shop, item_shop, monster, dan fungsi
Shop_Currency.lihat_monster, tambah_m, ubah_m, hapus_m,
lihat_potion, tambah_p, ubah_p, dan hapus_p terdefinisi}
{F.S. : Tidak ada output.}

```

#### KAMUS LOKAL

```

    pilihan: string
    lihat_apa: string
    tambah_apa: string
    ubah_apa: string
    hapus_apa: string

```

#### ALGORITMA

```

    output("Trasshaimase! Selamat datang kembali, Mr.
Monogram!")
    pilihan <- input(">>> Pilih aksi
(lihat/tambah/ubah/hapus/keluar): ")
    while pilihan != 'keluar' do
        if pilihan == 'lihat' then
            lihat_apa <- input(">>> Mau lihat apa?
(monster/potion): ")
            if lihat_apa == 'monster' then
                Shop_Currency.lihat_monster(monster_shop,
monster)
            elif lihat_apa == 'potion' then
                Shop_Currency.lihat_potion(item_shop)
            elif pilihan == 'tambah' then
                tambah_apa <- input(">>> Mau nambahin apa?
(monster/potion): ")
                if tambah_apa == 'monster' then
                    monster_shop <- tambah_m(monster, monster_shop)
                elif tambah_apa == 'potion' then
                    item_shop <- tambah_p(item_shop)
            elif pilihan == 'ubah' then

```

```

        ubah_apa ← input(">>> Mau ubah apa?
(monster/potion): ")
        if ubah_apa == 'monster' then
            monster_shop ← ubah_m(monster_shop, monster)
        elif ubah_apa == 'potion' then
            item_shop ← ubah_p(item_shop)
        elif pilihan == 'hapus' then
            hapus_apa ← input(">>> Mau hapus apa?
(monster/potion): ")
            if hapus_apa == 'monster' then
                monster_shop ← hapus_m(monster_shop, monster)
            elif hapus_apa == 'potion' then
                item_shop ← hapus_p(item_shop)
            pilihan ← input(">>> Pilih aksi
(lihat/tambah/ubah/hapus/keluar): ")
            output("Dadah Mr. Yanto, sampai jumpa lagi!")

```

#### 14. F13 - Monster Management

##### Fungsi pembantu - Check Monster

```

function check_m(monster: list, type_m: str) → bool
{I.S. : monster dan type_m terdefinisi}
{F.S. : Mengembalikan True jika type_m ditemukan dalam monster,
False jika tidak}

```

##### KAMUS LOKAL

found: bool

##### ALGORITMA

```

found ← False
for m in monster do
    if m[1] == type_m then
        → found ← True
→ found

```

##### Fungsi pembantu - Lihat Monster

```

procedure lihat_m(monster: list) → None
{I.S. : monster terdefinisi}
{F.S. : Menampilkan daftar monster ke layar}

```

##### KAMUS LOKAL

spasi\_atk: integer  
 spasi\_def: integer

##### ALGORITMA

```

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

```



```

for m in monster do
  if m[0] != "id" then
    → spasi_atk <- len(' ATK Power ') - len(m[2]) - 1
    → spasi_def <- len(' DEF Power ') - len(m[3]) - 1
    if len(str(m[0])) > 1 then
      → output(f'{m[0]} | {m[1]} | {m[2]}' +
spasi_atk*" " + f'| {m[3]}' + spasi_def*" " + f'| {m[4]} ')
    else
      → output(f'{m[0]} | {m[1]} | {m[2]}' +
spasi_atk*" " + f'| {m[3]}' + spasi_def*" " + f'| {m[4]} ')

```

### Fungsi pembantu - Buat Monster

```

function buat_m(monster: list) → list
{I.S. : monster terdefinisi}
{F.S. : Mengembalikan daftar monster yang diperbarui dengan
monster baru jika ditambahkan}

```

### KAMUS LOKAL

```

type_m: string
found: bool
atk_m: string
def_m: string
hp_m: string
yakini_tambah: string

```

### ALGORITMA

```

output('Memulai pembuatan monster baru')
type_m <- input('>>> Masukkan type / Nama : ')
found <- check_m(monster, type_m)
while found == True do
  output('Nama sudah terdaftar, coba lagi!')
  found <- False
  type_m <- input('>>> Masukkan type / Nama : ')
  found <- check_m(monster, type_m)

atk_m <- input('>>> Masukkan ATK Power (1-1000): ')
while not(1 <= int(atk_m) <= 1000) do
  output('Masukan anda tidak valid, coba lagi!')
  atk_m <- input('>>> Masukkan ATK Power (1-1000): ')

def_m <- input('>>> Masukkan DEF Power (0-50): ')
while not(0 <= int(def_m) <= 50) do
  output('DEF Power harus bernilai 0-50, coba lagi!')
  def_m <- input('>>> Masukkan DEF Power (0-50): ')

```

```

hp_m <- input('>>> Masukkan HP (0-500): ')
while not(0 <= int(hp_m) <= 500) do
    output('HP harus bernilai 0-500, coba lagi!')
    hp_m <- input('>>> Masukkan HP (0-500): ')

output('Monster baru berhasil dibuat')
output(f'Type : {type_m}')
output(f'ATK Power : {atk_m}')
output(f'DEF Power : {def_m}')
output(f'HP : {hp_m}')

yakin_tambah <- input('>>> Tambahkan Monster ke database
(Y/N) : ')
if yakin_tambah == 'Y' then
    output('Monster baru telah ditambahkan!')
    → monster.append([len(monster), type_m, atk_m, def_m,
hp_m])
    → monster
else
    output('Monster gagal ditambahkan!')

```

### Fungsi utama - Monster Management

```

procedure monster_management(monster: list) → None
{I.S. : monster terdefinisi}
{F.S. : Menampilkan menu manajemen monster dan memproses input
pengguna}

```

### KAMUS LOKAL

pilih: string

### ALGORITMA

```

output('SELAMAT DATANG DI DATABASE PARA MONSTER !!!')
output('1. Tampilkan semua Monster')
output('2. Tambah Monster baru')
pilih <- input('>>> Pilih Aksi (1/2/keluar): ')

while pilih != 'keluar' do
    if pilih == '1' then
        → lihat_m(monster)
    elif pilih == '2' then
        → monster <- buat_m(monster)
    pilih <- input('>>> Pilih Aksi (1/2/keluar): ')

```

## 15. F14 - Load

```
Fungsi read_csv_folder(folder: string, user:list,
user_login:list, monster:list, monster_shop:list,
monster_inventory:list, item_shop:list,
item_inventory:list) -> (array[0...N] of array[0...M] of
string, array[0...N] of array[0...M] of string,
array[0...N] of array[0...M] of string, array[0...N] of
array[0...M] of string, array[0...N] of array[0...M] of
string, array[0...N] of array[0...M] of string)
{Fungsi untuk membaca file CSV dari folder yang diberikan.
}
{I.S.: folder adalah nama folder yang berisi file CSV. }
{ F.S.: Mengembalikan daftar data dari file CSV user,
monster, monster_shop, monster_inventory, item_shop, dan
item_inventory. }
```

### KAMUS LOKAL

```
folder : string
user : array[0...N] of array[0...M] of string
monster : array[0...N] of array[0...M] of string
monster_shop : array[0...N] of array[0...M] of string
monster_inventory : array[0...N] of array[0...M] of
string
item_shop : array[0...N] of array[0...M] of string
item_inventory : array[0...N] of array[0...M] of
string
```

### ALGORITMA

```
os.chdir(folder)
user : array[0...N] of array[0...M] of string <-
read_csv(user.csv)
monster : array[0...N] of array[0...M] of string <-
read_csv(monster.csv)
monster_shop : array[0...N] of array[0...M] of string
<- read_csv(monster_shop.csv)
monster_inventory : array[0...N] of array[0...M] of
string <- read_csv(monster_inventory.csv)
item_shop : array[0...N] of array[0...M] of string <-
read_csv(item_shop.csv)
item_inventory : array[0...N] of array[0...M] of
string <- read_csv(item_inventory.csv)
os.chdir(../)
-> (user, monster, monster_shop, monster_inventory,
item_shop, item_inventory)
```

### **Prosedur loading()**

```
{ Prosedur untuk memuat data dari folder CSV yang
diberikan sebagai argumen. }
{ I.S.: Tidak ada nama folder yang diberikan, fungsi
read_csv_folder sudah terdefinisi }
{ F.S.: Memuat data dari file CSV di folder yang diberikan
atau menampilkan pesan kesalahan jika folder atau file
tidak ditemukan. }
```

### **KAMUS LOKAL**

```
parser : argparse.ArgumentParser
args : argparse.Namespace
folder_name : string
directory : string
parent : string
path : string
list_csv : array[0...5] of string
csv : integer
```

### **ALGORITMA**

```
parser <-
argparse.ArgumentParser(description="Menjalankan main.py
dalam folder yang spesifik di dalam folder parent 'data'")

parser.add_argument("folder_name", nargs='?', help="Nama
folder di dalam parent 'data'")
args <- parser.parse_args()

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

folder_path <- os.path.join(os.getcwd(),"data",
args.folder_name)

if not os.path.exists(folder_path) then
    output(Nama folder yang diinputkan tidak ada)
else
    os.chdir(./ + directory)
    list_csv : array[0...5] of string <- [user.csv,
monster.csv, monster_shop.csv, monster_inventory.csv,
item_shop.csv, item_inventory.csv]
    csv transversal [0, len(list_csv) - 1]
```

```

        if not os.path.exists(list_csv[csv]) then
            output(list_csv[csv] + tidak ditemukan)
            -> None
    os.chdir(../)
    output(tes)
    -> directory

```

## 16. F15 - Save

```

Prosedur save(folder_name: string)
{ Prosedur untuk menyimpan data ke folder yang diberikan.
}
{ I.S.: folder_name adalah nama folder tempat data akan
disimpan. }
{ F.S.: Data disimpan di folder yang sesuai dengan nama
folder_name. }

```

### ALGORITMA

```

write_csv_to_folder(folder_name, user)
write_csv_to_folder(folder_name, monster)
write_csv_to_folder(folder_name, monster_shop)
write_csv_to_folder(folder_name, monster_inventory)
write_csv_to_folder(folder_name, item_shop)
write_csv_to_folder(folder_name, item_inventory)

```

### KAMUS LOKAL

```

folder_name : string

```

```

Prosedur new_save()
{ Prosedur untuk membuat folder baru dan menyimpan data di
dalamnya. }
{ I.S.: Tidak ada folder yang dibuat atau data yang
disimpan. }
{ F.S.: Membuat folder baru dan menyimpan data di
dalamnya. }

```

### ALGORITMA

```

folder_name : string <- input(Masukkan nama folder: )

if os.path.exists(./data) then
    if not os.path.exists(./data/ + folder_name) then
        output(Membuat folder data/ + folder_name +
        ...)
        os.mkdir(./data/ + folder_name)
        wait(0.75)

```

```

else
    output(Membuat folder data...)
    wait(0.75)
    os.mkdir('./data')
    output(Membuat folder data/ + folder_name + ...)
    wait(1)
    os.mkdir('./data/ + folder_name)

    save(folder_name)
    output(Berhasil menyimpan data di folder data/ +
folder_name + !)

```

#### KAMUS LOKAL

folder\_name : string

### 17. F16 - Exit

Prosedur save\_oc\_user\_login(user: array[1..n] of array[1..m] of string, user\_login: array[1..k] of array[1..l] of string)  
{ Prosedur untuk menyimpan OC user login ke dalam database. }  
{ I.S.: user adalah daftar pengguna, user\_login adalah data login pengguna. }  
{ F.S.: Data OC pengguna disimpan di database. }

#### KAMUS LOKAL

u : array[1..m] of string

#### ALGORITMA

```

u transversal [1, len(user) - 1]
    if u[0] = user_login[1][0] then
        u[4] = user_login[1][3]
    save_data(user.csv, user)
    output(Data OC anda sudah tersimpan di database)

```

Prosedur Exit()  
{ Prosedur untuk mengonfirmasi dan menyimpan perubahan sebelum keluar dari program. }  
{ I.S.: Tidak ada perubahan yang disimpan atau konfirmasi dari pengguna. }  
{ F.S.: Menyimpan perubahan (jika diinginkan) dan keluar dari program. }

#### KAMUS LOKAL

user\_login : array[1..k] of array[1..l] of string

```
user : array[1..n] of array[1..m] of string  
konfirmasi : string
```

#### **ALGORITMA**

```
    konfirmasi = input(Apakah Anda mau melakukan  
penyimpanan file yang sudah diubah? (y/n) )  
    while not ((konfirmasi = y) or (konfirmasi = Y) or  
(konfirmasi = n) or (konfirmasi = N)) do  
        konfirmasi = input(Apakah Anda mau melakukan  
penyimpanan file yang sudah diubah? (y/n) )  
        if (konfirmasi = y) or (konfirmasi = Y) then  
            save_oc_user_login(user, user_login)  
            Save.new_save()  
            exit()  
        else  
            exit()
```

## PENGUJIAN PROGRAM

```
PS C:\Users\DELL\Documents\1. FILE ITB\TUBES DASPRO\if1210-2024-tubes-k05-b\src> python 1.main.py cobab
```

Gambar 1. Memulai Program

```
Selamat datang di program OWCA!  
  
Masukkan perintah: (ketik 'help' untuk melihat semua perintah)  
>>> help
```

Gambar 2. Tampilan Awal

```
>>> help  
===== HELP =====  
Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu.  
1. login: Masuk ke dalam akun yang sudah terdaftar  
2. register: Membuat akun baru  
Footnote:  
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar  
2. Jangan lupa untuk memasukkan input yang valid
```

Gambar 3. Tampilan Help

```
>>> register  
Masukan username: soni  
Masukan password: 123  
Silahkan pilih salah satu monster sebagai monster awalmu.  
1. Chari  
2. Bulba  
3. Aspal  
4. Gatot  
5. Asepe  
6. Gajah  
7. Cobra  
8. Singa  
9. Udin_  
Silahkan pilih salah satu monster sebagai monster awalmu:  
1  
Selamat datang Agent soni. Mari kita mengalahkan Dr. Asep Spakbor dengan Chari!
```

Gambar 4. Tampilan Register



```
>>> login
Username : soni
Password : 123
Selamat datang, Agent soni!
Masukkan command 'help' untuk daftar command yang dapat kamu panggil.
```

**Gambar 5.** Tampilan Login

```
>>> help
Halo Agent soni. Kamu memanggil command HELP. Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian.
===== HELP =====
1. login - Untuk melakukan login ke dalam sistem
2. logout - Untuk melakukan logout pada sistem
3. inventory - Untuk melihat jumlah O.W.C.A Coin, monster, dan potion yang dimiliki
4. battle - Untuk bertarung melawan monster secara random
5. arena - Untuk melatih monster yang dimiliki dan mendapatkan hadiah berupa O.W.C.A Coin
6. shop_currency - Untuk membeli monster dan potion pada toko
7. laboratory - Untuk melakukan upgrade monster yang dimiliki di inventory
8. save - Untuk melakukan penyimpanan data
9. exit - Untuk keluar dari program
10. jackpot - untuk bermain jackpot dan mendapatkan koin/monster

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

**Gambar 6.** Tampilan Help(setelah Login)

```
>>> inventory
===== INVENTORY LIST (User ID: 7) =====
Jumlah O.W.C.A. Coin-mu sekarang 0
=====ITEMS=====
=====MONSTERS=====
13. Monster    Chari (Monster-ID: 1, Level: 1, HP: 250)
Ketikkan jenis item dan id untuk menampilkan detail item (monster/potion):
>>> monster
Pilih monster urutan berapa:
>>> 13
13. Monster    Chari (Monster-ID: 1, Level: 1, Atk Power: 100, Def Power: 40, HP: 250)
Ketikkan jenis item dan id untuk menampilkan detail item (monster/potion):
>>> █
```

**Gambar 7.** Tampilan Inventory



```

Bulba terkena damage sebesar 79
Name      : Bulba
ATK Power : 70
DEF Power : 40
HP        : 121
Level     : 1

=====TURN 2 (Bulba (musuh))=====
SCHWINKKK, Bulba (musuh) menyerang Chari !!!
Chari terkena damage sebesar 55
Name      : Chari
ATK Power : 100
DEF Power : 40
HP        : 195
Level     : 1

=====TURN 3 (Chari)=====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (angka): 2
Kamu tidak memiliki potion. Beli potion di shop.

=====TURN 4 (Bulba (musuh))=====
SCHWINKKK, Bulba (musuh) menyerang Chari !!!
Chari terkena damage sebesar 57
Name      : Chari
ATK Power : 100
DEF Power : 40
HP        : 138
Level     : 1

=====TURN 5 (Chari)=====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (angka): 1
SCHWINKKK, Chari menyerang Bulba(musuh) !!!

Bulba terkena damage sebesar 76
Name      : Bulba
ATK Power : 70
DEF Power : 40
HP        : 45
Level     : 1

=====TURN 6 (Bulba (musuh))=====
SCHWINKKK, Bulba (musuh) menyerang Chari !!!
Chari terkena damage sebesar 54
Name      : Chari
ATK Power : 100
DEF Power : 40
HP        : 84
Level     : 1

```

**Gambar 8(b).** Tampilan Battle

```
=====TURN 7 (Chari)=====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (angka): 1
SCHWINKKK, Chari menyerang Bulba(musuh) !!!

Bulba terkena damage sebesar 45
Name      : Bulba
ATK Power : 70
DEF Power : 40
HP        : 0
Level     : 1
Selamat, Anda berhasil mengalahkan monster Bulba. Anda mendapatkan 27 OC.
```

**Gambar 8(c).** Tampilan Battle



```

Gajah terkena damage sebesar 76
Name      : Gajah
ATK Power : 90
DEF Power : 45
HP        : 84
Level     : 1

=====TURN 2 (Gajah (musuh))=====
SCHWINKKK, Gajah (musuh) menyerang Chari !!!
Chari terkena damage sebesar 73
Name      : Chari
ATK Power : 100
DEF Power : 40
HP        : 177
Level     : 1

=====TURN 3 (Chari)=====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (angka): 1
SCHWINKKK, Chari menyerang Gajah(musuh) !!!

Gajah terkena damage sebesar 76
Name      : Gajah
ATK Power : 90
DEF Power : 45
HP        : 8
Level     : 1

=====TURN 4 (Gajah (musuh))=====
SCHWINKKK, Gajah (musuh) menyerang Chari !!!
Chari terkena damage sebesar 72
Name      : Chari
ATK Power : 100
DEF Power : 40
HP        : 105
Level     : 1

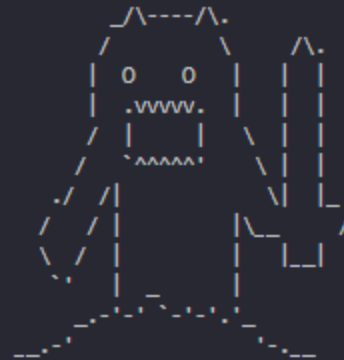
=====TURN 5 (Chari)=====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (angka): 1
SCHWINKKK, Chari menyerang Gajah(musuh) !!!

Gajah terkena damage sebesar 8
Name      : Gajah
ATK Power : 90
DEF Power : 45
HP        : 0
Level     : 1
Selamat, Anda berhasil mengalahkan monster Gajah.
STAGE CLEARED! Anda mendapatkan 30 OC pada stage ini.
Memulai stage berikutnya...

```

**Gambar 9(b).** Tampilan Arena

RAWR, Monster Aspal telah muncul !!!



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

1. Chari

Pilih monster untuk bertarung (angka): 1



Agent soni mengeluarkan monster Chari !!!

Name : Chari  
ATK Power : 100  
DEF Power : 40  
HP : 250  
Level : 1

=====TURN 1 (Chari)=====

1. Attack  
2. Use Potion  
3. Quit

Pilih perintah (angka): 1

SCHWINKKK, Chari menyerang Aspal(musuh) !!!

Aspal terkena damage sebesar 84

Name : Aspal  
ATK Power : 145  
DEF Power : 36  
HP : 85  
Level : 2



Gambar 9(c). Tampilan Arena

```
=====TURN 2 (Aspal (musuh))=====
SCHWINKKK, Aspal (musuh) menyerang Chari !!!
Chari terkena damage sebesar 117
Name      : Chari
ATK Power : 100
DEF Power : 40
HP        : 133
Level     : 1

=====TURN 3 (Chari)=====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (angka): 2
Kamu tidak memiliki potion. Beli potion di shop.

=====TURN 4 (Aspal (musuh))=====
SCHWINKKK, Aspal (musuh) menyerang Chari !!!
Chari terkena damage sebesar 118
Name      : Chari
ATK Power : 100
DEF Power : 40
HP        : 15
Level     : 1

=====TURN 5 (Chari)=====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (angka): 1
SCHWINKKK, Chari menyerang Aspal(musuh) !!!

Aspal terkena damage sebesar 84
Name      : Aspal
ATK Power : 145
DEF Power : 36
HP        : 1
Level     : 2

=====TURN 6 (Aspal (musuh))=====
SCHWINKKK, Aspal (musuh) menyerang Chari !!!
Chari terkena damage sebesar 15
Name      : Chari
ATK Power : 100
DEF Power : 40
HP        : 0
Level     : 1
Yahhh, Anda dikalahkan monster Aspal. Jangan menyerah, coba lagi !!!
=====STAT=====

Jumlah Hadiah      : 30
Jumlah Stage       : 1
Total Damage Diberikan : 328
Total Damage Diterima  : 395
```

Gambar 9(d). Tampilan Arena

```
Selamat datang di program OWCA!

Masukkan perintah: (ketik 'help' untuk melihat semua perintah)
>>> shop_currency
Irasshaimase! Selamat datang di SHOP!!
>>> Pilih aksi (lihat/beli/keluar): lihat
>>> Mau lihat apa? (monster/potion): monster
ID | Type           | ATK Power | DEF Power | HP   | Stok | Harga
1  | Chari            | 100       | 40        | 250  | 16   | 400
2  | Bulba            | 70        | 40        | 200  | 4    | 700
3  | Aspal            | 120       | 30        | 140  | 1    | 1000
4  | Gatot            | 90        | 50        | 130  | 0    | 550
5  | Asepe            | 100       | 40        | 120  | 5    | 600
6  | Gajah            | 90        | 45        | 160  | 2    | 500
7  | Cobra            | 95        | 50        | 180  | 20   | 450
>>> Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A koin mu: 1000
>>> Mau beli apa? (monster/potion): monster
>>> Masukkan id monster: 5
Berhasil membeli monster: Asepe. Monster sudah masuk ke inventory-mu
>>> Pilih aksi (lihat/beli/keluar): lihat
>>> Mau lihat apa? (monster/potion): potion
ID | Type           | Stok | Harga
1  | strength        | 2    | 20
2  | resilience       | 10   | 5
3  | healing          | 10   | 5
>>> Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A koin mu: 400
>>> Mau beli apa? (monster/potion): potion
>>> Masukkan id potion: 2
>>> Masukkan jumlah: 2
Berhasil membeli item: 2 resilience. Item sudah masuk ke inventory-mu
>>> Pilih aksi (lihat/beli/keluar): lihat
>>> Mau lihat apa? (monster/potion): potion
ID | Type           | Stok | Harga
1  | strength        | 2    | 20
2  | resilience       | 8    | 5
3  | healing          | 10   | 5
>>> Pilih aksi (lihat/beli/keluar): lihat
>>> Mau lihat apa? (monster/potion): monster
ID | Type           | ATK Power | DEF Power | HP   | Stok | Harga
1  | Chari            | 100       | 40        | 250  | 16   | 400
2  | Bulba            | 70        | 40        | 200  | 4    | 700
3  | Aspal            | 120       | 30        | 140  | 1    | 1000
4  | Gatot            | 90        | 50        | 130  | 0    | 550
5  | Asepe            | 100       | 40        | 120  | 4    | 600
6  | Gajah            | 90        | 45        | 160  | 2    | 500
7  | Cobra            | 95        | 50        | 180  | 20   | 450
>>> Pilih aksi (lihat/beli/keluar):
```

**Gambar 10.** Tampilan Shop\_currency

```

>>> shop_management
Irasshaimase! Selamat datang kembali, Mr. Monogram!
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Mau lihat apa? (monster/potion): monster
ID | Type          | ATK Power | DEF Power | HP  | Stok | Harga
1  | Chari          | 100       | 40        | 250 | 16   | 400
2  | Bulba          | 70        | 40        | 200 | 4    | 700
3  | Aspal          | 120       | 30        | 140 | 1    | 1000
4  | Gatot          | 90        | 50        | 130 | 0    | 550
5  | Asepe          | 100       | 40        | 120 | 5    | 600
6  | Gajah          | 90        | 45        | 160 | 2    | 500
7  | Cobra          | 95        | 50        | 180 | 20   | 450
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Mau lihat apa? (monster/potion): potion
ID | Type          | Stok | Harga
1  | strength      | 2    | 20
2  | resilience    | 10   | 5
3  | healing       | 10   | 5
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
>>> Mau nambahin apa? (monster/potion): monster
ID | Type          | ATK Power | DEF Power | HP
8  | Singa         | 100       | 40        | 140
9  | Udin_         | 110       | 40        | 200
>>> Masukkan id monster: 8
>>> Masukkan stok awal: 10
>>> Masukkan harga: 550
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
>>> Mau nambahin apa? (monster/potion): potion
['strength', 'resilience', 'healing']
ID | Type |
1  | magic |
2  | power |
>>> Masukkan type potion: magic
>>> Masukkan stok awal: 5
>>> Masukkan harga: 10
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): ubah
>>> Mau ubah apa? (monster/potion): monster
ID | Type          | ATK Power | DEF Power | HP  | Stok | Harga
1  | Chari          | 100       | 40        | 250 | 16   | 400
2  | Bulba          | 70        | 40        | 200 | 4    | 700
3  | Aspal          | 120       | 30        | 140 | 1    | 1000
4  | Gatot          | 90        | 50        | 130 | 0    | 550
5  | Asepe          | 100       | 40        | 120 | 5    | 600
6  | Gajah          | 90        | 45        | 160 | 2    | 500
7  | Cobra          | 95        | 50        | 180 | 20   | 450
8  | Singa          | 100       | 40        | 140 | 10   | 550
>>> Masukkan id monster: 7
>>> Masukkan stok baru: 15
>>> Masukkan harga baru: 600
Cobra telah berhasil diubah dengan stok baru sejumlah 15 dan dengan harga baru 600!

```

**Gambar 11(a).** Tampilan Shop\_management

```

>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): ubah
>>> Mau ubah apa? (monster/potion): potion
ID | Type           | Stok | Harga
1  | strength        | 2    | 20
2  | resilience       | 10   | 5
3  | healing          | 10   | 5
4  | magic           | 5    | 10
>>> Masukkan id potion: 3
>>> Masukkan stok baru: 8
>>> Masukkan harga baru: 15
healing telah berhasil diubah dengan stok baru sejumlah 8 dan dengan harga baru 15!
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): hapus
>>> Mau hapus apa? (monster/potion): monster
ID | Type           | ATK Power | DEF Power | HP | Stok | Harga
1  | Chari          | 100       | 40        | 250 | 16  | 400
2  | Bulba         | 70        | 40        | 200 | 4   | 700
3  | Aspal         | 120       | 30        | 140 | 1   | 1000
4  | Gatot         | 90        | 50        | 130 | 0   | 550
5  | Asepe         | 100       | 40        | 120 | 5   | 600
6  | Gajah         | 90        | 45        | 160 | 2   | 500
7  | Cobra         | 95        | 50        | 180 | 15  | 600
8  | Singa         | 100       | 40        | 140 | 10  | 550
>>> Masukkan id monster: 8
>>> Apakah anda yakin ingin menghapus Singa dari shop (y/n)? y
Yey, anda berhasil menghapus monste Singa
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): hapus
>>> Mau hapus apa? (monster/potion): potion
ID | Type           | Stok | Harga
1  | strength        | 2    | 20
2  | resilience       | 10   | 5
3  | healing          | 8    | 15
4  | magic           | 5    | 10
>>> Masukkan type potion: magic
>>> Apakah anda yakin ingin menghapus magic Potion dari shop (y/n)? y
Yey, anda berhasil menghapus potion magic
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Mau lihat apa? (monster/potion): monster
ID | Type           | ATK Power | DEF Power | HP | Stok | Harga
1  | Chari          | 100       | 40        | 250 | 16  | 400
2  | Bulba         | 70        | 40        | 200 | 4   | 700
3  | Aspal         | 120       | 30        | 140 | 1   | 1000
4  | Gatot         | 90        | 50        | 130 | 0   | 550
5  | Asepe         | 100       | 40        | 120 | 5   | 600
6  | Gajah         | 90        | 45        | 160 | 2   | 500
7  | Cobra         | 95        | 50        | 180 | 15  | 600
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Mau lihat apa? (monster/potion): potion
ID | Type           | Stok | Harga
1  | strength        | 2    | 20
2  | resilience       | 10   | 5
3  | healing          | 8    | 15
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): 

```

**Gambar 11(b).** Tampilan Shop\_management

```
Masukkan perintah: (ketik 'help' untuk melihat semua perintah)
>>> laboratory
Selamat datang di Lab Dokter Asep !!!
===== MONSTER LIST =====
1. Chari (Level: 2)
2. Bulba (Level: 1)
3. Aspal (Level: 1)
===== UPGRADE PRICE =====
1. Level 1 -> Level 2: 300 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1000 OC
>>> Pilih monster: 2
Bulba akan di-upgrade ke level 2
Harga untuk melakukan upgrade 300
>>> Lanjutkan upgrade (Y/N): Y
Selamat, Bulba berhasil diupgrade ke level 2
>>> Pilih monster: 3
Aspal akan di-upgrade ke level 2
Harga untuk melakukan upgrade 300
>>> Lanjutkan upgrade (Y/N): Y
Selamat, Aspal berhasil diupgrade ke level 2
>>> Pilih monster: keluar

Masukkan perintah: (ketik 'help' untuk melihat semua perintah)
>>> laboratory
Selamat datang di Lab Dokter Asep !!!
===== MONSTER LIST =====
1. Chari (Level: 2)
2. Bulba (Level: 2)
3. Aspal (Level: 2)
===== UPGRADE PRICE =====
1. Level 1 -> Level 2: 300 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1000 OC
>>> Pilih monster: 
```

**Gambar 12.** Tampilan Laboratory

```

Selamat datang di program OWCA!

Masukkan perintah: (ketik 'help' untuk melihat semua perintah)
>>> monster_management
SELAMAT DATANG DI DATABASE PARA MONSTER !!!
1. Tampilkan semua Monster
2. Tambah Monster baru
>>> Pilih Aksi (1/2/keluar): 1
ID | Type       | ATK Power | DEF Power | HP
1  | Chari       | 100       | 40        | 250
2  | Bulba       | 70        | 40        | 200
3  | Aspal       | 120       | 30        | 140
4  | Gatot       | 90        | 50        | 130
5  | Asepe       | 100       | 40        | 120
6  | Gajah       | 90        | 45        | 160
7  | Cobra       | 95        | 50        | 180
8  | Singa       | 100       | 40        | 140
9  | Udin_       | 110       | 40        | 200
>>> Pilih Aksi (1/2/keluar): 2
Memulai pembuatan monster baru
>>> Masukkan type / Nama : Pipit
>>> Masukkan ATK Power (1-1000): 500
>>> Masukkan DEF Power (0-50): 45
>>> Masukkan HP (0-500): 200
Monster baru berhasil dibuat
Type : Pipit
ATK Power : 500
DEF Power : 45
HP : 200
>>> Tambahkan Monster ke database (Y/N) : Y
Monster baru telah ditambahkan!
>>> Pilih Aksi (1/2/keluar): 1
ID | Type       | ATK Power | DEF Power | HP
1  | Chari       | 100       | 40        | 250
2  | Bulba       | 70        | 40        | 200
3  | Aspal       | 120       | 30        | 140
4  | Gatot       | 90        | 50        | 130
5  | Asepe       | 100       | 40        | 120
6  | Gajah       | 90        | 45        | 160
7  | Cobra       | 95        | 50        | 180
8  | Singa       | 100       | 40        | 140
9  | Udin_       | 110       | 40        | 200
10 | Pipit       | 500       | 45        | 200
>>> Pilih Aksi (1/2/keluar): █

```

Gambar 13. Tampilan Monster\_management

```

>>> jackpot
Jumlah OC anda 1000
Apakah Anda siap untuk menguji keberuntungan? Menangkan Snorleks dengan 400 OC saja !!!
==== DAFTAR ITEM ====
1. Beras: 50 OC
2. Telur: 100 OC
3. Kopi: 200 OC
4. Baju: 300 OC
5. Koin: 500 OC
Mulai bermain? (y/n)
>>> y
=====
===== Beras Beras Beras =====
=====
JACKPOT!!! Selamat, Anda mendapatkan monster Bulba

```

Gambar 14. Tampilan Jackpot

```

src
├── __pycache__
├── data
│   ├── cobab
│   └── data1
├── .gitkeep
├── item_inventory.csv
├── item_shop.csv
├── monster_inventory.csv
├── monster_shop.csv
├── monster.csv
├── user_login.csv
└── user.csv

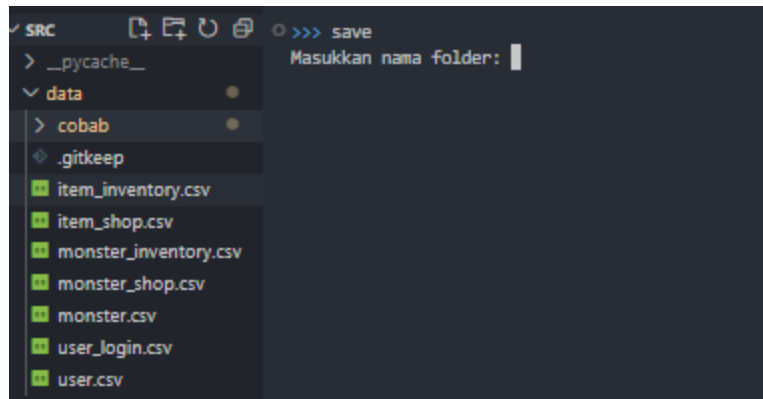
PS C:\Users\DELL\Documents\1. FILE ITB\TUBES DASPRO\if1210-2024-tubes-k05-b\src> python 1.main.py
Tidak ada nama folder yang diberikan!
Usage : python main.py <folder_name>
PS C:\Users\DELL\Documents\1. FILE ITB\TUBES DASPRO\if1210-2024-tubes-k05-b\src> python 1.main.py p
Folder 'p' tidak ditemukan.
PS C:\Users\DELL\Documents\1. FILE ITB\TUBES DASPRO\if1210-2024-tubes-k05-b\src> python 1.main.py cobab

Selamat datang di program OMCA!

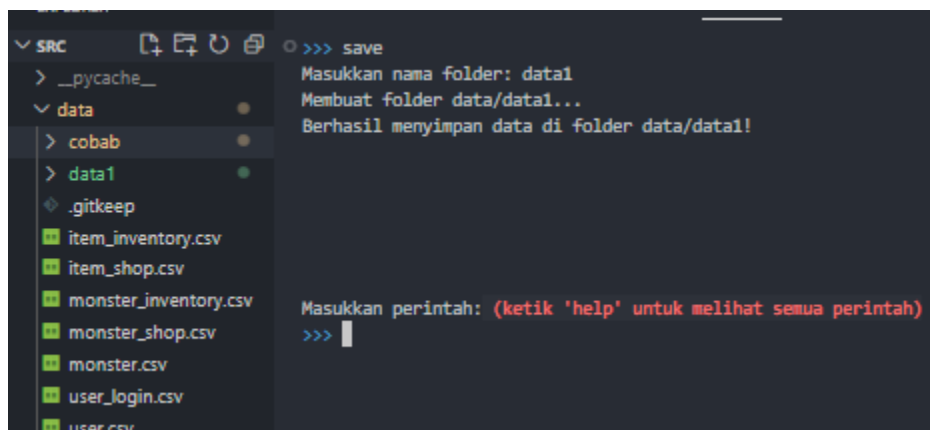
Masukkan perintah: (ketik 'help' untuk melihat semua perintah)
>>>

```

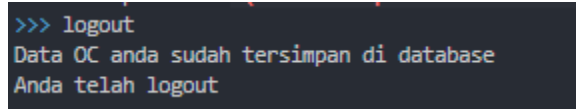
Gambar 15. Tampilan Load



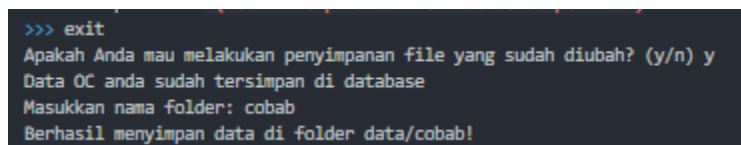
**Gambar 16(a).** Tampilan Save (Before)



**Gambar 16(b).** Tampilan Save (After)



**Gambar 17.** Tampilan Logout



**Gambar 18.** Tampilan Exit




## LAMPIRAN

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

Nomor Asistensi : 1  
 No. Kelompok/Kelas : B/K-05  
 Tanggal asistensi : 2 Mei 2024

Anggota kelompok	NIM / Nama (Hanya yang Hadir)	
	1	19623245/Ahmad Wicaksono
	2	19623105/Shannon Aurellius A. Lie
	3	16523175/B. Alfin Geraldine Baya
	4	16523045/Bryan R. Ryanto Purba
	5	19623305 / Abdullah Farhan
Asisten pembimbing	NIM/NAMA	
	13521135 / Nicholas Liem	

Catatan Asistensi:

<p><b>Rangkuman Diskusi</b></p> <p>Q : Apakah yang harus dilakukan untuk mengubah bentuk csv ke matrix</p> <p>A : membuat fungsi-fungsi pembantu yang dapat digunakan untuk memproses csv ke matrix dalam suatu file parseran</p> <p>Q : Apakah flowchart dibuat berdasarkan coding python yang ada?</p> <p>A : Bukan, tetapi berdasarkan urutan proses untuk setiap fitur (diberikan contoh referensi laporan)</p> <p>Q : Apakah nama file harus berupa f00, f01, dst?</p> <p>A : Tidak, saya sangat menyarankan untuk mengganti nama file dengan nama fungsi yang ingin dijalankan untuk mempermudah pemanggilan dan membuat nama file menjadi bermakna (tidak membingungkan).</p> <p><b>Tindak Lanjut</b></p> <ol style="list-style-type: none"> <li>1. Membuat file parseran untuk read_csv dan write_csv</li> <li>2. Membuat flowchart berdasarkan urutan proses untuk setiap fitur</li> <li>3. Menerima saran penggantian nama file program</li> </ol>
<p><b>Dokumentasi</b></p> 

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

Nomor Asistensi : 2  
 No. Kelompok/Kelas : B/K-05  
 Tanggal asistensi : 8 Mei 2024

Anggota kelompok	NIM / Nama (Hanya yang Hadir)
	1 19623245/Ahmad Wicaksono
	2 19623105/Shannon Aurellius A. Lie
	3 16523175/B. Alfin Geraldine Baya
	4 16523045/Bryan R. Ryanto Purba
	5 19623305 / Abdullah Farhan
Asisten pembimbing	NIM/NAMA
	13521135 / Nicholas Liem

**Catatan Asistensi:**

<b>Rangkuman Diskusi</b> Q : Apakah semua fungsi boleh di gabung ke dalam main.py ? Q : Apakah jumlah monster dapat di inisiasi sendiri atau tidak ? Q : Apakah wajib bikin flowchart ?
<b>Tindak Lanjut</b> 1. Membuat sebuah modul yang dapat mengimport semua fungsi yang ada 2. Membuat nama file yang lebih jelas 3. Membuat folder utils yang berisi fungsi fungsi yang bukan merupakan fungsi utama 4. Monster dapat dengan bebas di inisiasi dan disarankan jumlah monster sebanyak 10 5. Membuat flowchart untuk fungsi-fungsi yang penting saja
<b>Dokumentasi</b> 