

**Laporan Tugas Besar**  
**IF1210-07 Dasar Pemrograman**  
**SEMESTER II 2023/2024.**

*Program Fungsional Game*

**Kelompok K07-K :**

Jerry Alexander Tjoa ( 16523027 )

Adicahya Sukma Kencana Putra Sunjaya ( 16523187 )

Muhammad Zarfana ( 16523227 )

Haura Hafizha Harryson ( 16523257 )

Adinda Putri ( 19623137 )

Dzaky Aurelia Fawwaz (19623297)



SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG

2024

## **Pernyataan Kelompok**

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

Yang mengeluarkan pernyataan,

Jerry Alexander Tjoa ( 16523027 )

Adicahya Sukma Kencana Putra Sunjaya ( 16523187 )

Muhammad Zarfah ( 16523227 )

Haura Hafizha Harryson ( 16523257 )

Adinda Putri ( 19623137 )

Dzaky Aurelia Fawwaz (19623297)

# Daftar Isi

Pernyataan Kelompok.....	2
Daftar Isi.....	3
Daftar Tabel.....	4
Daftar Gambar.....	5
Deskripsi Persoalan .....	6
Daftar Pembagian Kerja Kelompok .....	8
Desain Perintah .....	10
Desain Kamus Data .....	29
Desain Dekomposisi Algoritmik dan Fungsional Program .....	35
Hasil Program.....	100
Lampiran Hasil Pengujian Program .....	126
Lampiran Form Asistensi.....	139

## **Daftar Tabel**

Tabel 1 Daftar Pembagian Kerja Kelompok .....	8
Tabel 2 Checklist Hasil Rancangan .....	9
Tabel Lampiran Hasil Pengujian Program .....	127

## Daftar Gambar

Gambar 1 Deskripsi Persoalan .....	6
Dokumentasi Form Asistensi 1 .....	140
Dokumentasi Form Asistensi 2 .....	141

## Deskripsi Persoalan



Di pinggiran kota Danville, Purry si Platypus, yang juga dikenal sebagai Agent P, sedang menghadapi tantangan yang besar. Dr. Asep Spakbor, seorang ilmuwan jahat, telah menciptakan monster-monster mengerikan yang mengancam keamanan kota Danville. Tugas Purry adalah untuk menghentikan ancaman ini, tetapi monster terbaru Dr. Asep Spakbor terlalu kuat bahkan bagi Purry.

Purry menyadari bahwa dia membutuhkan bantuan dari agen-agen lainnya untuk mengalahkan monster itu. Dia bergegas ke markas rahasia O.W.C.A. (Organisasi Warga Cool Abiez) di mana dia bertemu dengan agen-agen lainnya, yaitu kalian.

Purry mengatakan bahwa untuk mengalahkan Dr. Asep Spakbor, kalian harus bekerja sama untuk mengalahkan monster-monster kuat. Kalian harus merencanakan strategi yang matang agar dapat menyelesaikan misi ini dengan baik. Setelah mendengarkan perkataan Purry, kalian bersiap-siap untuk mengeksekusi rencana dengan teliti.

Purry pun meminta bantuan kalian untuk mencari dan melatih monster-monster sendiri untuk digunakan dalam pertempuran melawan Dr. Asep Spakbor. Kalian merasa tertantang oleh misi ini, namun kalian juga yakin bahwa dengan kerja tim dan tekad yang kuat, pasti bisa berhasil.

Kalian memutuskan untuk memulai misi pencarian monster di hutan terpencil yang diyakini menjadi tempat tinggal bagi banyak jenis monster. Petualangan kalian di hutan yang gelap dan misterius ini akan menguji keberanian dan ketangkasan kalian. Kalian

harus siap menghadapi segala tantangan yang mungkin muncul di perjalanan ini demi keselamatan kota Danville.

# Daftar Pembagian Kerja Kelompok

Table 1 Daftar Pembagian Kerja Kelompok

<b>Fitur</b>	<b>Implementasi</b>	<b>NIM Desainer</b>	<b>NIM Coder</b>	<b>NIM Tester</b>
Main	19623137 19623297	19623137 19623297	19623137 19623297	19623137 19623297
<u>F00 - RNG</u>	16523187	16523187	16523187	16523187
<u>F01 - Register</u>	19623137	19623137	19623137	19623137
<u>F02 - Login</u>	19623137	19623137	19623137	19623137
<u>F03 - Logout</u>	19623137	19623137	19623137	19623137
<u>F04 - Menu &amp; Help</u>	16523027	16523027	16523027	16523027
<u>F05 - Monster</u>	16523027 19623297	16523027	16523027	16523027
<u>F06 - Potion</u>	16523027	16523027	16523027	16523027
<u>F07 - Inventory</u>	16523027	16523027	16523027	16523027
<u>F08 - Battle</u>	19623297	19623297	19623297	19623297
<u>F09 - Arena</u>	19623297	19623297	19623297	19623297
<u>F10 - Shop &amp; Currency</u>	19623297	19623297	19623297	19623297
<u>F11 - Laboratory</u>	19623297	19623297	19623297	19623297
<u>F12 - Shop Management</u>	19623297	19623297	19623297	19623297
<u>F13 - Monster Management</u>	19623297	19623297	19623297	19623297
<u>F14 - Load</u>	19623137	19623137	19623137	19623137
<u>F15 - Save</u>	19623137	19623137	19623137	19623137
<u>F16 - Exit</u>	19623137	19623137	19623137	19623137
B03 – Monster Ball	19623297	16523027	16523027	16523027
B04 - Jackpot		16523027	16523027	16523027
CSV tools	16523257	16523257	16523257	16523257
Notasi Algoritmik	16523027, 16523187, 16523,227,16523,257			



**Table 2 Checklist Hasil Rancangan**

<b>Fitur</b>	<b>Design</b>	<b>Implementasi</b>	<b>Testing</b>
<u>F00 - Random Number Generator</u>	✓	✓	✓
<u>F01 - Register</u>	✓	✓	✓
<u>F02 - Login</u>	✓	✓	✓
<u>F03 - Logout</u>	✓	✓	✓
<u>F04 - Menu &amp; Help</u>	✓	✓	✓
<u>F05 - Monster</u>	✓	✓	✓
<u>F06 - Potion</u>	✓	✓	✓
<u>F07 - Inventory</u>	✓	✓	✓
<u>F08 - Battle</u>	✓	✓	✓
<u>F09 - Arena</u>	✓	✓	✓
<u>F10 - Shop &amp; Currency</u>	✓	✓	✓
<u>F11 - Laboratory</u>	✓	✓	✓
<u>F12 - Shop Management</u>	✓	✓	✓
<u>F13 - Monster Management</u>	✓	✓	✓
<u>F14 - Load</u>	✓	✓	✓
<u>F15 - Save</u>	✓	✓	✓
<u>F16 - Exit</u>	✓	✓	✓
<u>B03 - Monster Ball</u>	✓	✓	✓
<u>B04 - Jackpot</u>	✓	✓	✓

# Desain Perintah

## 1. F00 – RNG

```
# Sebagai admin, saya ingin menggunakan generator angka acak (RNG) untuk
menghasilkan bilangan acak dalam suatu rentang nilai tertentu menggunakan
algoritma Linear Congruential Generator (LCG) tanpa menggunakan library
random dari Python.
>>> generate_random_number
Masukan batas bawah: 10
Masukan batas atas: 50

# Menggunakan algoritma LCG untuk menghasilkan angka acak.
Angka acak yang dihasilkan dalam rentang 10 hingga 50 adalah: 37
```

## 2. F01 – Register

```
# Sebagai admin ketika user memasukkan masukan username dengan karakter
yang diperbolehkan (alfabet huruf besar (A-Z) atau kecil (a-z), angka (0-
9), underscore (_), atau tanda strip (-) dan tidak ada username yang sama
dengan username yang sudah terdaftar sebelumnya.

>>> register([], None, [], [['1', 'Dragon'], ['2', 'Phoenix'], ['3',
'Griffin']])

Masukkan username: kei_shafy16
Masukkan password: keshaf_chocopie

Silahkan pilih salah satu monster sebagai monster awalmu.

1. Dragon
2. Phoenix
3. Griffin

Monster pilihanmu: 2

Selamat datang Agent kei_shafy16. Mari kita mengalahkan Dr. Asep Spakbor
dengan Phoenix!

# Daftar pengguna yang terdaftar dan inventaris monster setelah
registrasi

[['457', 'kei_shafy16', 'keshaf_chocopie', 'Agent', '0']]
[['457', '2', '1']]
```

### 3. F02 – Login

```
# Sebagai admin, ketika user mencoba untuk login dengan memasukkan username
dan password yang sesuai dengan data yang terdaftar.
>>> login([[ '1', 'admin', 'admin123', 'Admin', '0'], [ '2', 'agentA',
'pass123', 'Agent', '10']], None, None, None, None)
Username: admin
Password: admin123
Selamat datang, Admin admin!
Masukkan command "HELP" untuk daftar command yang dapat kamu panggil.

# Daftar pengguna yang terdaftar dan status login setelah login berhasil
Admin
0
1
Admin
>>>

# Sebagai agen, ketika user mencoba untuk login dengan memasukkan username
dan password yang sesuai dengan data yang terdaftar.
>>> login([[ '1', 'admin', 'admin123', 'Admin', '0'], [ '2', 'agentA',
'pass123', 'Agent', '10']], None, None, None, None)
Username: agentA
Password: pass123
Selamat datang, Agent agentA!
Masukkan command "HELP" untuk daftar command yang dapat kamu panggil.

# Daftar pengguna yang terdaftar dan status login setelah login berhasil
agentA
10
2
Agent
>>>

# Ketika user mencoba login dengan username yang tidak terdaftar
>>> login([[ '1', 'admin', 'admin123', 'Admin', '0'], [ '2', 'agentA',
'pass123', 'Agent', '10']], None, None, None, None)
Username: unknownUser
Password: somepass
Username tidak terdaftar!
```

```
# Ketika user mencoba login dengan password yang salah
>>> login([[ '1', 'admin', 'admin123', 'Admin', '0'], [ '2', 'agentA',
'pass123', 'Agent', '10']], None, None, None, None)
Username: admin
Password: wrongpass
Password salah!

# Ketika user yang sudah login mencoba untuk login Kembali
>>> login([[ '1', 'admin', 'admin123', 'Admin', '0'], [ '2', 'agentA',
'pass123', 'Agent', '10']], 'admin', 1, 'Admin', 0)
Login gagal!

Anda telah login dengan username admin silahkan lakukan "LOGOUT" sebelum
melakukan login kembali.
>>>
```

#### 4. F03 – Logout

```
# Ketika user yang sedang login mencoba untuk logout
>>> logout('admin', 'Admin', 0, 1)
Anda telah logout. Silahkan login kembali untuk memulai permainan.

# Status login setelah logout berhasil
None
None
None
None

# Ketika user yang belum login mencoba untuk logout
>>> logout(None, None, None, None)
Logout gagal!
Anda belum login, silahkan login terlebih dahulu sebelum melakukan logout.

# Status login tetap tidak berubah setelah gagal logout
None
None
None
None
```

#### 5. F04 – Menu & Help

```

# Ketika user belum login dan memanggil help
>>> help(None, None)
===== 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
>>>

# Ketika Admin telah login dan memanggil menu_and_help
>>> help('Admin', 'admin')
===== HELP =====
Selamat datang, Admin admin
Berikut adalah hal-hal yang dapat kamu lakukan:
1. Logout: Keluar dari akun yang sedang digunakan
2. Shop Management: Melakukan manajemen pada SHOP sebagai tempat jual beli peralatan Agent
3. Monster Management: Menambahkan monster pada database
4. Owcadex: Membuka OWCADEX untuk melihat list monsters
5. Save: Menyimpan data game
6. Exit: Keluar dari game
Footnote:
1. Untuk menggunakan aplikasi, masukkan nama fungsi yang terdaftar
2. Pastikan memasukkan input yang valid
>>>

# Ketika Agent telah login dan memanggil menu_and_help
>>> menu_and_help('Agent', 'agentA')
===== HELP =====
Halo Agent agentA. Kamu memanggil command HELP. Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian. Berikut adalah hal-hal yang dapat kamu lakukan sekarang:
1. Logout: Keluar dari akun yang sedang digunakan
2. Inventory: Melihat monsters dan items yang dimiliki Agent
3. Owcadex: Membuka OWCADEX untuk melihat list monsters
4. Battle: Melakukan battle dengan monster lain
5. Arena: Menginisiasi Arena untuk melatih monster
6. Shop: Membuka shop untuk membeli items
7. Laboratory: Menaikkan level monster yang dimiliki
8. Jackpot: Melakukan gacha
9. Save: Menyimpan data game
10. Exit: Keluar dari game
Footnote:
1. Untuk menggunakan aplikasi, masukkan nama fungsi yang terdaftar
2. Pastikan memasukkan input yang valid
>>>

```

## 6. F05 – Monster

```

# Sebagai admin, saya ingin melakukan perhitungan damage yang dilakukan
oleh karakter menggunakan fungsi damage_dealt dan damage_taken.

>>> damage_dealt

Masukkan kekuatan serangan (atk_power): 100

# Menghasilkan damage yang dilakukan dalam rentang yang telah ditentukan.

Damage yang dihasilkan adalah: 85
>>> damage_taken

Masukkan kekuatan pertahanan (def_power): 50

Masukkan damage yang diterima (damage_dealt): 85

# Menghasilkan total damage yang diblokir dan total damage yang diterima.

Damage yang diterima setelah pertahanan adalah: 42.5

# Sebagai admin, saya ingin melihat daftar monster dalam sistem
menggunakan fungsi Owcadex.

>>> Owcadex

Masukkan jumlah monster: 2

# Menampilkan daftar monster dalam sistem.

=====
| .#####  ##  ##  ##.#####  ##          :####. :#####. :##  |
|  ## '###:###:##  ### '##  .##:          :## '##  ##  ##.##  |
|  ##  :###:###:##:##  .###          :##  ##  :###. '##'  |
|  ##  :###:###:###  ##  :###: :###: :##  ##  :###'  .##.  |
|  ##.##: '##  '##  ##.##:##'##          :##.##  :###. :##'##.  |
|                                     |
|  '###:  ##  :##  '###:##  :##          :###: :#####  ##  |
=====

Index 1
-----

```

```

Monster Type   : Fire
Attack Power   : 120
Defense Power  : 80
Monster Health : 300
=====

Index 2
-----

Monster Type   : Water
Attack Power   : 90
Defense Power  : 70
Monster Health : 250
=====

# Sebagai admin, saya ingin menampilkan grafik ASCII menggunakan fungsi
call_graphics.

>>> call_graphics
Masukkan ID grafik (1-5): 3

# Menampilkan grafik ASCII sesuai dengan ID yang dimasukkan.

Grafik 3

```

```

# Sebagai admin, saya ingin menggunakan generator angka acak (RNG) untuk
menghasilkan bilangan acak dalam suatu rentang nilai tertentu menggunakan
algoritma Linear Congruential Generator (LCG) tanpa menggunakan library
random dari Python.

>>> generate_random_number

Masukan batas bawah: 10
Masukan batas atas: 50

# Menggunakan algoritma LCG untuk menghasilkan angka acak.

Angka acak yang dihasilkan dalam rentang 10 hingga 50 adalah: 37

```

## 7. F06 – Potion

```
# Sebagai admin, saya ingin menggunakan strength_potion untuk meningkatkan
kekuatan serangan selama battle.
>>> strength_potion
Masukkan kekuatan serangan saat ini (battle_atk_power): 100
Masukkan penggunaan strength potion (0 jika belum digunakan, 1 jika sudah):
0

# Menggunakan strength potion untuk meningkatkan kekuatan serangan.
Strength potion hanya bisa digunakan sekali
Kekuatan serangan setelah menggunakan strength potion adalah: 105.0
Penggunaan strength potion sekarang: 1
>>>
```

```
# Sebagai admin, saya ingin menggunakan resilience_potion untuk
meningkatkan kekuatan pertahanan selama battle.
>>> resilience_potion
Masukkan kekuatan pertahanan saat ini (battle_def_power): 80
Masukkan penggunaan resilience potion (0 jika belum digunakan, 1 jika
sudah): 0

# Menggunakan resilience potion untuk meningkatkan kekuatan pertahanan.
Resilience potion hanya boleh digunakan sekali
Kekuatan pertahanan setelah menggunakan resilience potion adalah: 84.0
Penggunaan resilience potion sekarang: 1
>>>
```

```
# Sebagai admin, saya ingin menggunakan healing_potion untuk memulihkan
Kesehatan selama battle.
>>> healing_potion
Masukkan kesehatan saat ini (battle_current_hp): 60
Masukkan kesehatan maksimal (max_hp): 100
Masukkan penggunaan healing potion (0 jika belum digunakan, 1 jika sudah):
0

# Menggunakan healing potion untuk memulihkan kesehatan.
Healing potion hanya boleh digunakan sekali
Kesehatan setelah menggunakan healing potion adalah: 75.0
Penggunaan healing potion sekarang: 1
>>>
```

```
# Sebagai admin, saya ingin membaca data monster dari file CSV menggunakan
fungsi read_csv.
>>> read_csv
Masukkan path file CSV: monster.csv
```



```
# Membaca data monster dari file CSV.
Data monster yang dibaca adalah:
[['1', 'Fire', '120', '80', '300'],
 ['2', 'Water', '90', '70', '250'],
 ['3', 'Earth', '100', '90', '400'],
 ['4', 'Air', '110', '85', '350'],
 ['5', 'Electric', '130', '75', '320']]
>>>

# Sebagai admin, saya ingin menggunakan semua potion dalam pertempuran.
>>> battle
Masukkan path file CSV: monster.csv
Masukkan index monster yang akan digunakan (mulai dari 0): 4
Masukkan kesehatan saat ini (battle_current_hp): 60

# Menggunakan semua potion dan menampilkan status akhir monster.
Kekuatan serangan setelah menggunakan strength potion adalah: 136.5
Kekuatan pertahanan setelah menggunakan resilience potion adalah: 78.75
Kesehatan setelah menggunakan healing potion adalah: 75.0
>>>
```

## 8. F07 – Inventory

```
# Sebagai admin, saya ingin membaca data monster dari file CSV menggunakan
fungsi read_csv.
>>> read_csv
Masukkan path file CSV untuk monster: monster.csv
Masukkan path file CSV untuk monster inventory: monster_inventory.csv
Masukkan path file CSV untuk item inventory: item_inventory.csv

# Membaca data monster dan inventory dari file CSV.
Data monster yang dibaca adalah:
[['1', 'Fire', '120', '80', '300'],
 ['2', 'Water', '90', '70', '250'],
 ['3', 'Earth', '100', '90', '400'],
 ['4', 'Air', '110', '85', '350'],
 ['5', 'Electric', '130', '75', '320']]

Data monster inventory yang dibaca adalah:
[['1', '2', '1'],
 ['1', '4', '3'],
 ['1', '5', '2']]
```

```
Data item inventory yang dibaca adalah:
[['1', 'Monster Ball', '5'],
 ['1', 'Potion', '10']]
>>>
```

```
# Sebagai admin, saya ingin menampilkan inventory dari pengguna.
```

```
>>> INVENTORY
```

```
# Menampilkan inventory dari pengguna.
```

```
===== INVENTORY LIST (User ID : 1) =====
```

```
Jumlah O.W.C.A. Coin-mu sekarang 100
```

```
1. Monster (Name: Water , Lvl: 1 , HP: 250)
```

```
2. Monster (Name: Air , Lvl: 3 , HP: 367.5)
```

```
3. Monster (Name: Electric , Lvl: 2 , HP: 336.0)
```

```
4. Monster Ball (Qty: 5)
```

```
5. Potion (Type: Potion , Qty: 10)
```

```
Ketikkan id untuk menampilkan id item :
```

```
# Menampilkan detail dari item atau monster berdasarkan ID.
```

```
3
```

```
Monster
```

```
Name : Electric
```

```
ATK Power : 130.0
```

```
DEF Power : 75.0
```

```
HP : 320.0
```

```
Level : 2
```

```
Ketikkan id untuk menampilkan id item :
```

```
4
```

```
Monster Ball
```

```
Quantity : 5
```

```
Ketikkan id untuk menampilkan id item :
```

```
2
```

```
Monster
```

```
Name : Air
```

```
ATK Power : 110.0
```

```
DEF Power : 85.0
```

```
HP : 367.5
```

```
Level : 3
```

```
Ketikkan id untuk menampilkan id item :
```

```
KELUAR
```

```
# Keluar dari tampilan inventory.
```

## 9. F08 – Battle

```
# Sebagai admin, saya ingin menginisialisasi generator nomor acak (RNG) menggunakan algoritma Linear Congruential Generator (LCG) untuk menghasilkan bilangan acak dalam suatu rentang nilai tertentu.
```

```
>>> initialize_rng
```

```
Masukkan m, a, c, x0, dan rentang batas atas: 2147483648, 1103515245, 12345, 12345, 30
```

```
# Inisialisasi RNG menggunakan algoritma LCG.
```

```
RNG berhasil diinisialisasi dengan m = 2147483648, a = 1103515245, c = 12345, x0 = 12345, dan rentang batas atas = 30.
```

```

# Sebagai admin, saya ingin menampilkan daftar monster yang tersedia dalam
OWCA-DEX.
>>> display_monster_list
Daftar monster yang tersedia:
1. Fire
2. Water
3. Earth
4. Air
5. Electric

# Memilih monster untuk bertarung.
Pilih nomor monster untuk bertarung: 3

# Menampilkan monster yang dipilih untuk bertarung.
Agent X mengeluarkan monster Earth untuk bertarung!

Name      : Earth
ATK Power : 100
DEF Power : 90
Level     : 400

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

# Memilih aksi untuk turn ini.
Pilih aksi: 1

# Menampilkan aksi yang dipilih untuk turn ini.
Action no 1

```

## 10. F09 - Arena

```

# Sebagai agen, saya ingin memasukkan perintah selama giliran saya di
arena.
>>> input_perintah
Masukkan giliran: 1
Masukkan monster pengguna: [user_monster]

# Fungsi untuk menampilkan pilihan perintah dan menerima input perintah
dari pengguna.
===== Turns 1 ([user_monster])=====
1. Attack
2. Use Potion
3. Quit
Pilih perintah: 2

# Sebagai agen, saya ingin memasuki arena untuk bertarung dengan monster
lain.

```

```

>>> arena
Masukkan database: [database]
Masukkan user_id: 'user_id'
Masukkan peran pengguna: 'user_role'

# Fungsi utama untuk memasuki arena dan bertarung dengan monster.
Selamat datang di arena!

# Jika pengguna bukan agen.
Ini area bertarung agen!

# Selamat datang di arena.
=====STAGES 1=====
ROARRR [enemy_monster] telah muncul.
HUFTTT Agen telah mengeluarkan monster [user_monster].

# Memilih perintah.
===== Turns 1 ([user_monster])=====
1. Attack
2. Use Potion
3. Quit
Pilih perintah: 2

# Menampilkan daftar potion.
Potion tersedia:
1. Strength Potion
2. Resilience Potion
3. Healing Potion
Pilih potion: 1
Anda telah menggunakan Strength Potion.

# Jika pengguna memilih untuk menyerang.
Anda memilih untuk menyerang.
Pertarungan berlangsung...

# Jika pengguna memilih untuk keluar dari arena.
Pergi dari arena.

```

```

# Jika pengguna memenangkan pertarungan pada tahap tertentu dan lanjut ke
tahap berikutnya.
Keren euy anda mengalahkan monster [enemy_monster].
Anda mendapatkan [oc_reward] pada stage ini.
STAGE SELANJUTNYA :

```

```
=====STAGES 2=====
```

```
# Jika pengguna memenangkan pertarungan pada tahap terakhir (tahap 5).  
Anda telah mengalahkan final boss (monster [enemy_monster]).  
Anda mendapatkan [oc_reward] pada stage ini.
```

## 11. F10 - Shop & Currency

```
# Sebagai agen, saya ingin melihat daftar barang yang tersedia di toko  
(baik monster maupun potion).
```

```
>>> lihat
```

```
Selamat datang di Indoseptember! Silakan pilih aksi:
```

1. Lihat barang (monster)
2. Lihat barang (potion)
3. Keluar

```
# Memilih untuk melihat daftar monster yang tersedia di toko.
```

```
Pilih nomor aksi: 1
```

```
Daftar monster yang tersedia:
```

1. Fire - Stock: 10 - Harga: **100 OC**
2. Water - Stock: 15 - Harga: **150 OC**
3. Earth - Stock: 5 - Harga: **200 OC**
4. Air - Stock: 20 - Harga: **120 OC**
5. Electric - Stock: 8 - Harga: **180 OC**

```
# Memilih untuk melihat daftar potion yang tersedia di toko.
```

```
Pilih nomor aksi: 2
```

```
Daftar potion yang tersedia:
```

1. Health Potion - Stock: **20** - Harga: **50 OC**
2. Strength Potion - Stock: **15** - Harga: **80 OC**
3. Resilience Potion - Stock: **10** - Harga: **70 OC**
4. Speed Potion - Stock: **25** - Harga: **100 OC**
5. Mana Potion - Stock: **18** - Harga: **90 OC**

```
# Memilih untuk membeli barang di toko.
```

```
Pilih nomor aksi: 2
```

```
Mau beli apa (monster/potion)?
```

1. **Monster**

## 2. Potion

```
# Memilih untuk membeli monster.
Pilih nomor aksi: 1

Jumlah O.W.C.A Coin-mu sekarang: 5000

Masukkan ID monster yang ingin dibeli: 3
Monster Earth berhasil dibeli dengan harga 200 OC.
# Memilih untuk membeli potion.
Pilih nomor aksi: 2

Jumlah O.W.C.A Coin-mu sekarang: 4800

Masukkan ID potion yang ingin dibeli: 2
Mau beli berapa banyak potion? : 5
Potion Strength Potion berhasil dibeli dengan harga 400 OC.

# Keluar dari menu belanja.
Pilih nomor aksi: 3

Terimakasih sudah berbelanja di Indoseptember!
```

## 12. F11 - Laboratory

```
# Sebagai admin, saya ingin mengupdate inventory user berdasarkan data yang
diberikan.
>>> update_inventory_by_id
Masukkan general_inventory dari database: [general_inventory]
Masukkan user_inventory yang telah diperbarui: [user_inventory]
Masukkan user_id yang akan diupdate: 'user_id'

# F.S Jika user_id terdapat pada general_inventory, maka row pada user_id
akan diupdate dengan data user_inventory,
# jika tidak maka akan menambahkan row baru dari data user_inventory.
Inventory berhasil diperbarui.
```

```
# Sebagai admin, saya ingin mendapatkan inventory user berdasarkan user_id.
>>> get_user_inventory
Masukkan inventory dari database: [inventory]
Masukkan user_id yang ingin dicari: 'user_id'

# Fungsi mengeluarkan output data user dari database inventory yang
memiliki user_id.
Inventory user dengan user_id 'user_id' adalah: [user_inventory]
```

```
# Sebagai admin, saya ingin mendapatkan nama monster yang terdapat pada
inventory user.
>>> get_monster_name
Masukkan user_monster_inventory: [user_monster_inventory]
```

```
Masukkan tabel monster: [table_monster]

# Fungsi untuk mendapatkan nama monster yang terdapat pada
user_monster_inventory.
Nama monster yang terdapat pada inventory user adalah: [monster_names]
```

```
# Sebagai admin, saya ingin mencetak daftar monster beserta levelnya.
>>> print_monster_list
Masukkan user_monsters: [user_monsters]
Masukkan nama monster: [monster_names]

# Fungsi mencetak daftar monster beserta levelnya.
===== MONSTER LIST =====
1. [nama_monster_1] (Level: [level_1])
2. [nama_monster_2] (Level: [level_2])
...
```

```
# Sebagai admin, saya ingin mencetak harga upgrade level monster.
>>> print_upgrade_price
Masukkan upgrade_prices: [upgrade_prices]

# Fungsi mencetak harga upgrade level monster.
Level 1 -> Level 2: 300
Level 2 -> Level 3: 500
...
```

```
# Sebagai admin, saya ingin meng-upgrade level monster.
>>> upgrading_monster
Masukkan level monster: [monster_level]
Masukkan nama monster: 'monster_name'
Masukkan OC (OC saat ini: [oc]): [oc]
Masukkan harga upgrade: [upgrade_prices]

# F.S Jika awalnya monster level belum maksimum maka monster level ter-
upgrade sampai keinginan user dengan level maks, yaitu level 5.
Monster berhasil di-upgrade menjadi level [monster_level].
```

```
# Sebagai admin, saya ingin mengakses program laboratorium.
>>> laboratory
Masukkan database: [database]
Masukkan data user: [user_data]
```

```
# Program utama laboratorium.
SELAMAT DATANG DI LAB DR ASEP
===== MONSTER LIST =====
1. [nama_monster_1] (Level: [level_1])
2. [nama_monster_2] (Level: [level_2])
...

Level 1 -> Level 2: 300
Level 2 -> Level 3: 500
...

Pilih monster (1 - [banyak_monster]): [index_monster]
Angka OC Anda saat ini adalah: [oc]
Ingin meng-upgrade monster lain (y/n)? [next_monster]
```

### 13. F12 - Shop Management

```
# Sebagai admin, saya ingin menggunakan interface tabel untuk menampilkan data.
```

```
>>> table_interface
Masukkan tabel: [table]
```

```
# Fungsi untuk menampilkan tabel dalam bentuk yang rapi.
Tabel berhasil ditampilkan.
```

```
# Sebagai admin, saya ingin melakukan filter pada tabel berdasarkan id.
```

```
>>> filter_table_by_id
Masukkan tabel1: [table1]
Masukkan tabel2: [table2]
```

```
# Fungsi untuk melakukan filter pada tabel1 berdasarkan id yang tidak ada di tabel2.
Tabel baru berhasil dibuat setelah difilter.
```

```
# Sebagai admin, saya ingin menggabungkan dua tabel berdasarkan id yang sama.
```

```
>>> merge_tables_by_id
Masukkan tabel1: [table1]
Masukkan tabel2: [table2]
```

```
# Fungsi untuk menggabungkan dua tabel berdasarkan id yang sama.
Tabel berhasil digabungkan.
```

```
# Sebagai admin, saya ingin membuat ID untuk item potion yang baru.
```

```
>>> generate_id_potion
Masukkan tabel_potion: [table_potion]
```

```
# Fungsi untuk membuat ID untuk item potion yang baru.
Tabel ID potion berhasil dibuat.
```

```
# Sebagai admin, saya ingin mendapatkan indeks berdasarkan ID pada tabel.
```

```
>>> get_index_by_id
```



```
Masukkan ID yang dicari: 'id'  
Masukkan tabel: [table]
```

```
# Fungsi untuk mendapatkan indeks berdasarkan ID pada tabel.  
Indeks ID 'id' adalah: [indeks]
```

```
# Sebagai admin, saya ingin menghapus baris berdasarkan indeks pada tabel.  
>>> delete_row  
Masukkan indeks baris yang akan dihapus: [indeks]  
Masukkan tabel: [table]
```

```
# Fungsi untuk menghapus baris pada tabel berdasarkan indeks.  
Baris berhasil dihapus dari tabel.
```

```
# Sebagai admin, saya ingin memasukkan data baru ke dalam tabel.
```

```
>>> input_data  
Masukkan objek: 'objek'  
Masukkan status input (awal/baru/hapus): 'status_input'  
Masukkan data tabel: [shop_data]
```

```
# Fungsi untuk memasukkan data baru ke dalam tabel berdasarkan objek dan  
status input.  
Data berhasil dimasukkan ke dalam tabel.
```

```
# Sebagai admin, saya ingin mengelola toko.
```

```
>>> shop_management  
Masukkan database: [database]  
Masukkan data user: [user_data]
```

```
# Program utama untuk mengelola toko.  
Pilih aksi (lihat/tambah/ubah/hapus/keluar): [aksi]
```

#### 14. F14 - Load

```
# Sebagai admin, saya ingin mengecek apakah monster sudah ada di dalam  
tabel monster.
```

```
>>> is_monster_exist  
Masukkan nama monster: "Dragon"  
Apakah monster ada di tabel monster? False
```

```
# Sebagai admin, saya ingin memastikan bahwa input monster belum ada di  
tabel monster.
```

```
>>> input_monster_not_exist  
Masukkan nama monster: "Dragon"  
Masukkan monster yang belum ada: "Phoenix"  
Monster yang dimasukkan adalah: "Phoenix"
```

```
# Sebagai admin, saya ingin mengelola monster dalam database.
```

```
>>> monster_management  
Masukkan database: [database]
```

```

Masukkan peran pengguna: "admin"

# Selamat datang di menu manajemen monster.
SELAMAT DATANG DI DATABASE PARA MONSTER !!!
1. Tampilkan semua Monster
2. Tambah Monster baru
3. Cancel
Pilih aksi (1/2): 2

# Jika admin memilih untuk menambahkan monster baru.
Masukkan type/nama monster: "Phoenix"
Masukkan attack power: "150"
Masukkan def power (range 0 - 50): "30"
Masukkan HP monster: "500"

# Setelah monster berhasil dibuat.
Monster baru telah berhasil dibuat.
Type: Phoenix
ATK Power: 150
DEF Power: 30
HP: 500

Tambahkan monster ke database (Y/N): Y
Monster berhasil ditambahkan.
Ingin menambahkan monster lain (Y/N): N

# Jika admin memilih untuk menampilkan semua monster.
Pilih aksi (1/2): 1
Menampilkan semua monster:
[Daftar monster dalam tabel]

# Jika admin memilih untuk membatalkan.
Pilih aksi (1/2): 3
Mengakhiri manajemen monster.

# Jika pengguna bukan admin.
Tidak memiliki akses.

```

## 15. F15 – Save

```

# Sebagai pengguna, saya ingin menyimpan data sementara ke dalam
folder yang saya pilih.

```

```
>>> save
Masukkan nama folder: "data_pengguna"

# Jika folder belum ada, maka folder akan dibuat terlebih dahulu.
Saving...

Membuat folder data_penggun
Berhasil menyimpan data di folder data_pengguna!

# Jika folder sudah ada, data akan langsung disimpan ke dalam folder
tersebut
Saving...
Berhasil menyimpan data di folder data_pengguna!

# Data yang disimpan meliputi:
- user.csv
- monster.csv
- monster_shop.csv
- monster_inventory.csv
- item_inventory.csv
- item_shop.csv

# Output konfirmasi penyimpanan data.
Data berhasil disimpan di dalam folder data_pengguna.
```

#### 16. F16 - Exit

```
# Sebagai pengguna, saya ingin mengakhiri program dengan opsi untuk
menyimpan perubahan data.
```

```
>>> exit
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) Y
# Jika pengguna memilih "Y", data akan disimpan sebelum program berakhir.
Masukkan nama folder: "data_akhir"

# Jika folder belum ada, maka folder akan dibuat terlebih dahulu.
Saving...
Membuat folder data_akhir
Berhasil menyimpan data di folder data_akhir!
Program selesai.

# Jika folder sudah ada, data akan langsung disimpan ke dalam folder
tersebut.
Saving...
Berhasil menyimpan data di folder data_akhir!
Program selesai.

>>> exit
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) N

# Jika pengguna memilih "N", program akan langsung berakhir tanpa menyimpan
data.
Program selesai.
```

# Desain Kamus Data

## 1. F00 – Random Number Generator

**function** randomNumberGenerator (bottom,top : float) → integer  
bottom, top : float  
m,a,c,current\_time\_ns, pid, microseconds, seed, randomNumber :  
integer

## 2. F01 – Register

**function** register (registered\_user: array of any, logged\_in\_user:  
array of string)[] string  
registered\_user: array of any  
logged\_in\_user: array of string  
username, password : string

## 3. F02 - Login

**function** login (registered\_users, logged\_in\_user, user\_id, user\_role,  
user\_oc : string ) [] string  
registered\_users, logged\_in\_user, user\_id, user\_role, user\_oc: string

## 4. F03 – Logout

**function** logout(logged\_in\_user, user\_role, user\_oc, user\_id :  
string)[] string  
logged\_in\_user, user\_role, user\_oc, user\_id: string

## 5. F04 – Menu and Help

role, username, current\_user : string  
**procedure** user\_sebelum\_login ()  
**procedure** user\_setelah\_login ()

## 6. F05 – Monster

**function** attack (damage\_dealt, low\_atk, high\_atk : integer)[] integer  
**KAMUS LOKAL**  
damage\_dealt, low\_atk, high\_atk, atk\_power: integer

**function** defense (def\_power, damage\_dealt : integer )[] integer  
**KAMUS LOKAL**  
def\_power, damage\_dealt, damage\_defensed, damage\_taken: integer

**procedure** Owcadex (input monster\_array)  
**KAMUS LOKAL**  
monster\_array : array of any

## 7. F06 – Potion

**function** strength\_potion (battle\_atk\_power, strength\_potion\_use :  
integer)[] integer  
**KAMUS LOKAL**  
battle\_atk\_power, strength\_potion\_use: integer

**function** resilience\_potion (battle\_def\_power, resilience\_potion\_use : integer): integer  
damage\_dealt, low\_atk, high\_atk, atk\_power: integer battle\_atk\_power,  
strength\_potion\_use: integer

**function** healing\_potion (battle\_current\_hp, max\_hp,  
healing\_potion\_use : integer)[] integer

**KAMUS LOKAL**

battle\_current\_hp, max\_hp, healing\_potion\_use: integer

## 8. F07 – Inventory

count\_monster, count\_item : integer

inventory\_command : string

**procedure** take\_monster\_id\_level (input user\_id : integer)

**KAMUS LOKAL**

user\_id : integer

monster\_id, monster\_level : array of integer

**procedure** base\_monster\_in\_inventory (input monster\_id : integer)

**KAMUS LOKAL**

monster\_id : integer

monster\_level : array of integer

base\_monster : array of any

**function** level\_calculations (base\_monster : array of any)[] array of any

**KAMUS LOKAL**

base\_monster, monster\_in\_inventory : array of any

**procedure** take\_item\_id (input user\_id : integer)

**KAMUS LOKAL**

user\_id : integer

item\_id : array of integer

item\_inventory : array of any

**procedure** display\_inventory (input user\_id : integer)

**KAMUS LOKAL**

user\_id, count, owca\_coin : integer

current\_inventory\_monster, current\_inventory\_item : array of any

## 9. F08 – Battle

**procedure** select\_available\_monster (input user\_monsters : array of any)

**KAMUS LOKAL**

isFighting : Boolean

user\_monster : list

urutan : integer

**procedure** print\_status (input turns : integer, input

attacker\_monster, defender\_monster : array of any)

**KAMUS LOKAL**turns : integerattacker\_monster, defender\_monster : array of any**procedure** print\_potion (input user\_inventory : array of any)**KAMUS LOKAL**user\_inventory : array of any**procedure** random\_monster\_data (output monster, level\_monster : array of any)**KAMUS LOKAL**monster : integerlevel\_monster : array of integerindex\_monster : integermonster\_enemy : array of any**function** duel (input battle\_status, user\_monster, enemy\_monster list, input oc\_reward : integer) → integer**KAMUS LOKAL**damage\_dealt\_user, damage\_taken\_enemy, oc\_reward : integerbattle\_status, user\_monster, enemy\_monster : array of any**procedure** drink\_potion (input user\_item\_inventory : array of any, input/output potion\_used : integer, user\_monster : list, hp\_max : integer)**KAMUS LOKAL**potion\_arr, daftar\_id, user\_item\_inventory, user\_monster : array of anydrink\_potion, cancel, used : Boolean

potion\_used, hp\_max, id\_potion, index\_potion\_in\_inventory,

current\_stock : integertype\_potion : string**10. F09 – Arena****function** arena (database, user\_data : array of any) → array of any**KAMUS LOKAL**database, user\_data, user\_monster\_inventory, user\_item\_inventory, user\_monster\_, enemy\_monster\_tables, potion\_used, battle\_status : array of anyitem\_inventory, monster\_inventory, monster : stringstage, oc\_awal, hp\_max, turns : integerwin\_battle : Booleanoc\_rewards : array of integer**11. F10 – Shop & Currency****procedure** shop(database, user\_data : array of any)**KAMUS LOKAL**Database, user\_monster, user\_items : array of anyUser\_id, user\_role : string**function** beli(objek, data\_shop, user\_inventory, user\_data, menu : string) → array of any**KAMUS LOKAL**

objek,data\_shop,user\_inventory,user\_data,menu : string

## 12. F11 – Laboratory

**function** laboratory (database, user\_data : array of any)[]array of any

**KAMUS LOKAL**

monster\_name : string

input upgrade\_prices, upgrade prices : array of integer

database, user\_data, monster\_inventory, monster, user\_monster,

monster\_names : array of any

user\_id, oc, banyak\_monster, index\_monster : integer

isLaborating : boolean

**function** update\_inventory\_by\_id (general\_inventory, user\_inventory :  
array of any, user\_id : string) [] array of any

**KAMUS LOKAL**

general\_inventory, user\_inventory, : array of any

user\_id, new\_data : string

found : Boolean

user\_row, general\_row: integer

**function** get\_user\_inventory (inventory : array of any, user\_id :  
string) [] array of any

**KAMUS LOKAL**

inventory : array of any

user\_id : string

**function** get\_monster\_name (user\_monster\_inventory, table\_monster :  
array of any)[] array of any

**KAMUS LOKAL**

user\_monster\_inventory, table\_monster : array of any

monster\_names : array of strings

id\_monster, j : integer

found : boolean

**procedure** print\_monster\_list (input user\_monster, monster\_names :  
array of any)

**KAMUS LOKAL**

user\_monster : array of any

monster\_names : array of string

**procedure** print\_upgrade\_price (input upgrade\_prices : array of  
integer)

**KAMUS LOKAL**

upgrade\_prices : array of integer

## 13. F12 – Shop Management

**function** shop\_management(database,user\_data : array of any)[]array of  
any

**KAMUS LOKAL**

database, user\_data : array of any



**procedure** table\_interface(table : array of any)  $\square$  array of any  
**KAMUS LOKAL**  
table : array of any

**function** filter\_table\_by\_id(table1, table2 : array of any)  $\square$  array of any  
**KAMUS LOKAL**  
table1, table2 : array of any

**function** merge\_tables\_by\_id(table1,table2 : array of any)  $\square$  array of any  
**KAMUS LOKAL**  
table1, table2 : array of any

**function** generate\_id\_potion(table\_potion : array of any)  $\square$  integer  
**KAMUS LOKAL**  
table\_potion : string

**function** get\_index\_by\_id(id : string, table : array of any)  $\square$  integer  
**KAMUS LOKAL**  
id : string  
table : array of any

**function** delete\_row(indeks : integer ,table : array of any)  $\square$  array of any  
**KAMUS LOKAL**  
id : string  
table : array of any

**function** input\_data(objek:string,status\_input:string,shop\_data : array of any)  $\square$  integer  
**KAMUS LOKAL**  
objek : string  
status\_input : string  
shop\_data : array of any

#### 14. F13 – Monster Management

**function** is\_monster\_exist(monster:str,table\_monster)  $\square$  bool  
**KAMUS LOKAL**  
monster:string  
table\_monster : bool

**procedure** monster\_management(database,user\_data)  $\square$  string  
**KAMUS LOKAL**  
database,user\_data : array of any

#### 15. F14 – Load

**function** load(folder\_name : string)  $\square$  string  
**KAMUS LOKAL**  
folder\_name: string

#### 16. F15 - Save

**procedure** save(temp\_data : string)  
**KAMUS LOKAL**  
temp\_data : string

#### **17. F16 - Exit**

**procedure** exit(temp\_data : string )  
**KAMUS LOKAL**  
temp\_data : string

#### **18. B03 – Monster Ball**

**procedure** MonsterBall (input enemy\_array : array of any)  
  
**KAMUS LOKAL**  
succes : Boolean  
level, number : integer

#### **19. B04 - Jackpot**

**procedure** Jackpotgraphics (input number : integer)  
**KAMUS LOKAL**  
item,slot : array of any  
agree : string

# Desain Dekomposisi Algoritmik dan Fungsional

## 1. F00 – Random Number Generator

**function** randomNumberGenerator (input bottom,top : float) → integer

{Mengeluarkan sebuah angka random dalam range yang telah diinput}

{ I.S. Range terdefinisi }

{ F.S. randomNumber berhasil di generate sebagai output }

## 2. F01 – Register

**function** register (registered\_user: list, logged\_in\_user: list) → string

{ Melakukan registrasi pembuatan akun bagi user yang baru mau memulai permainan }

{ I.S. Data user terdefinisi }

{ F.S. Penambahan data user baru pada file }

## 3. F02 – Login

**function** login (registered\_users, logged\_in\_user) → string

{ Melakukan fungsi login untuk membantu user untuk bergabung ke dalam akun permainannya dengan data yang sudah tersimpan dalam permainan }

{ I.S. Registered user terdefinisi }

{ F.S. Logged\_in\_user terdefinisi }

## 4. F03 – Logout

**function** logout(logged\_in\_user) → string

{ Melakukan fungsi logout untuk membantu user keluar dari akun permainan }

{ I.S. logged\_in\_user terdefinisi }

{ F.S. logged\_in\_user mengalami perubahan }

## 5. F04 – Menu and Help

**procedure** user\_sebelum\_login ()

{ Mengeluarkan output tulisan sebagai petunjuk user jika user belum melakukan login }

{ I.S. user belum melakukan login }

{ F.S output help }

**procedure** user\_setelah\_login (input role, username : string)

{ Mengeluarkan output tulisan sebagai petunjuk user jika user telah login }

{ I.S. user telah melakukan login }

{ F.S output help }

## 6. F05 – Monster

**function** attack (damage\_dealt, low\_atk, high\_atk : integer) → integer

{ Mengeluarkan output attack berupa damage yang dilakukan }

{ I.S. Atk power terdefinisi }

{ F.S. Damage dealt terdefinisi }

**function** defense (def\_power, damage\_dealt : integer) → integer

{ Mengeluarkan output defense berupa damage yang dihiraukan }

{ I.S. def power terdefinisi }

{ F.S. damage defended dan damage taken terdefinisi }

**procedure** Owcadex (input monster\_array : array of string)

{ Mengeluarkan output owcadex }

{ I.S. monster\_array terdefinisi }

{ F.S. output owcadex terdefinisi }

## 7. F06 – Potion

**function** strength\_potion (battle\_atk\_power, strength\_potion\_use) → integer

{ Melakukan kalkulasi untuk penggunaan strength\_potion dalam battle }

{ I.S. atk\_power terdefinisi }

{ F.S. atk\_power mengalami perubahan sesuai kalkulasi }

**function** resilience\_potion (battle\_def\_power, resilience\_potion\_use) → integer

```

{ Melakukan kalkulasi untuk penggunaan resilience_potion dalam battle }
    { I.S. def_power terdefinisi }
    { F.S. def_power mengalami perubahan sesuai kalkulasi }
function healing_potion (battle_current_hp, max_hp, healing_potion_use) :
integer
{ Melakukan kalkulasi untuk penggunaan health_potion dalam battle }
{ I.S. health terdefinisi }
{ F.S. health mengalami perubahan sesuai kalkulasi }

```

## 8. F07 – Inventory

```

procedure take_monster_id_level (input user_id : integer)
{ Melakukan pengambilan monster_id serta level dari monster.csv dan user.csv }
    { I.S. array monster_id dan monster_level terdefinisi }
    { F.S. array monster_id mengalami modifikasi sesuai program }
procedure base_monster_in_inventory (input monster_id : integer)
{ Mengambil data base monster sesuai dengan id yang dimiliki tanpa perubahan stats monster }
    { I.S. monster_id terdefinisi dan array base_monster terdefinisi }
    { F.S. array base_monster mengalami modifikasi sesuai dengan program }
function level_calculations (base_monster : array of any ) : array of any
{ Melakukan kalkulasi dari monster di inventory sesuai dengan level monster tersebut }
    { I.S. array base_monster terdefinisi }
    { F.S. array monster_in_inventory terdefinisi }
procedure take_item_id (input user_id : integer)
{ Mengambil list dari item yang dimiliki user sesuai user_id }
    { I.S. user_id terdefinisi dan array item_id terdefinisi }
    { F.S. array current_inventory_item terdefinisi }
procedure display_inventory (input user_id : integer)
{ Melakukan fungsi utama fitur display dari inventory yang dimiliki user }
    { I.S. user_id, array current_inventory_monster, array current_inventory_item terdefinisi }
    { F.S. output dari display inventory terdefinisi }

```

## 9. F08 – Battle

**procedure** battles (input database, user\_data : array of any)

{ Melakukan fitur battle pada game serta semua function dan prosedur yang diperlukan dalam battle }

{ I.S. user\_data terdefinisi }

{ F.S. program battle terdefinisi dan terlaksana }

**procedure** select\_available\_monster (input user\_monsters : list)

{ Melakukan fitur untuk melakukan select monster yang dipake pada battle }

{ I.S. user\_monsters terdefinisi }

{ F.S. user\_monster termodifikasi sesuai program }

**procedure** print\_status (input turns : integer, input attacker\_monster, defender\_monster : list)

{ Melakukan print dari status monster }

{ I.S. turns, attacker\_monster, defender\_monster terdefinisi }

{ F.S. print status terdefinisi }

**procedure** print\_potion (input user\_inventory : list)

{ Melakukan fungsi jika user meminum potion }

{ I.S. potion\_arr terdefinisi }

{ F.S. potion\_quantity terdefinisi }

**procedure** random\_monster\_data (output monster, level\_monster : list)

{ Mengambil monster secara random dari data }

{ I.S. monster dan level\_monster terdefinisi }

{ F.S. monster\_enemy terdefinisi }

**function** duel (input battle\_status, user\_monster, enemy\_monster : list, input oc\_reward : integer)

{ Melakukan fungsi utama duel }

{ I.S. database dan user\_data terdefinisi }

{ F.S. oc dan user\_data mengalami modifikasi sesuai program }

**procedure** drink\_potion (input user\_item\_inventory : list, input/output potion\_used : integer, user\_monster : list, hp\_max : integer)

{ Melakukan fungsi potion jika user memilih potion }

{ I.S. potion\_arr terdefinisi }

{ F.S. user\_monster mengalami modifikasi sesuai program }

## 10. F09 – Arena

```
function arena (database, user_data : array of any) ↵ array of any
{ Melakukan fungsi utama arena sebagai salah satu fitur dalam game }

{ I.S. user_data terdefinisi }

{ F.S. function arena terdefinisi }
```

## 11. F10 – Shop & Currency

```
procedure beli (objek : string, data_shop, user_inventory, user_data, menu :
array of any)
```

```
    { I.S. user_inventory, oc terdefinisi dari database }

    { F.S. user_inventory, oc mengalami modifikasi sesuai program }
```

```
procedure shop (database, user_data : array of any)
```

```
    { I.S. database terdefinisi }

    { F.S. inventory termodifikasi sesuai program }
```

## 12. F11 – Laboratory

```
function laboratory (database, user_data : array of any) ↵ array of any
```

```
{ Fungsi utama laboratory untuk melakukan upgrade level pada monster yang
dimiliki user}
```

```
    { I.S. database serta user_data terdefinisi }

    { F.S. monster_inventory mengalami modifikasi sesuai program }
```

```
function update_inventory_by_id (general_inventory, user_inventory : array of
any, user_id : string) ↵ array of any
```

```
{ Melakukan penambahan row dari user_id dengan data user_inventory }
```

```
    { I.S general_inventory dari database(general_inventory), user_inventory
telah update }
```

```
    { F.S Jika user_id terdapat pada general_inventory, maka row pada user_id
akan diupdate dengan data user_inventory }
```

```
function get_user_inventory (inventory : array of any, user_id : string )↵
array of any
```

```
{ Fungsi mengeluarkan output data user dari database inventory yang memiliki
user_id }
```

```
    { I.S. inventory dan user_id terdefinisi }
```

```

{ F.S. user_inventory terdefinisi }

function get_monster_name (user_monster_inventory, table_monster : array of
any) → array of any

{ Fungsi untuk mendapatkan nama monster yang terdapat pada
user_monster_inventory }

    { I.S. user_monster_inventory dan table_monster terdefinisi }
    { F.S. monster_names terdefinisi }

procedure print_monster_list (input user_monster, monster_names : array of
any)

{ Procedure untuk mencetak list monster }

    { I.S. user_monster dan monster_names terdefinisi }
    { F.S. monster list tercetak }

procedure print_upgrade_price (input upgrade_prices : array of integer)

{ Procedure untuk mencetak harga dari upgrade level }

    { I.S. upgrade_prices terdefinisi }
    { F.S. harga upgrade_level tercetak }

function upgrading_monster (input monster_level, oc : integer, input
monster_name : string, input upgrade_prices : array of integer)

    { I.S monster level belum terupgrade atau sudah maksimum (level 5) }
    { F.S Jika awalnya monster level belum max maka monster level
terupgrade sampai keinginan user dengan level maks,yaitu level 5 }

```

### 13. F12 – Shop Management

```

function shop_management(database,user_data) → array of any

{ Melakukan fungsi utama shop_management untuk penambahan item ke shop }

    { I.S. database dan user_data terdefinisi }
    { F.S. database mengalami modifikasi sesuai program }

procedure table_interface(table : array of any)

{ Membuat menu interface shop dalam bentuk tabel }

    { I.S. table terdefinisi }
    { F.S. table mengalami output sesuai program }

function filter_table_by_id(table1, table2 : array of any) → array of any

{ Membuat tabel baru sesuai filter id }

    { I.S. tabel 1 dan tabel 2 terdefinisi }

```



```

    { F.S. new_table terdefinisi }
function merge_tables_by_id(table1,table2 : array of any) ⌈ array of any
{ Menggabungkan tabel sesuai id }
    { I.S. tabel 1 dan tabel 2 terdefinisi }
    { F.S. merged_table terdefinisi }
function generate_id_potion(table_potion : array of any) ⌈ integer
{ Mengeluarkan sebuah id_potion}
    { I.S. tabel potion terdefinisi }
    { F.S. new_table terdefinisi }
function get_index_by_id(id:str,table) ⌈ integer
{ Mengeluarkan index berdasarkan id }
    { I.S. id dan table terdefinisi }
    { F.S. id_index terdefinisi }
function delete_row(indeks:int,table : array of any) ⌈ array of any
{ Menghilangkan tabel yang tidak diperlukan berdasarkan indeks }
    { I.S. index dan table terdefinisi }
    { F.S. new_table terdefinisi }
function input_data(objek:str,status_input:str,shop_data: array of any)⌈
integer
{ Melakukan fitur input data shop }
    { I.S. objek, status_input dan shop_data terdefinisi }
    { F.S. stock,harga dan id terdefinisi }

```

#### 14. F13 – Monster Management

```

function is_monster_exist(monster:str,table_monster) ⌈ bool
{ Melakukan pengecekan jika monster sudah ada }
    { I.S. monster dan table_monster terdefinisi }
    { F.S. monster mengalami modifikasi sesuai program }
procedure monster_management(database,user_data) ⌈ string
{ Melakukan fungsi utama monster management untuk penambahan monster baru}
    { I.S. database dan user_data terdefinisi }
    { F.S. database mengalami modifikasi sesuai program }

```

### 15. F14 – Load

```
function check_validity_path(path : str) → string
{ Melakukan validasi pada path folder game }
    { I.S. path terdefinisi }
    { F.S. pengecekan path }
function LOAD(folder_name : str) → string
{ Melakukan load dalam progress yang telah dilakukan dalam game }
    { I.S. path terdefinisi }
    { F.S. folder terdefinisi }
```

### 16. F15 – Save

```
function save(matriks : list, ftype : str) → string
{ Melakukan save progress yang dilakukan dalam game }
    { I.S. folder terdefinisi }
    { F.S. game data disimpan }
```

### 17. F16 – Exit

```
procedure exit (input temp_data : array of any )
{ Melakukan exit dari game yang dimainkan }
    { I.S. temp_data terdefinisi }
    { F.S. game melakukan exit }
```

### 18. B03 – Monster Ball

```
procedure MonsterBall (input enemy_array : array)
{Melakukan fungsi utama monster ball}
    {I.S. enemy_array terdefinisi}
    {F.S. monster_inventory mengalami modifikasi sesuai program }
```

### 19. B04 – Jackpot

```
procedure Jackpotgraphics (input number <-- integer)
{Melakukan fungsi print untuk gambar-gambar pada fungsi utama jackpot}
```

{I.S. number terdefinisi}

{F.S. output gambar graphics}

# Notasi Algoritmik Program

- **F00 – Random Number Generator**

## **Program Random Number Generator**

{Mengeluarkan sebuah angka random dalam range yang telah diinput}

{ I.S. Range terdefinisi }

{ F.S. randomNumber berhasil di generate sebagai output }

## **KAMUS**

bottom, top : float

m,a,c,current\_time\_ns, pid, microseconds, seed, randomNumber : integer

## **REALISASI PROSEDUR**

function randomNumberGenerator (bottom,top : float) → integer

## **ALGORITMA**

{bottom = batas bawah range nilai}

{top = batas atas range nilai}

{inisiasi nilai m,a,c, dan seed untuk LCG (most optimal values)}

$m \leftarrow 2^{32}$

$a \leftarrow 1103515245$

$c \leftarrow 12345$  {ngambil waktu sekarang → diconvert jadi int}

time.sleep(0.005)

{seed}

current\_time\_ns ← time.time\_ns()

microseconds ← datetime.datetime.now().microsecond

seed ← current\_time\_ns \* pid + microseconds

randomNumber ←  $((a * seed + c) \% m) / m$

```
randomNumber ← randomNumber * (top - bottom) + bottom {scale and shift ke  
range yang diinginkan}
```

```
return → int(randomNumber) {ngembaliin nilai random}
```

- **F01 - Register**

### **Program Register**

{Melakukan registrasi pembuatan akun bagi user yang baru mau memulai permainan}

{I.S. Data user terdefinisi}

{F.S. Penambahan data user baru pada file}

### **KAMUS LOKAL**

registered\_user: list, logged\_in\_user: list: string

username, password : string

### **REALISASI PROSEDUR**

function register (registered\_user: list, logged\_in\_user: list): string

### **ALGORITMA**

```
    registered_users ← database["user"]  
    monster_inventory ← database["monster_inventory"]  
    monster ← database["monster"]  
    if logged_in_user = None then  
  
        username ← input ("Masukkan username: ")  
        password ← input ("Masukkan password: ")  
  
        if val.check_username_availability(registered_users, username)  
        and val.check_username_validity(username) then  
            generated_id ← RNG.randomNumberGenerator (1, 1000)  
  
            while val.is_registered_id(registered_users,  
generated_id) do  
  
                generated_id ← RNG.randomNumberGenerator  
(1, 1000)  
  
            registered_users.append([str(generated_id), username,  
password, "Agent", "0"])  
  
            output ("\nSilahkan pilih salah satu monster sebagai  
monster awalmu.")  
  
            monster_name ← get_data_by_column_title(monster,  
"type")  
  
            monster_id ← get_data_by_column_title(monster, "id")  
  
            i traversal [1 ... len(monster))]  
                output (f"{i}. {monster[i][1]}")
```

```

monster_choice ← int (input ("\nMonster pilihanmu:
"))
monster_inventory.append([generated_id,
monster_id[monster_choice-1], monster_name[monster_choice-1]])
    output(f"Selamat datang Agent {username}. Maria kita
mengalahkan Dr. Asep Spakbor dengan {monster_name[monster_choice-1]}!")
else
    if not val.check_username_availability(username) then
        output(f"\nUsername {username} sudah terpakai,
silahkan gunakan username lain!")
    else
        output("\nUsername hanya boleh berisi alfabet, angka,
underscore, dan strip!")

else
    output(f'\nRegister gagal!\nAnda telah login dengan username
{logged_in_user} silahkan lakukan "LOGOUT" sebelum melakukan register.')
return → database

```

## - F02 - Login

### **Program Login**

{Melakukan fungsi login untuk membantu user untuk bergabung ke dalam akun permainannya dengan data yang sudah tersimpan dalam permainan}

{I.S. Registered user terdefinisi}

{F.S. Logged\_in\_user terdefinisi}

### **KAMUS**

```

function login (registered_users, logged_in_user, user_id, user_role, user_oc):
string
registered_users, logged_in_user, user_id, user_role, user_oc: string

```

### **REALISASI PROSEDUR**

```

function login (registered_users, logged_in_user, user_id, user_role, user_oc):
string

```

### **ALGORITMA**

```

    registered_users = database["user"]
    if logged_in_user = None then
        username ← input ("Username: ")
        password ← input ("Password: ")
        registered_username ← uf.get_data_by_column_title(registered_users,
'username')
        if username not in registered_username then
            output ("Username tidak terdaftar!")
            user_id,user_role ← None,None
        else

```

```

        user_password ← uf.get_row(registered_users, 'username',
username)[0][2]
        if password = user_password then
            logged_in_user ← username
            user_id ← int(uf.get_row(registered_users, 'username',
username)[0][0])
            user_role ← uf.get_row(registered_users, 'username',
username)[0][3]
            if user_role = "Admin" then
                output(f'Selamat datang, Admin {username}!\nMasukkan
command "HELP" untuk daftar command yang dapat kamu panggil.')
            else
                output(f'Selamat datang, Agent {username}!\nMasukkan
command "HELP" untuk daftar command yang dapat kamu panggil.')
            else
                output("Password salah!")
                user_id,user_role ← None,None
        else
            output(f'Login gagal!\nAnda telah login dengan username
{logged_in_user} silahkan lakukan "LOGOUT" sebelum melakukan login kembali.')
            return → logged_in_user, user_id, user_role

```

- **F03 – Logout**

#### **Program Logout**

{Melakukan fungsi logout untuk membantu user keluar dari akun permainan}  
 {I.S. logged\_in\_user terdefinisi}  
 {F.S. logged\_in\_user mengalami perubahan}

#### **REALISASI PROSEDUR**

```
function logout(logged_in_user, user_role, user_oc, user_id): string
```

#### **KAMUS LOKAL**

```
logged_in_user, user_role, user_oc, user_id: string
```

#### **ALGORITMA**

```

    if logged_in_user ≠ None then
        return → None,None,None
    else
        output(f'Logout gagal!\nAnda belum login, silahkan login terlebih
dahulu sebelum melakukan logout.')

```

- **F04 – Menu and Help**

### **Program Menu and Help**

```
{Mengeluarkan output tulisan sebagai petunjuk user jika user belum
melakukan login}
{I.S. user belum melakukan login}
{F.S. output help}
```

### **KAMUS**

```
procedure menu_and_help(user_role, logged_in_user)
user_role, logged_in_user: string
```

### **ALGORITMA PROGRAM UTAMA**

```
    if logged_in_user = None then
        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")
    else
        output("===== HELP =====")
        if user_role.lower() = "admin" then
            output("Selamat datang, Admin",logged_in_user)
            output("Berikut adalah hal-hal yang dapat kamu
lakukan:")
            output("1. Logout: Keluar dari akun yang sedang
digunakan")
            output("2. Shop Management: Melakukan manajemen
pada SHOP sebagai tempat jual beli peralatan Agent")
            output("3. Monster Management : Menambahkan monster
pada database")
            output("4. Save : Menyimpan data game")
            output("5. Exit : Keluar dari game")
            output("Footnote:")
            output("1. Untuk menggunakan aplikasi, masukkan
nama fungsi yang terdaftar")
            output("2. Pastikan memasukkan input yang valid")
        else
            output("Halo Agent",logged_in_user+" .Kamu
memanggil command HELP. Kamu memilih jalan yang benar, semoga")
            output("kamu tidak sesat kemudian. Berikut adalah
hal-hal yang dapat kamu lakukan sekarang:")
```



```

digunakan")
    output("1. Logout : Keluar dari akun yang sedang
yang dimiliki Agent")
    output("2. Inventory : Melihat monsters dan items
list monsters")
    output("3. Owcadex : Membuka OWCADEX untuk melihat
lain")
    output("4. Battle : Melakukan battle dengan monster
monster")
    output("5. Arena : Menginisiasi Arena untuk melatih
items")
    output("6. Shop : Membuka shop untuk membeli
dimiliki")
    output("7. Laboratory : Menaikan level monster yang
    output("8. Jackpot : Melakukan gacha")
    output("9. Save : Menyimpan data game")
    output("10. Exit : Keluar dari game")
    output("Footnote:")
    output("1. Untuk menggunakan aplikasi, masukkan
nama fungsi yang terdaftar")
    output("2. Pastikan memasukkan input yang valid")

```

- **F05 - Monster**

#### **Program Monster**

```
{Melakukan kalkulasi damage serta owcadex}
```

#### **KAMUS**

```

function damage_dealt(atk_power:int): integer
function damage_taken(def_power:int, damage_dealt:int): float
procedure Owcadex (array)

```

#### **REALISASI PROSEDUR**

```
function damage_dealt(atk_power:int): integer
```

#### **KAMUS LOKAL**

```
atk_power: integer
```

#### **ALGORITMA**

```

low_atk ← atk_power * 0.7
high_atk ← atk_power * 1.3
return → randomNumberGenerator (low_atk, high_atk)

```

#### **REALISASI PROSEDUR**

```
function damage_taken(def_power:int, damage_dealt:int): float
```

#### **KAMUS LOKAL**

```
def_power, damage_dealt: integer
```

## ALGORITMA

```
damage_defensed ← damage_dealt * (def_power / 100)
```

```
return → (damage_dealt - damage_defensed)
```

## REALISASI PROSEDUR

```
procedure Owcadex (array)
```

## KAMUS LOKAL

array : array of any

## ALGORITMA

[illegible]





## ALGORITMA PROGRAM UTAMA

### REALISASI PROSEDUR

```
function strength_potion (battle_atk_power, strength_potion_use): integer
{ Melakukan kalkulasi untuk penggunaan strength_potion dalam battle }
{ I.S. atk_power terdefinisi }
{ F.S. atk_power mengalami perubahan sesuai kalkulasi }
```

### KAMUS LOKAL

battle\_atk\_power, strength\_potion\_use: integer

### ALGORITMA

```
if (strength_potion_use > 0) then
    output ("Strength potion hanya bisa digunakan sekali")
    battle_atk_power ← battle_atk_power
else
    battle_atk_power ← battle_atk_power * 1.05
    strength_potion_use += 1
return → (battle_atk_power, strength_potion_use)
```

### REALISASI PROSEDUR

```
function resilience_potion (battle_def_power, resilience_potion_use):
integer
{ Melakukan kalkulasi untuk penggunaan resilience_potion dalam battle }
{ I.S. def_power terdefinisi }
{ F.S. def_power mengalami perubahan sesuai kalkulasi }
```

### KAMUS LOKAL

battle\_def\_power, resilience\_potion\_use: integer

### ALGORITMA

```
if (resilience_potion_use > 0) then
    output ("Resilience potion hanya boleh digunakan sekali")
    battle_def_power ← battle_def_power
else
    battle_def_power ← battle_def_power * 1.05
    resilience_potion_use += 1
return → (battle_def_power, resilience_potion_use)
```

### REALISASI PROSEDUR

```
function healing_potion (battle_current_hp, max_hp, healing_potion_use):
integer
{ Melakukan kalkulasi untuk penggunaan health_potion dalam battle }
{ I.S. health terdefinisi }
{ F.S. health mengalami perubahan sesuai kalkulasi }
```

### KAMUS LOKAL

battle\_current\_hp, max\_hp, healing\_potion\_use: integer

### ALGORITMA

```
healing ← battle_current_hp * 1.25
if (healing_potion_use > 0) then
    output ("Healing potion hanya boleh digunakan sekali")
    battle_current_hp ← battle_current_hp
else
    if (healing ≥ max_hp) then
        battle_current_hp ← max_hp
    else
        battle_current_hp ← healing
return → (battle_current_hp, healing_potion_use)
```

### • F07 - Inventory

#### Program Inventory

{ Melakukan fungsi utama inventory dengan menunjukkan monster serta item yang dimiliki user }

{ I.S. user\_id serta current\_inventory terdefinisi }

{ F.S. output current\_inventory terdefinisi }

### KAMUS LOKAL

procedure take\_monster\_id\_level (input user\_id : integer)

procedure base\_monster\_in\_inventory (input monster\_id : integer)

function level\_calculations (base\_monster : array of any ) → array of any

procedure take\_item\_id (input user\_id : integer)

procedure display\_inventory (input user\_id : integer)

count\_monster, count\_item : integer

inventory\_command : string

### ALGORITMA

display\_inventory(user\_id)

count\_monster ← integer(len(current\_inventory\_monster))

```

count_item ← integer(len(current_inventory_item))

output("Ketikkan id untuk menampilkan id item :")

(inventory_command)

while (inventory_command ≠ "KELUAR") do

    if (1 ≤ int(inventory_command) ≤ (count_monster + count_item)) then

        if(1 ≤ int(inventory_command) ≤ count_monster) then

            take_monster ←
current_inventory_monster[(inventory_command)-1]

            output("Monster")

            output("Name          :",take_monster[1])

            output("ATK Power       :",take_monster[2])

            output("DEF Power        :",take_monster[3])

            output("HP              :",take_monster[4])

            output("Level           :",take_monster[5])

        else

            none

        if(count_monster < int(inventory_command) ≤ (count_monster + count_item)) then

            take_item ← current_inventory_item[(inventory_command)-1-
count_monster]

            if(take_item[1]="Monster Ball") then

                output("Monster Ball")

                output("Quantity :",take_item[2])

            else

                output("Potion")

                output("Type          :",take_item[1])

                output("Quantity    :",take_item[2])

            output("Ketikkan id untuk menampilkan id item :")

```

```

        input(inventory_command)

    else

        output("Command diluar range, silahkan masukan range
valid")

        output("Ketikkan id untuk menampilkan id item :")

        input(inventory_command)

```

#### **REALISASI PROSEDUR**

```

procedure take_monster_id_level (input user_id : integer)

{ Melakukan pengambilan monster_id serta level dari monster.csv dan
user.csv }

{ I.S. array monster_id dan monster_level terdefinisi }

{ F.S. array monster_id mengalami modifikasi sesuai program }

```

#### **KAMUS LOKAL**

```

user_id : integer

monster_id, monster_level : array of integer

```

#### **ALGORITMA**

```

monster_id ← []

monster_level ← []

i traversal [0 .. len(monster_inventory)]

    if (user_id = (monster_inventory[1+i][0])) then

        monster_id ← (monster_inventory[1+i][1])

        monster_level ← (monster_inventory[1+i][2])

monster_id ← take_monster_id_level(user_id)[0]

```

#### **REALISASI PROSEDUR**

```

procedure base_monster_in_inventory (input monster_id : integer)

{ Mengambil data base monster sesuai dengan id yang dimiliki tanpa
perubahan stats monster }

```



```
{ I.S. monster_id terdefinisi dan array base_monster terdefinisi }
{ F.S. array base_monster mengalami modifikasi sesuai dengan program }
```

#### **KAMUS LOKAL**

```
monster_id : integer
monster_level : array of integer
base_monster : array of any
```

#### **ALGORITMA**

```
monster_level ← take_monster_id_level(user_id)[1]
base_monster ← []
  i traversal [0 .. len(monster_id)]
    j traversal [0 .. len(monster_array)-1]:
      if (int(monster_id[i]) =
int(monster_array[1+j][0])) then
        base_monster ←
(monster_array[1+j]+list(monster_level[i]))
      else
        none
```

#### **REALISASI PROSEDUR**

```
function level_calculations (base_monster : array of any ) → array of any
{ Melakukan kalkulasi dari monster di inventory sesuai dengan level monster
tersebut }
{ I.S. array base_monster terdefinisi }
{ F.S. array monster_in_inventory terdefinisi }
```

#### **KAMUS LOKAL**

```
base_monster, monster_in_inventory : array of any
```

#### **ALGORITMA**

```
monster_in_inventory ← base_monster
```

```

i traversal [0 .. len(base_monster)]

    if (int(monster_in_inventory[i][5] ≠ 1)) then

        monster_in_inventory[i][2] ←
float(monster_in_inventory[i][2]) + (float(monster_in_inventory[i][2]) *
(float(monster_in_inventory[i][5RE])*0.1))

        monster_in_inventory[i][3] ← float(monster_in_inventory[i][3])
+ (float(monster_in_inventory[i][3]) *
(float(monster_in_inventory[i][5])*0.1))

        monster_in_inventory[i][4] ← float(monster_in_inventory[i][4])
+ (float(monster_in_inventory[i][4]) *
(float(monster_in_inventory[i][5])*0.1))

current_inventory_monster ← level_calculation(base_monster)

```

#### **REALISASI PROSEDUR**

```

procedure take_item_id (input user_id : integer)

{ Mengambil list dari item yang dimiliki user sesuai user_id }

{ I.S. user_id terdefinisi dan array item_id terdefinisi }

{ F.S. array current_inventory_item terdefinisi }

```

#### **KAMUS LOKAL**

```

user_id : integer

item_id : array of integer

item_inventory : array of any

```

#### **ALGORITMA**

```

item_id ← []

i traversal [0 .. len(item_inventory)-1]

    if (user_id = (item_inventory[1+i][0]))then

        item_id ← (item_inventory[1+i])

current_inventory_item ← take_item_id(user_id)

```

#### **REALISASI PROSEDUR**

```

procedure display_inventory (input user_id : integer)

```

{ Melakukan fungsi utama fitur display dari inventory yang dimiliki user }

{ I.S. user\_id, array current\_inventory\_monster, array  
current\_inventory\_item terdefinisi }

{ F.S. output dari display inventory terdefinisi }

#### **KAMUS LOKAL**

user\_id, count, owca\_coin : integer

current\_inventory\_monster, current\_inventory\_item : array of any

#### **ALGORITMA**

count  $\leftarrow$  0

output("===== INVENTORY LIST (User ID :",(user\_id)+") =====")

output("Jumlah O.W.C.A. Coin-mu sekarang",owca\_coin)

i traversal [0 .. len(current\_inventory\_monster)]

output((i+1)+". Monster  
(Name:",current\_inventory\_monster[0+i][1],"",  
Lvl:",current\_inventory\_monster[0+i][5],"",  
HP:",(current\_inventory\_monster[0+i][4])+")")

        count  $\leftarrow$  count + 1

j traversal [0.. len(current\_inventory\_item)]

    if (current\_inventory\_item[0+j][1] = "Monster Ball") then

output((count+1+j)+". Monster Ball  
(Qty:",current\_inventory\_item[0+j][2])+")")

else

output((count+1+j)+". Potion  
(Type:",current\_inventory\_item[0+j][1],"",  
Qty:",(current\_inventory\_item[0+j][2])+")")

#### **• F08 - Battle**

Program Battle

{ Melakukan fitur battle pada game serta semua function dan prosedur yang diperlukan dalam battle }

#### **KAMUS**

```

procedure select_available_monster (input user_monsters : list)

procedure print_status (input turns : integer, input attacker_monster,
defender_monster : list)

procedure print_potion (input user_inventory : list)

procedure random_monster_data (output monster, level_monster : list)

function duel (input battle_status, user_monster, enemy_monster : list,
input oc_reward : integer)

procedure drink_potion (input user_item_inventory : list, input/output
potion_used : integer, user_monster : list, hp_max : integer)

user_data, item_inventory, monster_inventory, monster,
user_monster_inventory, user_item_inventory, user_monster,
enemy_monster_tables, battle_status, daftar_perintah : list

enemy_level, hp_max, turns : integer

isFighting : boolean

perintah : string

```

#### **ALGORITMA UTAMA**

```

    if (user_role ≠ "agent" ) then
        return → ("Ini area bertarung agen!")
    user ← database ["user"]
    item_inventory ← database["item_inventory"]
    monster_inventory ← database["monster_inventory"]
    monster ← database["monster"]
    enemy_level ← randomNumberGenerator(1,5)
    user_monster_inventory ←
get_user_inventory(monster_inventory,user_id)
    user_item_inventory ← get_user_inventory(monster,item_inventory)
    user_monsters ←
level_calculation(merge_tables_by_id(monster,user_monster_inventory))
    enemy_monsters_tables ←
level_calculation([random_monster_data(monster,enemy_level)])
    enemy_monster ← enemy_monsters_tables[0]
    print_stat_awal(monster=enemy_monster,graphics=graphics,text=
f"ROARRR {enemy_monster[1]} telah muncul")
    isFighting, user_monster ← select_available_monster(user_monsters)
    hp_max ← integer(user_monster[4])

```

```

turns ← 1
potion_used ← []
print_stat_awal(monster=user_monster,graphics=graphics,text=f"HUFTTT
Agen telah mengeluarkan monster {user_monster[1]}")
battle_status ← [turns,isFighting,False,0,0,0]
while (isFighting) do
    input(perintah)
    daftar_perintah ← ["1","2","3","4"]
    while (perintah not in daftar_perintah) do
        output("Fitur perintah tersebut belum ada")
        input(perintah)
    depend on (perintah)
        perintah = "2" : print_potion(user_inventory =
user_item_inventory)
        drink_potion(user_item_inventory=user_item_inventory,potion_use
d=potion_used,user_monster=user_monster,hp_max=hp_max)
        perintah = "3" : index_monsterball ←
get_index_by_id(user_id,user_item_inventory)
        get_monster ← False
        if index_monsterball ≠ -999 then
            output("Anda tidak mempunyai monsterball")
        else
            get_monster ← MonsterBall(enemy_monster)
        if get_monster then
            user_item_inventory[index_monsterball][2] -= 1

            update_inventory_by_id(item_inventory,user_item_inventory
,user_id)
            isFighting ← False
            index_in_inventory ←
get_index_by_id(enemy_monster[0],user_monster_inventory)
            if index_in_inventory ≠ -999 then
                #
                convert_to_oc ← randomNumberGenerator(0,100)
                output(f"Sayangnya monster sudah terdapat pada
inventory, sehingga monster diconvert menjadi {convert_to_oc}")

```

```

        battle_status[5] += convert_to_oc
    else
        new_monster ← [user_id] + [enemy_monster[0]]
+ [enemy_monster[5]]
        monster_inventory.append(new_monster)
        perintah == "4" : isFighting ← False
        output("Selamat berjumpa lagi di battle selanjutnya")
        if isFighting then
            oc_reward ← randomNumberGenerator(0,50)

        duel(battle_status,user_monster,enemy_monster,oc_reward)
            turns,isFighting ←
battle_status[0],battle_status[1]
            win ← battle_status[2]
            if win and not isFighting then
                output(f"Keren monster {enemy_monster[1]} berhasil
anda bunuh ")
            else if not win and not isFighting then
                output("Anda kalah, mungkin next time ,asahlah
skill hoki di arena dulu dek")

        output(f"Total OC yang berhasil anda dapatkan : {battle_status[5]}")
        index_user ← get_index_by_id(user_id,user)
        user[index_user][4] +=battle_status[5]

```

#### **REALISASI PROSEDUR**

```
function MonsterBall (enemy_array:list)->bool
```

#### **KAMUS LOKAL**

```
enemy_array:list
```

#### **ALGORITMA**

```
success ← False
```

```
level ← enemy_array[5]
```

```
number ← randomNumberGenerator(0,100)
```

```
if(level = 1)then
```

```

    if(number ≤ 75)then
        success ← True
if(level = 2)then
    if(number ≤ 50)then
        success ← True
if(level = 3)then
    if(number ≤ 25)then
        success ← True
if(level = 4)then
    if(number ≤ 10)then
        success ← True
if(level = 5)then
    if(number ≤ 5)then
        success ← True
if (success = True)then
    output("Swoosshhhh, Anda mengeluarkan Monster Ball !!!")
    output("Selamat, Anda berhasil mendapatkan Monster",enemy_array[1])
    output("Name          :",enemy_array[1])
    output("ATK Power       :",enemy_array[2])
    output("DEF Power       :",enemy_array[3])
    output("HP              :",enemy_array[4])
    output("Level           :",enemy_array[5])
else
    output("Swoosshhhh, Anda mengeluarkan Monster Ball !!!")
    output("Yahhh, Anda belum berhasil mendapatkan monster Zuko !!!")
return → success

```

#### **REALISASI PROSEDUR**

```
procedure print_stat_awal(monster:list,graphics:list,text:str): string
```

#### **KAMUS LOKAL**

```
monster:list,graphics:list,text:str
```

#### **ALGORITMA**

```

random_index ← randomNumberGenerator(0,len(graphics))
output(graphics[random_index])
output(text)
output(f"Monster Type    : {monster[1]}")
output(f"Attack Power    : {int(monster[2])}",)
output(f"Defense Power   : {int(monster[3])}",)
output(f"Monster Health  : {int(monster[4])}",)
output(f"Level : {monster[5]}")
output("=====")

```

### REALISASI PROSEDUR

```

procedure select_available_monster (input user_monsters : list)
{Melakukan fitur untuk melakukan select monster yang dipake pada battle}
{ I.S. user_monsters terdefinisi }
{ F.S. user_monster termodifikasi sesuai program }

```

### KAMUS LOKAL

```

isFighting : boolean
user_monster : list
urutan : integer

```

### ALGORITMA

```

isFighting ← True
user_monster ← [ ]
urutan ← 1
output("=====MONSTER LIST=====")
i traversal [1..len(user_monsters)]
    output("{urutan}. {user_monsters[i][1]} (level {user_monsters[i][5]})")
    urutan ← urutan + 1
(index_monster)
attempts_id ← 10

```

```

while (index_monster ≥ len(user_monsters) or index_monster < 1 or not
is_all_same_type(index_monster,check_integer=True) do

```



```

    output("Monster tidak ada, Masukkan pilihan monster yang ada !")
    input(index_monster)
    attempts_id -= 1
if attempts_id = 0 then
    isFighting ← False
    output("Sepertinya anda belum memiliki monster, silahkan beli di shop terlebih dahulu")
else
    user_monster ← user_monsters[index_monster]

```

### **REALISASI PROSEDUR**

```
function input_perintah(turns,user_monster) : string
```

### **KAMUS LOKAL**

```
turns,user_monster : string
```

### **ALGORITMA**

```

output(f"===== Turns {turns} ({user_monster[1]})=====")
output("1. Attack")
output("2. Use Potion")
output("3. Monsterball")
output("4. Quit")
perintah ← str(input("Pilih perintah : "))
return → perintah

```

### **REALISASI PROSEDUR**

```

procedure print_status
(turns:int,attacker_monster:list,defender_monster:list,attacker:list)
{ Melakukan print dari status monster }
{ I.S. turns, attacker_monster, defender_monster terdefinisi }
{ F.S. print status terdefinisi }

```

### **KAMUS LOKAL**

```
turns : integer
```

```
attacker_monster, defender_monster,attacker : array of any
```

### ALGORITMA

```
if (defender_monster[4] ≤ 0 ) then  
    defender_monster[4] ← 0  
if attacker = "enemy" then  
    output(f"=====Turns {turns}({attacker_monster[1]})  
=====")  
output("Monster Type    : {defender_monster[1]}")  
output("Attack Power    : {defender_monster[2]}",)  
output("Defense Power   : {defender_monster[3]}",)  
output("Monster Health  : {defender_monster[4]}",)  
output("Level : {defender_monster[5]}")  
output("=====")
```

### REALISASI PROSEDUR

```
procedure print_potion (user_inventory)  
{ Melakukan fungsi jika user meminum potion }  
{ I.S. potion_arr terdefinisi }  
{ F.S. potion_quantity terdefinisi }
```

### KAMUS LOKAL

user\_inventory : array of any

### ALGORITMA

```
potion_arr ← ["strength","resilience","healing"]  
potion_quantity ← []  
for potion in potion_arr :  
    found ← False  
    for row in user_inventory :  
        if (potion = row[0]) then  
            potion_quantity.append (row[1])  
            found ← True  
            break  
  
    if (not found) then
```

```

        potion_quantity.append("0")

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

        output(f"1.Strength Potion (Quantity : {potion_quantity[0]})----
Increase ATK Power")

        output(f"2.Resilience Potion (Quantity : {potion_quantity[1]})----
Increase DEF power")

        output(f"3.Healing  Potion (Quantity : {potion_quantity[2]})---
Restore Health")

        output(f"4.Cancel")

        output("=====")

```

#### **REALISASI PROSEDUR**

```

procedure random_monster_data (input monster : list, input level_monster :
integer, output monster_enemy : list)

```

```

{ Mengambil monster secara random dari data }
{ I.S. monster dan level_monster terdefinisi }
{ F.S. monster_enemy terdefinisi }

```

#### **KAMUS LOKAL**

```

monster : integer
level_monster : array of integer
index_monster : integer
monster_enemy : array of any

```

#### **ALGORITMA**

```

index_monster ← randomNumberGenerator(1,len(monster))
monster_enemy ← monster[index_monster] + [str(level_monster)]
return → (monster_enemy)

```

#### **REALISASI PROSEDUR**

```

function duel (input battle_status, user_monster, enemy_monster : list,
input oc_reward : integer)
{ Melakukan fungsi utama duel }
{ I.S. database dan user_data terdefinisi }
{ F.S. oc dan user_data mengalami modifikasi sesuai program }

```

#### **KAMUS LOKAL**

```

damage_dealt_user, damage_taken_enemy, oc_reward : integer
battle_status, user_monster, enemy_monster : array of any

```

#### ALGORITMA

```

    damage_dealth_user ← damage_dealt(atk_power=user_monster[2])
    damage_taken_enemy ←
damage_taken(def_power=enemy_monster[3],damage_dealt=damage_dealth_user)
    enemy_monster[4] ← integer(user_monster[4]-damage_taken_enemy)

    print_status(turns=battle_status[0],attacker_monster =
user_monster,defender_monster=enemy_monster)

    time.sleep(0.5)

    damage_dealth_enemy ← damage_dealt(atk_power=enemy_monster[2])
    damage_taken_user ←
damage_taken(def_power=user_monster[3],damage_dealt=damage_dealth_enemy)
    user_monster[4] ← int(user_monster[4]-damage_taken_user)


    depend on (enemy_monster, user_monster)

    enemy_monster[4] ≤ 0 :battle_status[1] False

                                battle_status[2] True

                                battle_status[5] += oc_reward

    user_monster[4] ≤ 0:
print_status(turns=battle_status[0],attacker_monster=enemy_monster,defender
_monster=user_monster,attacker="enemy")

                                battle_status[1] ← False
                                battle_status[2] ← False

    else

        battle_status[1] ← True
        battle_status[2] ← False

        output(f"WH000ZZZ Monster {enemy_monster[1]} menyerang balik
anda")

        print_status(turns=battle_status[0],attacker_monster=enemy_mons
ter,defender_monster=user_monster)


    battle_status[0] += 1

    battle_status[3] += damage_dealth_user

    battle_status[4] += damage_taken_user

```

## REALISASI PROSEDUR

```
procedure drink_potion (input user_item_inventory : list, input/output  
potion_used : integer, user_monster : list, hp_max : integer)
```

```
{ Melakukan fungsi potion jika user memilih potion }
```

```
{ I.S. potion_arr terdefinisi }
```

```
{ F.S. user_monster mengalami modifikasi sesuai program }
```

## KAMUS LOKAL

```
potion_arr, daftar_id, user_item_inventory, user_monster : list
```

```
drink_potion, cancel, used : boolean
```

```
potion_used, hp_max, id_potion, index_potion_in_inventory, current_stock :  
integer
```

```
type_potion : string
```

## ALGORITMA

```
    potion_arr ← ["strength","resilience","healing"]  
    daftar_id ← ["1","2","3","4"]  
    drink_potion ← False  
    input(id_potion)  
    while (id_potion not in daftar_id) do  
        output("Input salah")  
        input(id_potion)  
    if (id_potion = "4") then  
        output("Engga jadi beli")  
        cancel ← True  
        index_potion_in_inventory ← -999  
    else  
        type_potion ← potion_arr[int(id_potion)-1]  
        index_potion_in_inventory ←  
get_index_by_id(type_potion,user_item_inventory)  
        cancel ← False  
    if (not cancel and (type_potion in potion_used)) then  
        cancel ← True  
        depend on (type_potion)
```

```

    type_potion = "stregth" : output("Tadi udah minum jamu
kuat")

    type_potion = "resilience" : output("Udah minum bro")
else
    output("Tadi udah minum teh healing ini")

    if (index_potion_in_inventory ≠ -999 and not cancel) then
        current_stock ←
int(user_item_inventory[index_potion_in_inventory][1] )
        drink_potion ← False
        if (current_stock = "0") then
            kosong ← True
        else
            user_item_inventory[index_potion_in_inventory][1] ←
current_stock-1
            kosong ← False
            drink_potion ← True
        else if (index_potion_in_inventory = -999 and not cancel)then
            output (f"anda tidak memiliki potion {type_potion}")
            kosong ← True

    if (not cancel and drink_potion) then
        potion_used.append(type_potion)
        depend on (type_potion)

        type_potion = "strength" and not kosong : new_atk ←
int(user_monster[2] *1.05 )
        user_monster[2] ← new_atk
        output(f"Setelah meminum ruan ini atk power meningkat
menjadi {new_atk}")

        type_potion = "resilience" and not kosong : new_def ←
int(user_monster[3]*1.05)
        user_monster[3] ← new_def
        output(f"Setelah meminum ramuan ini, def power meningkat
menjadi {new_def}")

```

```

        type_potion = "healing" and not kosong : user_monster[4]
← int(user_monster[4]*1.25)

        if user_monster[4] > hp_max then
            user_monster[4] ← hp_max

            output("Nyawa musuh telah disembuhkan menjadi
{user_monster[4]}")

```

- **F09 – Arena**

PROGRAM ARENA

```

function arena (database, user_data : array of any) → array of any

{ Melakukan fungsi utama arena sebagai salah satu fitur dalam game }

{ I.S. user_data terdefinisi }

{ F.S. function arena terdefinisi }

```

**KAMUS**

database, user\_data, user\_monster\_inventory, user\_item\_inventory\_  
user\_monster\_, enemy\_monster\_tables, potion\_used, battle\_status : array of  
any

item\_inventory, monster\_inventory, monster : string

stage, oc\_awal, hp\_max, turns : integer

win\_battle : Boolean

oc\_rewards : array of integer

**ALGORITMA**

```

    user_data ← []

    if (user_data[3] ≠ "agent" ) then

        output("Ini area bertarung agen!")

    item_inventory ← database["item_inventory"]

    monster_inventory ← database["monster_inventory"]

```

```

monster ← database["monster"]

stage ← 1

user_monster_inventory ←
get_user_inventory(monster_inventory,user_data[0])

user_item_inventory ← get_user_inventory(item_inventory,user_data[0])

user_monsters ←
level_calculation(merge_tables_by_id(monster,user_monster_inventory))

enemy_monsters_tables ←
level_calculation([random_monster_data(monster,stage)])

enemy_monster ← enemy_monsters_tables[0]


Arena, user_monster ← select_available_monster(user_monsters)

oc_awal ← integer(user_data[4])

hp_max ← integer(user_monster[4])

potion_used ← []

turns ← 1

win_battle ← False

oc_rewards ← [30,50,100,120,150]

battle_status ← [turns,Arena,win_battle,0,0,0]

while (Arena) do

    output(enemy_monster)

    perintah ← (input("Pilih perintah : "))

    daftar_perintah ← ["1","2","3"]

    while (perintah not in daftar_perintah) do

        output("Fitur perintah tersebut belum ada")

        perintah ← (input("Pilih perintah : "))

    depend on (perintah)

```



```

(perintah = "1" ) :

    oc_reward ← oc_rewards[stage-1]

    duel(battle_status,user_monster,enemy_monster
    ,oc_reward)

    turns,Arena,win_battle ←
    battle_status[0],battle_status[1],battle_status[2]

    depend on (win_battle and stage)

(win_battle and stage ≠ 5) :

    Arena ← True

    new_monster ←
level_calculation([random_monster_data(monster,stage)])

    user_monster ← new_monster[0]

    output("Keren euy anda
mengalahkan monster {enemy_monster[1]} ")

    output("Anda mendapatkan
{oc_reward} pada stage ini")

    output("STAGE SELANJUTNYA : ")

    potion_used ← []

    turns ← 1

    stage ← stage + 1

    win_battle ← False

    battle_status[2] ← win_battle

    output("=====STAGES
{stage}=====")

(win_battle and stage = 5 ) :

    Arena ← False

    output("=====STAGES
{stage}=====")

```

```

                                output("anda telah mengalahkan
final boss(monster {enemy_monster[1]}))")

                                output("Anda mendapatkan
{oc_reward} pada stage ini")

                                (not win_battle and not Arena ) :

                                stage ← stage - 1

                                output("MONSTER ANDA WAFATTT
,TETAP SEMANGAT, ANDA KEREN TELAH BERHASIL MEMBANTAI HINGGA STAGES
{stage}")

                                (perintah = "2") :

                                print_potion(user_inventory=user_item_inventory)

drink_potion(user_item_inventory=user_item_inventory,potion_used=potion_use
d,user_monster=user_monster,hp_max=hp_max)

                                else

                                Arena ← False

                                output("Pergi dari arena")

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

                                output("Total hadiah : {oc_awal-battle_status[5]}")

                                output("Jumlah stage : {stage}")

                                output("Damage diberikan : {battle_status[3]}")

                                output("Damage diterima : {int(battle_status[4])}")

                                user_data[4] ← string(integer(user_data[4]) + battle_status[5])

                                update_inventory_by_id(item_inventory,user_item_inventory,user_data[0
])

```

- **F10 - Shop & Currency**

Program Shop & Currency

{Fitur ini membantu user untuk membeli barang-barang tambahan}

```
{I.S. user_inventory, oc diambil dari database}
{F.S. penambahan pada user_inventory dan pengurangan OC berdasarkan
pembelian user}
```

#### KAMUS

```
procedure shop(database,user_data)
Database, user_monster, user_items : array of any
User_id, user_role : string
function beli(objek:str,data_shop,user_inventory,user_data,menu)
```

#### ALGORITMA UTAMA

```
{ Inisialisasi data dari database }
monster ← database["monster"]
monster_shop ← database["monster_shop"]
monster_inventory ← database["monster_inventory"]
item_shop ← database["item_shop"]
item_inventory ← database["item_inventory"]
user_data ← ['5', 'Kenny_agen_rahasia', 'kribogeming55', 'agent',
'6699'] { sample , nanti diadjust lagi }
user_id ← user_data[0]
user_role ← user_data[3]
user_items ← get_user_inventory(item_inventory,user_id)
user_monsters ← get_user_inventory(monster_inventory,user_id)

if user_role ≠ "agent" then {Shop untuk agent?}
    return → output("Ini shop untuk role agent")

output("Selamat datang di indoseptember, Selamat belanja :) \n")
is_Shopping ← True
while is_Shopping do
    menu_potion ← generate_id_potion(item_shop)
    menu_monster ← merge_tables_by_id(monster,monster_shop)
    aksi ← str(input("Pilih aksi (lihat/beli/keluar) \n:")).lower()
    depend on (aksi)
    aksi ≠ "lihat" and aksi ≠ "beli" and aksi ≠ "keluar" : { cek input
        output ("Aksi yang anda berikan tidak valid, silahkan
input lagi ") }
    aksi = "lihat" :
        objek ← str(input(("Mau lihat apa (monster/potion)") )).lower()
        depend on (objek)
        objek = "monster" : table_interface(menu_monster)
        objek = "potion" : table_interface(menu_potion)
    else
        output("Input tidak valid")

    aksi = "beli" :
```

```

        objek ← str(input(("Mau beli apa
(monster/potion)\n:"))).lower()
        output(f"Jumlah O.W.C.A Coin mu sekarang {user_data[4]}\n")
        depend on (objek)
            objek = "monster":
                beli(objek="monster",data_shop=monster_shop,user_inventor
y=user_monsters,user_data=user_data,menu=menu_monster)
                update_inventory_by_id(monster_inventory,user_monsters,us
er_id) { perbarui database monster_inventory }
                objek = "potion" then
                beli(objek="potion",data_shop=item_shop,user_inventory=us
er_items,user_data=user_data,menu=menu_potion)
                update_inventory_by_id(item_inventory,user_items,user_id)
{ perbarui database item_inventory }
            else
                output("Input tidak valid
")
        else { aksi == keluar }
        is_Shopping ← False
        output("Terimakasih sudah berbelanja")

```

#### REALISASI PROSEDUR

```

function beli(objek:str,data_shop,user_inventory,user_data,menu): string
{ I.S. user_inventory, oc terdefinisi dari database }
{ F.S. user_inventory, oc mengalami modifikasi sesuai program }

```

#### KAMUS LOKAL

objek:str,data\_shop,user\_inventory,user\_data,menu : string

#### ALGORITMA

```

        oc ← int(user_data[4])
        could_purchase ← True
        id ← (input(f"Masukkan id {objek} : "))
        index_in_shop ← get_index_by_id(id,menu)
        depend on (objek)
            objek = "potion" : index_in_inventory ←
get_index_by_id(id,generate_id_potion(user_inventory))
            objek = "monster" : index_in_inventory ←
get_index_by_id(id,user_inventory)
        else
            output("Fitur belum ada")
            could_purchase ← False
        depend on (index_in_shop,index_in_inventory)
            index_in_shop = -999 : { i == -999 berarti index tidak ada
}
            output(f"{objek} tidak terdapat pada shop")
            could_purchase ← False
        index_in_inventory ≠ -999 and objek ="monster" :

```

```

        output(f"Monster {menu[index_in_shop][1]} sudah terdapat
di inventory")
        could_purchase ← False
    if could_purchase then { purchase objek (kondisi telah tervalidasi) }
        stock ← int(data_shop[index_in_shop][1])
        price ← int(data_shop[index_in_shop][2])
        purchase ← 1
        if objek = "potion" then
            purchase ← int(input("Mau beli berapa banyak potion? : "))
            total_price ← price * purchase
            if purchase > stock then
                output("Stock tidak cukup")
            else if total_price > oc then
                output("Saldo oc tidak cukup")
            else { stock tersedia dan saldo cukup }
                oc -= total_price
                user_data[4] ← str(oc)
                data_shop[index_in_shop][1] ← str(stock-purchase)
                nama_objek ← menu[index_in_shop][1]
                output(f"{objek} {nama_objek} berhasil dibeli dengan harga
{total_price}")
            if objek = "monster" then
                user_inventory.append([id,str(purchase)])
            else if (objek = "potion" and index_in_inventory = -999)
then
                user_inventory.append([nama_objek,str(purchase)])
            else { objek == "potion" dan potion ada di inventory }
                current_amount ← user_inventory[index_in_inventory][1]
                new_amount ← purchase + int(current_amount)
                user_inventory[index_in_inventory][1] ← str(new_amount)
            output(user_inventory)

```

### • F11 - Laboratory

PROGRAM LABORATORY

```

function laboratory (database, user_data : array of any) array of any
{ Fungsi utama laboratory untuk melakukan upgrade level pada monster yang
dimiliki user}
{ I.S. database serta user_data terdefinisi }
{ F.S. monster_inventory mengalami modifiskasi sesuai program }

```

### KAMUS

```

function update_inventory_by_id (general_inventory, user_inventory : array
of any, user_id : string) array of any
function get_user_inventory (inventory : array of any, user_id : string )
array of any

```

```

function get_monster_name (user_monster_inventory, table_monster : array of
any)array of any
procedure print_monster_list (input user_monster, monster_names : array of
any)
procedure print_upgrade_price (input upgrade_prices : array of integer)
function upgrading_monster (input monster_level, oc : integer, input
monster_name : string, input upgrade_prices : array of integer)
database, user_data, monster_inventory, monster, user_monster,
monster_names = array of any
upgrade_prices : array of integer
user_id, oc, banyak_monster, index_monster : integer
isLaborating : boolean

```

#### ALGORITMA UTAMA

```

monster_inventory ← database["monster_inventory"]
monster ← database["monster"]
upgrade_prices ← ["300","500","800","1000"]
user_data ← []
user_id ← user_data[0]
oc ← user_data[4]
user_monsters ← get_user_inventory(monster_inventory,user_id)
monster_names ← get_monster_name(user_monsters,monster)
output("SELAMAT DATANG DI LAB DR ASEP")
isLaborating ← True
while (isLaborating = True) do
    print_monster_list(user_monsters,monster_names)
    print_upgrade_price(upgrade_prices)
    banyak_monster ← len(monster_names)-1
    input(index_monster)
    while (not is_range_valid((index_monster),banyak_monster+1) or not
is_all_same_type((index_monster),check_integer=True)) do
        index_monster ←
input_un till_type_valid(s(index_monster),is_all_same_type, message="Input
harus berupa integer",type="integer")
        index_monster ←
input_un till_range_valid(index_monster,len(monster_names),f"Range pilihan
tidak valid \nPilih monster (1-{len(monster_names)}: )")
        index_monster ← (index_monster)
        monster_level ← user_monsters[index_monster][1]
        monster_name ← monster_names[index_monster]
        monster_level,oc ←
upgrading_monster(int(monster_level),monster_name,int(oc),upgrade_prices)
        user_monsters[index_monster][1] ← monster_level
        next_monster ← input("Ingin mengupgrade monster lain (y/n)?")
        if (next_monster = "n" ) then
            user_data[4] ← oc
            update_inventory_by_id(monster_inventory,user_monsters,user_id)

```

```

        isLaboratoring ← False
    else
        isLaboratoring ← True

```

#### REALISASI PROSEDUR

```

function update_inventory_by_id (general_inventory, user_inventory : array
of any, user_id : string) array of any
{ Melakukan penambahan row dari user_id dengan data user_inventory }
{ I.S general_inventory dari database(general_inventory), user_inventory
telah update }
{ F.S Jika user_id terdapat pada general_inventory, maka row pada user_id
akan diupdate dengan data user_inventory }

```

#### KAMUS LOKAL

```

general_inventory, user_inventory, : array of any
user_id, new_data : string
found : boolean
user_row, general_row: integer

```

#### ALGORITMA

```

j traversal [1 .. len(user_inventory)]
    found ← False
    user_row ← user_inventory[j]
    i traversal [1 .. len(general_inventory)]
        general_row ← general_inventory[i]
        if (general_row[0] = user_id) and (general_row[1] = user_row[0])
then
        general_inventory[i] ← [user_id] + user_row
        found ← True
        break
    if (found = True) then
        new_data ← [user_id] + user_row
        general_inventory ← (new_data)

```

#### REALISASI PROSEDUR

```

function get_user_inventory (inventory : array of any, user_id : string )
array of any
{ Fungsi mengeluarkan output data user dari database inventory yang
memiliki user_id }
{ I.S. inventory dan user_id terdefinisi }
{ F.S. user_inventory terdefinisi }

```

#### KAMUS LOKAL

```

Inventory : array of any
User_id : string

```

#### ALGORITMA

```

user_inventory ← [inventory[0][1:]]
row traversal [0 .. inventory[1:]]
    if row[0] = user_id then
        user_inventory ← (row[1:])
    else
        none

```

#### REALISASI PROSEDUR

```

function get_monster_name (user_monster_inventory, table_monster : array of
any)array of any
{ Fungsi untuk mendapatkan nama monster yang terdapat pada
user_monster_inventory }
{ I.S. user_monster_inventory dan table_monster terdefinisi }
{ F.S. monster_names terdefinisi }

```

#### KAMUS LOKAL

```

user_monster_inventory, table_monster : array of any
monster_names : array of strings
id_monster, j : integer
found : boolean

```

#### ALGORITMA

```

monster_names ← ["Monster Names : "]
i traversal [1 . . len(user_monster_inventory)]
    id_monster ← user_monster_inventory[i][0]
    found ← False
    j ← 0
    while (not found) do
        if (id_monster = table_monster[j][0]) then
            monster_name ← table_monster[j][1]
            monster_names ← (monster_name)
            found ← True
        else
            none
        j ← j + 1

```

#### REALISASI PROSEDUR

```

procedure print_monster_list (input user_monster, monster_names : array of
any)
{ Procedure untuk mencetak list monster }
{ I.S. user_monster dan monster_names terdefinisi }
{ F.S. monster list tercetak }

```

#### KAMUS LOKAL

```

User_monster : array of any
Monster_names : array of string

```



### ALGORITMA

```
output("===== MONSTER LIST =====")
i traversal [1 .. len(user_monsters)]
    output("{i}. {monster_names[i]} (Level : {user_monsters[i][1]})")
output(" ")
```

### REALISASI PROSEDUR

```
procedure print_upgrade_price (input upgrade_prices : array of integer)
{ Procedure untuk mencetak harga dari upgrade level }
{ I.S. upgrade_prices terdefinisi }
{ F.S. harga upgrade_level tercetak }
```

### KAMUS LOKAL

upgrade\_prices : array of integer

### ALGORITMA

```
i traversal [0 .. 4]
    output("Level {i+1} -> Level{i+2} : {upgrade_prices[i]}")
output(" ")
```

### REALISASI PROSEDUR

```
function upgrading_monster (input monster_level, oc : integer, input
monster_name : string, input upgrade_prices : array of integer)
{ I.S monster level belum terupgrade atau sudah maksimum (level 5) }
{ F.S Jika awalnya monster level belum max maka monster level terupgrade
sampai keinginan user dengan level maks,yaitu level 5 }
```

### KAMUS LOKAL

monster\_level, oc : integer  
monster\_name, next\_upgrade : string  
upgrade\_prices : array of integer

### ALGORITMA

```
    while (True) do
        if (monster_level = 5 ) then
            output("Monster Level sudah maksimal")
            break
        upgrade_price ← upgrade_prices[monster_level-1]
        if ((upgrade_price) > oc ) then
            output("OC anda tidak cukup")
            break
        oc ← oc - (upgrade_price)
        monster_level ← monster_level - 1
        output("Selamat, {monster_name} berhasil diupgrade menjadi level
{monster_level}")
        next_upgrade ← input("Lanjutkan upgrade (y/n) : ")
        depend on (next_upgrade)
```

```

        next_upgrade = "n" : break
    next_upgrade ≠ "y" : output("erorr, input salah") break

```

- **F12 – Shop Management**

### **Program Shop Management**

```

{ Melakukan fungsi utama shop_management untuk penambahan item ke shop }

{ I.S. database dan user_data terdefinisi }

{ F.S. database mengalami modifikasi sesuai program }

```

### **KAMUS**

```

function shop_management(database,user_data)

```

### **ALGORITMA PROGRAM UTAMA**

```

    monster ← database['monster']
    monster_shop ← database['monster_shop']
    Item_shop ← database['item_shop']

    if user_role.lower() ≠ "admin" then {cek akses}
        output("Anda tidak memiliki akses ke shop management")
        return → monster_shop,item_shop

    isManaging ← True
    while isManaging do

        menu_monsters ← merge_tables_by_id(monster,monster_shop)
        menu_items ← generate_id_potion(item_shop)
        aksi ← str(input("Pilih aksi(lihat/tambah/ubah/hapus/keluar)
:")) .lower()
        depend on (aksi)
            aksi = "lihat" : objek ← str(input(f"Mau {aksi}
apa?(monster/potion?")) .lower()
            depend on (objek)
                objek = "monster" : table_interface(menu_monsters,"TIDAK
ADA MONSTER DI DALAM SHOP")
                objek = "potion" : table_interface(menu_items,"TIDAK ADA
POTION DI DALAM SHOP")
            else
                output("Input Salah")
                aksi = "tambah" : objek ← str(input(f"Mau {aksi}
apa?(monster/potion?")) .lower()
                depend on (objek)
                    objek = "monster" : not_added ←
filter_table_by_id(monster,monster_shop)

```

```

        if len(not_added) = 1 then
            output("Semua jenis monster telah ada di
shop")
        else
            table_interface(not_added,"")
            stock,harga,id_monster ←
input_data(objek,"awal",not_added)
            monster_shop.append([id_monster,stock,harga])

            objek = "potion" : potion ←
[["type"],["strength"],["resilience"],["healing"]] {jenis potion
hardcode}
            not_added ←
generate_id_potion(filter_table_by_id(potion,item_shop))
            if len(not_added) = 1 then
                output("Semua jenis potion telah ada di
shop")
            else
                table_interface(not_added,"")
                stock ,harga ,id_potion ←
input_data(objek,"awal",not_added)
                item_shop.append([id_potion,stock,harga])
            else
                output("input tidak valid")
                aksi = "ubah" : objek ← str(input(f"Mau {aksi}
apa?(monster/potion?")).lower()
                depend_on (objek)
                objek = "monster" : stock ,harga ,id_monster
← input_data(objek,"baru",menu_monsters)
                if len(monster_shop) = 1 then
                    output("Belum terdapat monster apapun pada
monster shop, silahkan tambahkan terlebih dahulu pada fitur tambah")
                else
                    stock,harga,id_monster ←
input_data(objek,"baru",menu_monsters)
                    index_monster ←
get_index_by_id(id_monster,monster_shop)
                    monster_shop[index_monster] ←
[id_monster,stock,harga]
                    objek = "potion" :
                    if len(item_shop) = 1 then
                        output("Belum terdapat potion apapun
pada potion shop, silahkan tambahkan terlebih dahulu pada fitur tambah")
                    else
                        stock,harga, id_potion ←
input_data(objek,"baru",menu_items)
                        index_potion ← id_potion

```

```

        item_shop[index_potion][1] ← stock
        item_shop[index_potion][2] ← harga
    else
        output("input salah")
    aksi = "hapus" : objek ← str(input("Mau hapus apa min?
(monster/potion?"))).lower()
    depend_on (objek)
    objek = "monster" :
        if len(monster_shop) = 1 then
            output("MONSTER PADA SHOP MONSTER TELAH
KOSONG")
        else
            table_interface(menu_monsters,"")
            _,_,id_monster ←
input_data(objek,"hapus",menu_monsters)
            yakin ← str(input("Yakin mengubah (y/n)
")).lower()
            depend_on (yakin)
            yakin = "y" :
                index_monster ←
                get_index_by_id(id_monster,monster_
r_shop)
                monster_shop ←
delete_row(index_monster,monster_shop)
            yakin = "n" :
                output("Tidak jadi
mengubah")
            else
                output("input salah")
            objek = "potion" :
                if len(item_shop) = 1 then
                    output("POTION PADA SHOP POTION TELAH
KOSONG")
                else
                    table_interface(menu_items, "POTION
PADA SHOP POTION TELAH KOSONG")
                    _,_,id_potion ←
input_data(objek,"hapus",menu_items)
                    yakin ← str(input("Yakin mengubah (y/n)
")).lower()
                    depend_on (yakin)
                    yakin = "y" :
                        output(item_shop)
                        item_shop ←
delete_row(int(id_potion),item_shop)
                        output(item_shop)

```

```

                                else
                                output("Tidak jadi
mengubah")

                                else
                                output("input salah")
                                aksi = "keluar" :
                                    isManaging ← False
                                    output("terimakasih")
                                else
                                    output("Fitur belum ada")
                                return → monster_shop,item_shop

```

#### REALISASI PROSEDUR

```
procedure table_interface(table,message)
```

#### KAMUS LOKAL

table,message : array of any

#### ALGORITMA

```

    {Cari panjang maks tiap kolom}
    column_length ← []
    if len(table) = 1 then
        return → output("tabel kosong")
    else
        col traversal [len(table[0])]
        max_column_length ← max(len(row[col]) for row in table)
        column_length.append(max_column_length)
        {Cetak tabel}
        for row in table:
            i traversal[len(row)]
            item ← row[i]
            kol ← column_length[i]
            output(" "+ str(item) + " " * (kol - len(str(item))+1) +
"|", end="")
            output()

```

#### REALISASI PROSEDUR

```
function filter_table_by_id(table1, table2)
```

#### KAMUS LOKAL

table1, table2 : array of any

#### ALGORITMA

```

    new_table ← []
    new_table.append(table1[0])
    i traversal [1...len(table1)]    {Mulai dari baris kedua}

```

```

    elemen_exist ← False
    j traversal [1... len(table2)] {Mulai dari baris kedua}
        if table1[i][0] = table2[j][0] then
            elemen_exist ← True
            break

    if not elemen_exist then
        new_table.append(table1[i])
    return → new_table

```

#### **REALISASI PROSEDUR**

```
function merge_tables_by_id(table1,table2)
```

#### **KAMUS LOKAL**

table1,table2 : array of any

#### **ALGORITMA**

```

header ← table1[0] + table2[0][1:]
merged_table ← [header]

for row1 in table1[1:]:
    id1 ← row1[0]
    for row2 in table2[1:]:
        id2 ← row2[0]
        if id1 = id2 then
            merged_table.append(row1 + row2[1:])
            break
return → merged_table

```

#### **REALISASI PROSEDUR**

```
function generate_id_potion(table_potion)
```

#### **KAMUS LOKAL**

table\_potion : string

#### **ALGORITMA**

```

new_table ← ["ID" + table_potion[0]]
i ← 1
for row in table_potion[1:] :
    new_table.append([str(i)] + row)
    i += 1
return → new_table

```

#### **REALISASI PROSEDUR**

```
function delete_row(indeks:int,table) : integer
```

#### KAMUS LOKAL

indeks:int,table : integer

#### ALGORITMA

```
indeks_maks ← len(table)-1
if indeks_maks < indeks or indeks < 0 then
    output("indeks tidak valid")
    return → table
else
    new_table ← []
    i traversal (indeks_maks+1)
        if i ≠ indeks then
            new_table.append(table[i])
    return → new_table
```

#### REALISASI PROSEDUR

function input\_data(objek:str,status\_input:str,shop\_data): integer

#### KAMUS LOKAL

objek:str,status\_input:str,shop\_data : integer

#### ALGORITMA

```
stock,harga ← "0","0" {status_input == hapus}
id ← int(input((f"Masukkan id {objek} : ")))
while (get_index_by_id(id,shop_data)) = -999 do
    id ← str(input((f"Id tidak terdapat pada shop \n Masukkan id
{objek} : ")))
    if status_input ≠ "hapus" then
        stock ← str(input(f"Masukkan stok {status_input} : "))
        while not is_all_same_type(stock,check_integer=True) or not
is_range_valid(stock,99999) do
            stock ← input_untill_type_valid(stock,f"Masukkan stock
{status_input} :",type="integer")
            stock ← input_untill_range_valid(stock,99999,f"Input Tidak Valid, stock harus ≥ 0\nMasukkan
stok{status_input} :")
            if status_input ≠ hapus then
                harga ← str(input(f"Masukkan harga {status_input} : "))
                while not is_all_same_type(harga,check_integer=True) or not
is_range_valid(harga,99999) do
                    harga ← input_untill_type_valid(harga,message=f"Masukkan harga
{status_input} : ",type="integer")
                    harga ← input_untill_range_valid(harga,99999,message=f"Input Tidak Valid, Harga
harus ≥ 0\nMasukkan harga {status_input} :")
                return → int(stock),int(harga),int(id)
```

- **F13 - Monster Management**

### **Program Monster Management**

{ Melakukan fungsi utama monster management untuk penambahan monster baru }

{ I.S. database dan user\_data terdefinisi }

{ F.S. database mengalami modifikasi sesuai program }

### **REALISASI PROSEDUR**

function is\_monster\_exist(monster:str,table\_monster) : bool

### **KAMUS LOKAL**

monster:str,table\_monster : bool

### **ALGORITMA**

```
found ← False
i ← 1
while not found and (i<len(table_monster)) do
    if monster.lower() = table_monster[i][1].lower() then
        found ← True
    i +=1
return → found
```

### **REALISASI PROSEDUR**

function input\_monster\_not\_exist(monster:str,table\_monster): string

### **KAMUS LOKAL**

monster:str,table\_monster : string

### **ALGORITMA**

```
while is_monster_exist(monster,table_monster) do
    monster ← str(input("Masukkan monster yang belum ada : "))
return → monster
```

### **REALISASI PROSEDUR**

procedure monster\_management(database,user\_data) : string

### **KAMUS LOKAL**

database,user\_data : array of any

### **ALGORITMA**

```
tabel_monster ← database['monster']

if user_role = "admin" then
    isManaging ← True
    while isManaging do
```



```

        output("SELAMAT DATANG DI DATABASE PARA MONSTER !!!\n1.
Tampilkan semua Monster \n2. Tambah Monster baru\n3.Cancel")
        aksi ← str(input("Pilih aksi(1/2) :"))
        if aksi = "1" then
            table_interface(tabel_monster,"DATABASE MONSTER KOSONG")
        else if aksi = "2" then
            nama_monster ← str(input("Masukkan type/nama monster :"))
            while not
is_all_same_type(nama_monster,check_integer=False) or
is_monster_exist(nama_monster,tabel_monster) do
                nama_monster ←
input_untill_type_valid(nama_monster,is_all_same_type,message="Masukkan
type/nama monster: ",type="string")
                nama_monster ←
input_monster_not_exist(nama_monster,tabel_monster)

                attack_power ← str(input("Masukkan attack power : "))
                while not is_all_same_type(attack_power,check_integer=True)
do
                    attack_power ←
input_untill_type_valid(attack_power,is_all_same_type,message="Masukkan
attack power : ",type="integer")

                def_power ← str(input("Masukkan def power (range 0 - 50) :
"))
                while not is_all_same_type(def_power,check_integer=True) or
not is_range_valid(def_power,51) do
                    def_power ←
input_untill_type_valid(def_power,is_all_same_type,message="Masukkan
def_power (range 0 - 50) : ",type="integer")
                    def_power ←
input_untill_range_valid(def_power,51,"Masukkan range def yang benar (0-
50)")

                hp_monster ← str(input("Masukkan HP monster : "))
                while not is_all_same_type(hp_monster,check_integer=True)
do
                    hp_monster ←
input_untill_type_valid(hp_monster,is_all_same_type,message="Masukkan HP
monster",type="integer")

                id_monster ← str(len(tabel_monster))

                new_monster ←
[int(id_monster),nama_monster,int(attack_power),int(def_power),int(hp_monst
er)]

                output("\n \n Monster baru telah berhasil dibuat")

```

```

        output(f"Type : {nama_monster} ")
        output(f"ATK Power : {attack_power}")
        output(f"DEF Power : {def_power}")
        output(f"HP : {hp_monster}\n")

        tambahkan ← str(input("Tambahkan monster ke database(Y/N) :
"")).lower()

        if tambahkan = "y" then
            isManaging ← False
            tabel_monster.append(new_monster)
            output("Monster berhasil ditambahkan")
            next_command ← str(input("Ingin menambahkan monster
lain :?"))

            if next_command = "y" then
                isManaging ← True
            else {tambahkan == "n"}
                output("Monster gagal ditambahkan")
            else if aksi = "3" then
                isManaging ← False
            else
                output("Input tidak valid")
        else
            output("Tidak memiliki akses" )

```

#### **F14 - Load**

##### **Program Load**

```

{ Melakukan load dalam progress yang telah dilakukan dalam game }
{ I.S. path terdefinisi }
{ F.S. folder terdefinisi }

```

##### **KAMUS**

```
function load () : string
```

##### **ALGORITMA PROGRAM UTAMA**

```

    parser ← argparse.ArgumentParser()
    parser.add_argument("name", type=str, help="name of saved game",
nargs="?", const='')
    args ← parser.parse_args()
    folder ← args.name

    if folder = None then
        output('Tidak ada nama folder yang diberikan!\nUsage :
python main.py <nama folder>\nJika kamu belum pernah bermain, silahkan
masukkan "python main.py database.")
        sys.exit()
    else
        path ← "C:/Users/ASUS/TUBESDAS/" + folder

```

```

        if (os.path.exists(path)) then
            database ← converting(folder)
        else
            output(f"Folder {folder} tidak ditemukan!")
            sys.exit()
        output('Selamat datang di Program OWCA!\nMasukkan command
"HELP" untuk melihat daftar command yang dapat kamu panggil.')
        return → database

```

#### REALISASI PROSEDUR

```
function integer_adjustment (tables,column_exception) : integer
```

#### KAMUS LOKAL

tables,column\_exception : array of any

#### ALGORITMA

```

    i traversal [1...len(tables)]
    j traversal [len(tables[0])]
    if j not in column_exception then
        tables[i][j] ← int(tables[i][j])
    return → tables

```

#### REALISASI PROSEDUR

```
function converting(folder_name:str) : dict
```

#### KAMUS LOKAL

folder\_name: string

#### ALGORITMA

```

    path ← "C:/Users/ASUS/TUBESDAS/" + folder_name

    if check_path_validity(path) then
        output("LOADING...")
        time.sleep(2)
        {Baca csv ubah ke tables , type blm di adjust (masih string)}
        user ← read_csv(path + "/user.csv")
        monster ← read_csv(path + "/monster.csv")
        monster_shop ← read_csv(path + "/monster_shop.csv")
        monster_inventory ← read_csv(path+"/monster_inventory.csv")
        item_shop ← read_csv(path + "/item_shop.csv")
        item_inventory ← read_csv(path + "/item_inventory.csv")

        database ← {
            "user" : integer_adjustment(user,[1,2,3]),
            "monster" : integer_adjustment(monster,[1]),
            "monster_shop" :
integer_adjustment(monster_shop,[ ]),
            "monster_inventory" :
integer_adjustment(monster_inventory,[ ]),
            "item_shop" : integer_adjustment(item_shop,[0]),

```

```

        "item_inventory"
:integer_adjustment(item_inventory,[1])
    }
    else if folder_name = "" then
        output("Tidak ada nama folder diberikan!")
        {program berhenti}
    else
        output(f"Folder {folder_name} tidak ditemukan!")
        {program berhenti}
    return → database

```

## **F15 - Save**

### **Program Save**

```

{ Melakukan save progress yang dilakukan dalam game }
{ I.S. folder terdefinisi }
{ F.S. game data disimpan }

```

### **REALISASI PROSEDUR**

```

procedure save(temp_data): string

```

### **KAMUS LOKAL**

```

temp_data : string

```

### **ALGORITMA**

```

    folder_name ← input ("Masukkan nama folder: ")
    path ← "C:/Users/ASUS/TUBESDAS/" + folder_name
    if not(os.path.exists(path)) then
        os.mkdir(path)
        output("Saving...\n")
        time.sleep(2)
        output(f"Membuat folder {folder_name}")
        time.sleep(1)
        output(f"Berhasil menyimpan data di folder
{folder_name}!")
    else
        output("Saving...\n")
        time.sleep(2)
        output(f"Berhasil menyimpan data di folder
{folder_name}!")
    uf.array_to_csv(temp_data["users"], path + "/user.csv")
    uf.array_to_csv(temp_data["monster"], path + "/monster.csv")
    uf.array_to_csv(temp_data["monster_shop"], path +
"/monster_shop.csv")

```

```

        uf.array_to_csv(temp_data["monster_inventory"], path +
"/monster_inventory.csv")
        uf.array_to_csv(temp_data["item_inventory"], path +
"/item_inventory.csv")
        uf.array_to_csv(temp_data["item_shop"], path + "/item_shop.csv")

```

## **F16 - Exit**

### **Program Exit**

```

{ Melakukan exit dari game yang dimainkan }

{ I.S. temp_data terdefinisi }

{ F.S. game melakukan exit }

```

### **REALISASI PROSEDUR**

```

procedure exit(database): string

```

### **KAMUS LOKAL**

```

database : string

```

### **ALGORITMA**

```

    validation ← input("Apakah Anda mau melakukan penyimpanan file yang
sudah diubah? (y/n) ")

    while not ((validation.upper() = "Y") or (validation.upper() = "N")) do
        validation ← input("Apakah Anda mau melakukan penyimpanan file yang
sudah diubah? (y/n) ")

    if validation.upper() = "Y" then
        save(database)
        sys.exit()
    else
        sys.exit()

```

## **B03 - Monster Ball**

### **Program Monster Ball**

```

{Melakukan fungsi utama monster ball}

{I.S. enemy_array terdefinisi}

{F.S. monster_inventory mengalami modifikasi sesuai program }

```

### **REALISASI PROSEDUR**

```

procedure MonsterBall (input enemy_array : array)

```

### **KAMUS LOKAL**

sukses : boolean

level, number : integer

#### ALGORITMA

success  $\leftarrow$  False

level  $\leftarrow$  enemy\_array[5]

number  $\leftarrow$  randomNumberGenerator(0,100)

depend on (level)

(level = 1) : if (number  $\leq$  75) then (success  $\leftarrow$  True)

(level = 2): if (number  $\leq$  50) then (success  $\leftarrow$  True)

(level = 3): if(number  $\leq$  25) then (success  $\leftarrow$  True)

(level = 4): if (number  $\leq$  10) then (success  $\leftarrow$  True)

(level = 5): if (number  $\leq$  5) then (success  $\leftarrow$  True)

if (success = True) then

output("Swoosshhhh, Anda mengeluarkan Monster Ball !!!")

output("Selamat, Anda berhasil mendapatkan Monster",enemy\_array[1])

output("Name : ",enemy\_array[1])

output("ATK Power : ",enemy\_array[2])

output("DEF Power : ",enemy\_array[3])

output("HP : ",enemy\_array[4])

output("Level : ",enemy\_array[5])

else

output("Swoosshhhh, Anda mengeluarkan Monster Ball !!!")

output("Yahhh, Anda belum berhasil mendapatkan monster Zuko !!!")

#### B04 - Jackpot

##### Program Jackpot

{Melakukan fungsi jackpot untuk melakukan gacha owcacoins serta mendapatkan monster}

{I.S. Owacoin terdefinisi}

{F.S. Owacoin bertambah maupun berkurang serta penambahan monster}

## KAMUS LOKAL

procedure Jackpotgraphics (input number  $\leftarrow$  integer)

{Melakukan fungsi print untuk gambar-gambar pada fungsi utama jackpot}

owcacoins, total, number : integer

item, slot : list

agree : string

## ALGORITMA PROGRAM UTAMA

item  $\leftarrow$  [{"apple",5}, {"pear",10}, {"cherry",20}, {"cherry",50}, {"SEVEN",100}]

output("=====")

output("                    WELCOME TO JACKPOT ")

output("=====")

output("1. Apple      = 5 oc")

output("2. Pear       = 10 oc")

output("3. Cherry     = 20 oc")

output("4. Clover     = 50 oc")

output("5. SEVEN      = 100 oc")

output("WOULD YOU LIKE TO PLAY (-100 oc) ? (Y/N)")

input(agree)

if (agree = "Y") then

if(owcacoins  $\geq$  100) then





```

        (slot[0] = 3) : total ← 900
                        output("Anda Mendapatkan 1000 OC")
        (slot[0] = 4) : total ← 500
                        output("Anda Mendapatkan 500 OC dan
Monster Roga")
                        output(monster roga)
        (slot[0] = 5) : total ← 1000
                        output("GACORRRRR ANDA MENDAPATKAN 1000
OC DAN MONSTER SHREK")
                        output(monster shrek)
        owcacoins ← owcacoins + total
    else
        output("NOT ENOUGH OWCACOIN")

```

#### **REALISASI PROSEDUR**

```

procedure Jackpotgraphics (input number ← integer)
{Melakukan fungsi print untuk gambar-gambar pada fungsi utama jackpot}
{I.S. number terdefinisi}
{F.S. output gambar graphics}

```

#### **KAMUS LOKAL**

```

number : integer

```

#### **ALGORITMA**

```

depend on (number)
    number = 1 :output(""
        .: .:~:~:

```

```

      7 8 9
    . 1 2 3 4 5 6 .
  . 1 2 3 4 5 6 7 8 .
' 1 2 3 4 5 6 7 8 9
' 1 2 3 4 5 6 7 8 9
' 1 2 3 4 5 6 7 8 9
' 1 2 3 4 5 6 7 8 9 """)

```

```
number = 2 :output(""
```

```

      1 2 3
    ' 1 2 3 .
      1 2 3 .
    . 1 2 3 4 .
  . 1 2 3 4 5 6 .
 1 2 3 4 5 6 7 8 9
' 1 2 3 4 5 6 7 8 9
' 1 2 3 4 5 6 7 8 9 """)

```

```
number = 3 :output(""
```

```

      ..
    . 1 2 3 4 5 .
  . 1 2 3 4 5 6 7 .
 1 2 3 4 5 6 7 8 9
 1 2 3 4 5 6 7 8 9
 1 2 3 4 5 6 7 8 9
 1 2 3 4 5 6 7 8 9
 1 2 3 4 5 6 7 8 9 """)

```

```
number = 4 :output(""
```

```

      1 2 3 4 5 .
    . 1 2 3 4 5 6 7

```

```

        :|||||:..
        '|||||:
        "||||"
        :|' '|'"")
number = 5 :output(""
        |||:|:|
        ||'|:|'
        :|||'
        :|||:
        :|:|:|"")

```

# Hasil Program

## ● F00 - Random Number Generator

```
# RANDOM NUMBER GENERATOR VIA LCG ALGORITHM
import time
import datetime
"""
Random Number Generator (RNG) akan digunakan untuk menghasilkan bilangan acak dari
sebuah Range angka. Fitur ini akan digunakan untuk berbagai fitur, terutama pada
Monster dan Battle. Implementasi RNG dilarang menggunakan library random python
dan harus menggunakan algoritma Linear Congruential Generator (LCG).
"""

def randomNumberGenerator(bottom:float, top:float):
    # bottom = batas bawah range nilai
    # top = batas atas range nilai
    # inisiasi nilai m, a, c, dan seed untuk LCG (most optimal values)
    m = 2**32
    a = 1103515245
    c = 12345 # ngambil waktu sekarang -> diconvert jadi int
    time.sleep(0.005)
    #seed
    current_time_ns = time.time_ns()
    microseconds = datetime.datetime.now().microsecond
    seed = current_time_ns + microseconds
    randomNumber = ((a * seed + c) % m) / m

    randomNumber = randomNumber * (top - bottom) + bottom # scale and shift ke range yang diinginkan
    return int(randomNumber) # ngembaliin nilai random
```

## ● F01 - Register

```
1  from src.utilityfunction import get_data_by_column_title
2  import src.validation as val
3  import src.F00 as RNG
4
5  def register(logged_in_user: str, database: dict):
6      registered_users = database["user"]
7      monster_inventory = database["monster_inventory"]
8      monster = database["monster"]
9      if logged_in_user == None:
10         logged_in_user = input("Masukkan username: ")
11         password = input("Masukkan password: ")
12
13         if val.check_username_availability(registered_users, logged_in_user) and val.check_username_validity(logged_in_user):
14             generated_id = RNG.randomNumberGenerator(1, 1000)
15
16             while val.is_registered_id(registered_users, generated_id):
17                 generated_id = RNG.randomNumberGenerator(1, 1000)
18             registered_users.append([str(generated_id), logged_in_user, password, "Agent", "0"])
19
20             print("\nSilahkan pilih salah satu monster sebagai monster awalmu.")
21             monster_name = get_data_by_column_title(monster, "type")
22             monster_id = get_data_by_column_title(monster, "id")
23
24             for i in range(1, len(monster)):
25                 print(f"{i}. {monster[i][1]}")
26             monster_choice = int(input("\nMonster pilihanmu: "))
27             monster_inventory.append([generated_id, monster_id[monster_choice-1], monster_name[monster_choice-1]])
28             print(f"Selamat datang Agent {logged_in_user}. Maria kita mengalahkan Dr. Asep Spakbor dengan {monster_name[monster_choice-1]}!")
29
30         else:
31             if not val.check_username_availability(registered_users, logged_in_user) :
32                 print(f"\nUsername {logged_in_user} sudah terpakai, silahkan gunakan username lain!")
```

```

33         else:
34             print("\nUsername hanya boleh berisi alfabet, angka, underscore, dan strip!")
35
36     else:
37         print(f'\nRegister gagal!\nAnda telah login dengan username {logged_in_user} silahkan lakukan "LOGOUT" sebelum melakukan register.')
38     return database
39

```

## ● F02 - Login

```

1     import src.utilityfunction as uf
2
3
4     def login(database:dict,logged_in_user,user_id,user_role):
5         registered_users = database["user"]
6         if logged_in_user == None:
7             username = input("Username: ")
8             password = input("Password: ")
9             registered_username = uf.get_data_by_column_title(registered_users, 'username')
10            if username not in registered_username:
11                print("Username tidak terdaftar!")
12                user_id,user_role = None,None
13            else:
14                user_password = uf.get_row(registered_users, 'username', username)[0][2]
15                if password == user_password:
16                    logged_in_user = username
17                    user_id = int(uf.get_row(registered_users, 'username', username)[0][0])
18                    user_role = uf.get_row(registered_users, 'username', username)[0][3]
19                    if user_role == "Admin":
20                        print(f'Selamat datang, Admin {username}!\nMasukkan command "HELP" untuk daftar command yang dapat kamu panggil.')
21                    else:
22                        print(f'Selamat datang, Agent {username}!\nMasukkan command "HELP" untuk daftar command yang dapat kamu panggil.')
23                else:
24                    print("Password salah!")
25                    user_id,user_role = None,None
26            else:
27                print(f'Login gagal!\nAnda telah login dengan username {logged_in_user} silahkan lakukan "LOGOUT" sebelum melakukan login kembali.')
28            return logged_in_user, user_id, user_role

```

## ● F03 - Logout

```

def logout(logged_in_user):
    if logged_in_user != None:
        print("logout telah berhasil")
    else:
        print(f'Logout gagal!\nAnda belum login, silahkan login terlebih dahulu sebelum melakukan logout.')
    return None,None,None

```

## ● F04 - Menu and Help



```

31         print("Defense Power :",array[1+i][3])
32         print("Monster Health :",array[1+i][4])
33         print("=====")
34     else:
35         print('Oucadex hanya ditujukan untuk Agent. Silahkan masukkan command "HELP" untuk melihat daftar command yang dapat kamu panggil.')
36
37     # Print graphics
38     graphics = [
39         """
40             | | | | | | | | | |
41             | | | | | | | | | |
42             | | | | | | | | | |
43             | | | | | | | | | |
44             | | | | | | | | | |
45             | | | | | | | | | |
46             | | | | | | | | | |
47             | | | | | | | | | |
48             | | | | | | | | | |
49             | | | | | | | | | |
50             | | | | | | | | | |
51             | | | | | | | | | |
52         """,
53         """
54             | | | | | | | | | |
55             | | | | | | | | | |
56             | | | | | | | | | |
57             | | | | | | | | | |
58             | | | | | | | | | |
59             | | | | | | | | | |
60             | | | | | | | | | |

```

```

61             | | | | | | | | | |
62             | | | | | | | | | |
63             | | | | | | | | | |
64             | | | | | | | | | |
65             | | | | | | | | | |
66             | | | | | | | | | |
67             | | | | | | | | | |
68             | | | | | | | | | |
69             | | | | | | | | | |
70
71         """,
72         """
73             | | | | | | | | | |
74             | | | | | | | | | |
75             | | | | | | | | | |
76             | | | | | | | | | |
77             | | | | | | | | | |
78             | | | | | | | | | |
79             | | | | | | | | | |
80             | | | | | | | | | |
81             | | | | | | | | | |
82             | | | | | | | | | |
83             | | | | | | | | | |
84             | | | | | | | | | |
85             | | | | | | | | | |
86             | | | | | | | | | |
87
88         """,
89         """
90             | | | | | | | | | |

```

```

91         .dP#  .P  'P#
92         HH#  P#  P#  P#
93         P'  P#  P#  P#  P#
94         P'P#  P#  P#  P#  P#
95         P'P#  P#  P#  P#  P#
96         P'P#  P#  P#  P#  P#
97         P'P#  P#  P#  P#  P#
98         P'P#  P#  P#  P#  P#
99         P'P#  P#  P#  P#  P#
100        P'P#  P#  P#  P#  P#
101        P'P#  P#  P#  P#  P#
102        P'P#  P#  P#  P#  P#
103        P'P#  P#  P#  P#  P#
104        P'P#  P#  P#  P#  P#
105        P'P#  P#  P#  P#  P#
106        P'P#  P#  P#  P#  P#
107        P'P#  P#  P#  P#  P#
108        P'P#  P#  P#  P#  P#
109        P'P#  P#  P#  P#  P#
110        P'P#  P#  P#  P#  P#
111        P'P#  P#  P#  P#  P#
112        P'P#  P#  P#  P#  P#
113        P'P#  P#  P#  P#  P#
114        P'P#  P#  P#  P#  P#
115        P'P#  P#  P#  P#  P#
116        P'P#  P#  P#  P#  P#
117        P'P#  P#  P#  P#  P#
118        P'P#  P#  P#  P#  P#
119        P'P#  P#  P#  P#  P#
120        P'P#  P#  P#  P#  P#

```

```

121        P'P#  P#  P#  P#  P#
122        P'P#  P#  P#  P#  P#
123        P'P#  P#  P#  P#  P#
124        P'P#  P#  P#  P#  P#
125        P'P#  P#  P#  P#  P#
126        P'P#  P#  P#  P#  P#
127        P'P#  P#  P#  P#  P#
128        P'P#  P#  P#  P#  P#

```

## ● F06 - Potion

```

1  # Memerlukan stats monster selama battle saja, pengambilan stats diambil dalam fitur battle
2  # Strength Potion
3  # def strength_potion(battle_atk_power, strength_potion_use):
4  #     if (strength_potion_use > 0):
5  #         print("Strength potion hanya bisa digunakan sekali")
6  #         battle_atk_power = battle_atk_power
7  #     else :
8  #         battle_atk_power = battle_atk_power * 1.05
9  #         strength_potion_use += 1
10 #     return (battle_atk_power, strength_potion_use)
11
12 # # Resilience Potion
13 # def resilience_potion(battle_def_power, resilience_potion_use):
14 #     if (resilience_potion_use > 0):
15 #         print("Resilience potion hanya boleh digunakan sekali")
16 #         battle_def_power = battle_def_power
17 #     else:
18 #         battle_def_power = battle_def_power * 1.05
19 #         resilience_potion_use += 1
20 #     return (battle_def_power, resilience_potion_use)
21
22 # # Healing Potion
23 # def healing_potion(battle_current_hp , max_hp, healing_potion_use):
24 #     healing = battle_current_hp * 1.25
25 #     if (healing_potion_use > 0):
26 #         print("Healing potion hanya boleh digunakan sekali")
27 #         battle_current_hp = battle_current_hp
28 #     else:
29 #         if (healing >= max_hp):
30 #             battle_current_hp = max_hp
31 #         else:
32 #             battle_current_hp = healing
33 #     return (battle_current_hp, healing_potion_use)
34
35 # perhitungan langsung pada battle

```



## ● F07 - Inventory

```

1  def take_monster_id_level (user_id, monster_inventory):
2      monster_id = []
3      monster_level = []
4      for i in range (len(monster_inventory)-1):
5          if (user_id == int(monster_inventory[i+1][0])):
6              monster_id.append(monster_inventory[i+1][1])
7              monster_level.append(monster_inventory[i+1][2])
8      return (monster_id, monster_level)
9
10 def base_monster_in_inventory (monster_id,user_id,monster_array,monster_inventory):
11     monster_level = take_monster_id_level(user_id,monster_inventory)[1]
12     base_monster = []
13     for i in range (len(monster_id)):
14         for j in range (len(monster_array)-1):
15             if (int(monster_id[i]) == int(monster_array[j+1][0])):
16                 base_monster.append(monster_array[j+1]+list(monster_level[i]))
17     print(base_monster)
18     return (base_monster)
19
20 def level_calculation(base_monster):
21     monster_in_inventory = base_monster
22     for i in range (len(base_monster)):
23         header = False
24         if str(monster_in_inventory[i][0]).lower() == "id" :
25             header = True
26         if (str(monster_in_inventory[i][5]) != "1" and not header :
27             #atk_power
28             monster_in_inventory[i][2] = (float(monster_in_inventory[i][2]) + (float(monster_in_inventory[i][2])) * (float(monster_in_inventory[i][5])*0.1))
29             #def_power
30             monster_in_inventory[i][3] = (float(monster_in_inventory[i][3]) +(float(monster_in_inventory[i][3])) * (float(monster_in_inventory[i][5])*0.1))
31             #health

```

```

32         monster_in_inventory[i][4] = (float(monster_in_inventory[i][4]) +(float(monster_in_inventory[i][4])) * (float(monster_in_inventory[i][5])*0.1))
33     elif (str(monster_in_inventory[i][5])) == "1" and not header :
34         monster_in_inventory[i][2] = float(monster_in_inventory[i][2])
35         monster_in_inventory[i][3] = float(monster_in_inventory[i][3])
36         monster_in_inventory[i][4] = float(monster_in_inventory[i][4])
37     return(monster_in_inventory)
38
39
40 def take_item_id (user_id,item_inventory):
41     item_id = []
42     for i in range (len(item_inventory)-1):
43         if (user_id == int(item_inventory[i+1][0])):
44             item_id.append(item_inventory[i+1])
45     return (item_id)
46
47 def display_inventory(user_id,current_inventory_item,current_inventory_monster,owca_coin):
48     count = 0
49     print("----- INVENTORY LIST (User ID :",str(user_id)+") -----")
50     print("Jumlah O.W.C.A. Coin-mu sekarang",owca_coin)
51     for i in range(len(current_inventory_monster)):
52         print(str(i+1)+". Monster      (Name:",current_inventory_monster[0+i][1],", Lvl:",current_inventory_monster[0+i][5],", HP:",str(current_inventory_monster[0+i][4])+")")
53         count+=1
54     for j in range(len(current_inventory_item)):
55         if (current_inventory_item[0+j][1] == "Monster Ball"):
56             print(str(count+1+j)+". Monster Ball  (Qty:",current_inventory_item[0+j][2])+")")
57         else:
58             print(str(count+1+j)+". Potion      (Type:",current_inventory_item[0+j][1],", Qty:",str(current_inventory_item[0+j][2])+")")
59
60 def inventory(database,user_id):
61     monster_inventory = database["monster inventory"]

```

```

61     monster_inventory = database["monster_inventory"]
62     item_inventory = database["item_inventory"]
63     monster_array = database["monster"]
64     owca_coin = 837
65     monster_id = take_monster_id_level(user_id, monster_inventory)[0]
66     base_monster = base_monster_in_inventory(monster_id, user_id, monster_array, monster_inventory)
67     current_inventory_monster = level_calculation(base_monster)
68     current_inventory_item = take_item_id(user_id, item_inventory)
69     display_inventory(user_id, current_inventory_item, current_inventory_monster, owca_coin)
70     count_monster = int(len(current_inventory_monster))
71     count_item = int(len(current_inventory_item))
72     inventory_command = input()
73     while (inventory_command != "KELUAR"):
74         if (1 <= int(inventory_command) <= (count_monster + count_item)):
75             if (1 <= int(inventory_command) <= count_monster):
76                 take_monster = current_inventory_monster[int(inventory_command)-1]
77                 print("Monster")
78                 print("Name          :", take_monster[1])
79                 print("ATK Power   :", take_monster[2])
80                 print("DEF Power   :", take_monster[3])
81                 print("HP          :", take_monster[4])
82                 print("Level       :", take_monster[5])
83             if (count_monster < int(inventory_command) <= (count_monster + count_item)):
84                 take_item = current_inventory_item[int(inventory_command)-1-count_monster]
85                 if (take_item[1] == "Monster Ball"):
86                     print("Monster Ball")
87                     print("Quantity :", take_item[2])
88                 else:
89                     print("Potion")
90                     print("Type      :", take_item[1])
91                     print("Quantity  :", take_item[2])

```

```

92         print("Ketikkan id untuk menampilkan id item :")
93         inventory_command = input()
94     else:
95         print("Command diluar range, silahkan masukan range valid")
96         print("Ketikkan id untuk menampilkan id item :")
97         inventory_command = input()
98

```

## ● F08 - Battle

```

1  from src.F11 import get_user_inventory
2  from src.validation import get_index_by_id, is_all_same_type
3  from src.F00 import randomNumberGenerator
4  from src.F05 import graphics
5  from src.F12 import merge_tables_by_id
6  from src.F05 import damage_dealt, damage_taken
7  from src.F07 import level_calculation
8  from src.F11 import update_inventory_by_id
9  import time
10
11  def MonsterBall (enemy_array:list)->bool:
12      # monsterball
13      success = False
14      level = enemy_array[5]
15      number = randomNumberGenerator(0,100)
16      if(level == 1):
17          if(number <= 75):
18              success = True
19      if(level == 2):
20          if(number <= 50):
21              success = True
22      if(level == 3):
23          if(number <= 25):
24              success = True
25      if(level == 4):
26          if(number <= 10):
27              success = True
28      if(level == 5):
29          if(number <= 5):
30              success = True
31      if (success == True):

```

```

36         print("DEF Power      :",enemy_array[3])
37         print("HP              :",enemy_array[4])
38         print("Level          :",enemy_array[5])
39     else:
40         print("Swoosshhhh, Anda mengeluarkan Monster Ball !!!")
41         print("Yahhh, Anda belum berhasil mendapatkan monster Zuko !!!")
42     return success
43
44     def print_stat_awal(monster:list,graphics:list,text:str) :
45         # Procedure mencetak stat monster pada awal battle
46         random_index = randomNumberGenerator(0,len(graphics))
47         print(graphics[random_index])
48         print(text)
49         print(f"Monster Type      : {monster[1]}")
50         print(f"Attack Power      : {int(monster[2])}")
51         print(f"Defense Power     : {int(monster[3])}")
52         print(f"Monster Health    : {int(monster[4])}")
53         print(f"Level : {monster[5]}")
54         print("=====")
55     def select_available_monster(user_monsters:list):
56         # Fungsi menerima input user_monster(inventory monster yang dimiliki user)
57         # Fungsi mengeluarkan output isFighting bernilai True dan user_monster yang tidak kosong jika inventory monster user tidak kosong
58         isFighting = True
59         user_monster = []
60         urutan = 1
61         print("=====MONSTER LIST=====")
62         for i in range(1,len(user_monsters)) : # printing monster list
63             print(f"{urutan}. {user_monsters[i][1]} (level {user_monsters[i][5]}) ")
64             urutan +=1
65
66         urutan += 1
67
68         index_monster = str(input("pilih monster :"))
69         attempts_id = 10
70         while int(index_monster)>=len(user_monsters) or int(index_monster) < 1 or not is_all_same_type(index_monster,check_integer=True): # validating input
71             print("Monster tidak ada, Masukkan pilihan monster yang ada !")
72             index_monster = str(input("pilih monster : "))
73             attempts_id -=1
74         if attempts_id == 0 :
75             isFighting = False
76             print("Sepertinya anda belum memiliki monster, silahkan beli di shop terlebih dahulu")
77         else :
78             user_monster = user_monsters[int(index_monster)]
79
80         return isFighting,user_monster
81
82     def input_perintah(turns,user_monster) :
83         # Fungsi mencetak perintah yang tersedia serta meminta user melakukan pemilihan perintah
84         print(f"===== Turns {turns} ({user_monster[1]})=====")
85         print("1. Attack")
86         print("2. Use Potion")
87         print("3. Monsterball")
88         print("4. Quit")
89         perintah = str(input("Pilih perintah : "))
90         return perintah
91
92     def print_status(turns:int,attacker_monster:list,defender_monster:list,attacker:list) :
93         # Procedure mencetak keadaan monster setelah diserang
94         if defender_monster[4] <= 0 : # cek hp
95             defender_monster[4]= 0
96         if attacker == "enemy" :
97             print(f"===== Turns {turns} ({attacker_monster[1]})=====")

```

```

95     print(f"Monster Type : {defender_monster[1]}")
96     print(f"Attack Power : {int(defender_monster[2])}")
97     print(f"Defense Power : {int(defender_monster[3])}")
98     print(f"Monster Health : {int(defender_monster[4])}")
99     print(f"Level : {defender_monster[5]}")
100    print("=====")
101
102    def print_potion(user_inventory) :
103        # I.S potion quantity kosong
104        # F.S potion quantity terisi kuantitas tiap potion yang dimiliki user
105        potion_arr = ["strength", "resilience", "healing"]
106        potion_quantity = []
107        for potion in potion_arr :
108            found = False
109            for row in user_inventory :
110                if potion == row[0] :
111                    potion_quantity.append(row[1])
112                    found = True
113                    break
114            if not found :
115                potion_quantity.append("0")
116        print(f"-----POTION LIST-----")
117        print(f"1.Strength Potion (Quantity : {potion_quantity[0]})---Increase ATK Power")
118        print(f"2.Resilience Potion (Quantity : {potion_quantity[1]})---Increase DEF power")
119        print(f"3.Healing Potion (Quantity : {potion_quantity[2]})---Restore Health")
120        print(f"4.Cancel")
121        print("=====")
122
123    def random_monster_data(monster, level_monster) :
124        # Fungsi mengambil monster lawan secara acak
125        index_monster = random.randint(1, len(monster))
126        monster_enemy = monster[index_monster] + [str(level_monster)]
127        return monster_enemy
128
129
130
131    def duel(battle_status, user_monster, enemy_monster, oc_reward) :
132        # I.S battle_status belum diperbarui
133        # F.S battle status diperbarui setelah melakukan serangkaian pertarungan
134
135        # Player menyerang monster bot terlebih dahulu
136        damage_dealt_user = damage_dealt(atk_power=user_monster[2]) # Damage yang diberikan user ke monster bot lawan
137        damage_taken_enemy = damage_taken(def_power=enemy_monster[3], damage_dealt=damage_dealt_user) # Damage yang diterima monster bot lawan
138        enemy_monster[4] = int(enemy_monster[4] - damage_taken_enemy) # Pengurangan hp bot monster lawan
139        print_status(turns=battle_status[0], attacker_monster = user_monster, defender_monster=enemy_monster, attacker="user") # Mencetak status atribut dan hp monster bot setelah diserang
140        time.sleep(0.5) # jeda waktu
141
142        # Monster bot kemudian memberikan serangan balasan ke user
143        damage_dealt_enemy = damage_dealt(atk_power=enemy_monster[2]) # Damage yang diberikan monster ke bot lawan
144        damage_taken_user = damage_taken(def_power=user_monster[3], damage_dealt=damage_dealt_enemy) # Damage yang diterima monster bot lawan
145        user_monster[4] = int(user_monster[4] - damage_taken_user) # Pengurangan hp bot user
146
147
148        # Mengecek apakah sudah ada monster yang wafat atau belum
149        if enemy_monster[4] <= 0 : # User menyerang terlebih dahulu sehingga kemungkinan monster bot wafat lebih dulu
150            battle_status[1] = False # Battle berakhir
151            battle_status[2] = True # User menang
152
153            battle_status[5] += oc_reward # Menambahkan hadiah oc
154        elif user_monster[4] <= 0 : # Jika monster user telah wafat
155            print_status(turns=battle_status[0], attacker_monster=enemy_monster, defender_monster=user_monster, attacker="enemy")
156            battle_status[1] = False # Battle berakhir
157            battle_status[2] = False # User kalah
158        else : # Pertarungan belum selesai karena belum ada monster yang wafat
159            battle_status[1] = True # Battle masih berlanjut
160            battle_status[2] = False # Belum ada Pemenang
161            print(f"WHOOZZZ Monster {enemy_monster[1]} menyerang balik anda")
162            print_status(turns=battle_status[0], attacker_monster=enemy_monster, defender_monster=user_monster, attacker="enemy")
163
164            battle_status[0] += 1 # turns berikutnya
165            battle_status[3] += damage_dealt_user #
166            battle_status[4] += damage_taken_user
167
168    def drink_potion(user_item_inventory, potion_used: list, user_monster: list, hp_max: int) :
169        # I.S user_monster masih belum terupdate
170        # F.S Stat pada user_monster terupdate jika user menggunakan potion
171        potion_arr = ["strength", "resilience", "healing"]
172        daftar_id = ["1", "2", "3", "4"]
173        drink_potion = False
174        # pilih potion
175        id_potion = str(input("Pilih type : "))
176        while id_potion not in daftar_id : # Meminta hingga input id_potion valid
177            print("Input salah")
178            id_potion = str(input("Pilih potion : "))
179
180        if id_potion == "4" : # User tidak jadi menggunakan potion
181            print("Engga jadi beli")
182            cancel = True

```

```

168 def drink_potion(user_item_inventory,potion_used:list,user_monster:list,hp_max:int) :
169     index_potion_in_inventory = get_index_by_id(type_potion,user_item_inventory)
170     cancel = False
171     # Mengecek apakah potion yang dipilih sudah digunakan
172     if not cancel and (type_potion in potion_used) :
173         cancel = True
174         if type_potion == "strength" :
175             print(f"Tadi udah minum jamu kuat")
176         elif type_potion == "resilience" :
177             print(f"Udah minum resilience bro")
178         else : # type_potion == "healing"
179             print(f"Tadi udah minum teh healing ini")
180
181     # Mengecek apakah terdapat stock potion yang tersedia jika user memilih untuk menggunakan potion
182     if index_potion_in_inventory != -999 and not cancel:
183         current_stock = int(user_item_inventory[index_potion_in_inventory][1] )
184         drink_potion = False
185         if current_stock == 0 :
186             kosong = True
187         else :
188             user_item_inventory[index_potion_in_inventory][1] = current_stock-1
189             kosong = False
190             drink_potion = True
191     elif index_potion_in_inventory == -999 and not cancel :
192         print(f"anda tidak memiliki potion (type_potion)")
193         kosong = True
194     # Potion akan digunakan player jika beberapa kondisi berikut terpenuhi(tidak dicancel,potion blm digunakan, dan )
195     if not cancel and drink_potion:
196         potion_used.append(type_potion)
197         if type_potion == "strength" and not kosong: # Efek tiap potion berlaku selama battle

```

```

217         new_atk = int(user_monster[2] *1.05 )
218         user_monster[2] = new_atk
219         print(f"Setelah minum ruan ini atk power meningkat menjadi {new_atk}")
220     elif type_potion == "resilience" and not kosong :
221         new_def = int(user_monster[3]*1.05 )
222         user_monster[3] = new_def
223         print(f"Setelah minum ramuan ini, def power meningkat menjadi {new_def}")
224     elif type_potion == "healing" and not kosong :
225         user_monster[4] = int(user_monster[4]*1.25)
226         if user_monster[4] > hp_max :
227             user_monster[4] = hp_max
228             user_monster[4] = hp_max
229         print(f"Nyawa monster telah disembuhkan menjadi {user_monster[4]}")
230
231
232 def battles(database,user_id,user_role) :
233
234     if user_role != "agent" :
235         return print("Ini area bertarung agen!")
236     # Mengambil data database
237     user = database["user"]
238     item_inventory = database["item_inventory"]
239     monster_inventory = database["monster_inventory"]
240     monster = database["monster"]
241     # Random monster level untuk battle
242     enemy_level = randomNumberGenerator(1,5)
243     user_monster_inventory = get_user_inventory(monster_inventory,user_id) # penyimpanan monster user
244     user_item_inventory = get_user_inventory(item_inventory,user_id) # penyimpanan item user
245     user_monsters = level_calculation(merge_tables_by_id(monster,user_monster_inventory)) # Data user monster yang telah dikalkulasi berdasarkan level
246     enemy_monsters_tables = level_calculation([random_monster_data(monster,enemy_level)]) # Data enemy monster yang telah dikalkulasi berdasarkan level

```

```

248     print_stat_awal(monster=enemy_monster,graphics=graphics,text= f"ROARRRR {enemy_monster[1]} telah muncul") # Spek awal stat monster musuh
249     isFighting, user_monster = select_available_monster(user_monsters) # Memilih monster mengecek apakah bisa melanjutkan pertarungan
250     hp_max = int(user_monster[4])
251     turns = 1
252     potion_used = []
253     print_stat_awal(monster=user_monster,graphics=graphics,text=f"HUFTTT Agen telah mengeluarkan monster {user_monster[1]}") # Spek awal stat monster user
254     battle_status = [turns,isFighting,False,0,0,0] # Status battle
255     # Battle status,=> index 3 {total damage death user} index 4{total damage_taken_user} index 5 {penambahan oc user}
256     while isFighting :
257         # Meminta input perintah
258         perintah = input_perintah(turns,user_monster)
259         daftar_perintah = ["1","2","3","4"]
260         while perintah not in daftar_perintah :
261             print("Fitur perintah tersebut belum ada")
262             perintah = str(input("Pilih perintah : "))
263         # jika user memilih potion maka user akan diberikan fitur minum potion terlebih dahulu sebelum menyerang
264         if perintah == "2" :
265             print_potion(user_inventory=user_item_inventory)
266             drink_potion(user_item_inventory=user_item_inventory,potion_used=potion_used,user_monster=user_monster,hp_max=hp_max)
267         # jika user memilih menggunakan monsterball maka, monsterball akan digunakan ,
268         # pada program ini kesempatan menggunakan monsterball hanya sekali tiap battle
269         elif perintah == "3" :
270             index_monsterball = get_index_by_id(user_id,user_item_inventory)
271             get_monster = False
272             if index_monsterball != -999 :
273                 print("Anda tidak mempunyai monsterball")
274             else :
275                 get_monster = MonsterBall(enemy_monster)
276             if get_monster :
277                 user_item_inventory[index_monsterball][2] -= 1 # satu buah monsterball telah digunakan

```

```

279         isFighting = False # pertarungan berakhir karena monster musuh tertangkap
280         index_in_inventory = get_index_by_id(enemy_monster[0],user_monster_inventory)
281         if index_in_inventory != -999 : # mmonster yang tertangkap sudah ada di inventory
282             # converting monster to oc
283             convert_to_oc = randomNumberGenerator(0,100)
284             print(f"Sayangnya monster sudah terdapat pada inventory, sehingga monster diconvert menjadi {convert_to_oc}")
285             battle_status[5] += convert_to_oc
286         else : # monster belum ada di inventory
287             new_monster = [user_id] + [enemy_monster[0]] + [enemy_monster[5]]
288             monster_inventory.append(new_monster) # menambahkan monster ke database inventory
289         # Jika user memilih untuk kabur dari pertandingan
290         elif perintah == "4" :
291             isFighting = False
292             print("Selamat berjumpa lagi di battle selanjutnya")
293         # Jika pertarungan masih dapat berlangsung, maka monster user dan monster bot akan bertarung
294         if isFighting :
295             oc_reward = randomNumberGenerator(0,50)
296             duel(battle_status,user_monster,enemy_monster,oc_reward) # pertarungan
297             turns,isFighting = battle_status[0],battle_status[1] # update elemen dengan battle status
298             win = battle_status[2]
299             if win and not isFighting:
300                 print(f"Keren monster {enemy_monster[1]} berhasil anda bunuh ") # User menang
301             elif not win and not isFighting :
302                 print("Anda kalah, mungkin next time ,asahlah skill hoki di arena dulu dek") # user kalah
303
304
305         # update database
306         print(f"Total OC yang berhasil anda dapatkan : {battle_status[5]}")
307         index_user = get_index_by_id(user_id,user)
308         user[index_user][4] += battle_status[5] # menambahkan oc ke database

```

## ● F09 - Arena

```

1  from src.F08 import random_monster_data,drink_potion,print_potion,duel,select_available_monster,print_stat_awal
2  from src.F08 import graphics
3  from src.F11 import get_user_inventory,update_inventory_by_id
4  from src.F12 import merge_tables_by_id
5  from src.F07 import level_calculation
6  from src.validation import get_index_by_id
7  def input_perintah(turns,user_monster) :
8      # Fungsi mencetak perintah yang tersedia serta meminta user melakukan pemilihan perintah
9      print(f"===== Turns {turns} ({user_monster[1]})=====")
10     print("1. Attack")
11     print("2. Use Potion")
12     print("3. Quit")
13     perintah = str(input("Pilih perintah : "))
14     return perintah
15
16  def arena(database,user_id,user_role) :
17
18     if user_role != "agent" : #check akses role
19         return print("Ini area bertarung agen!")
20     # database
21     user = database["user"]
22     item_inventory = database["item_inventory"]
23     monster_inventory = database["monster_inventory"]
24     monster = database["monster"]
25
26     stage = 1 # stage awal
27     user_monster_inventory = get_user_inventory(monster_inventory,user_id) # penyimpanan monster user
28     user_item_inventory = get_user_inventory(item_inventory,user_id) # penyimpanan item user
29     user_monsters = level_calculation(merge_tables_by_id(monster,user_monster_inventory)) # Data user monster yang telah dikalkulasi berdasarkan level
30     enemy_monsters_tables = level_calculation([random_monster_data(monster,stage)]) # Data enemy monster yang telah dikalkulasi berdasarkan level

```

```

31     enemy_monster = enemy_monsters_tables[0] # enemy monster
32
33     print(f"=====STAGES {stage}=====")
34     print_stat_awal(monster=enemy_monster,graphics=graphics,text=f"ROARRR {enemy_monster[1]} telah muncul") # base atribut monster musuh
35     Arena, user_monster = select_available_monster(user_monsters) # cek apakah user memiliki monster sehingga pertarungan dapat berlangsung
36     if Arena : # pertarungan bisa berlangsung
37         print_stat_awal(monster=user_monster,graphics=graphics,text=f"HUFTTT Agen telah mengeluarkan monster {user_monster[1]}")
38         base_atk,base_def,hp_max = user_monster[2],user_monster[3],user_monster[4]
39         potion_used = [] # Hp maks monster user
40         turns = 1
41         win_battle = False #
42         oc_rewards = [30,50,100,120,150]
43         battle_status = [turns,Arena,win_battle,0,0,0] # indeks, 1 = 3 = damage dealt, 1 = 4 = damage taken, 1 = 5 = oc reward
44     else :
45         battle_status = [0,False,False,0,0,0]
46     while Arena : # Arena
47         perintah = input_perintah(turns,user_monster) # input perintah
48         daftar_perintah = ["1","2","3"]
49         while perintah not in daftar_perintah :
50             print("Fitur perintah tersebut belum ada")
51             perintah = str(input("Pilih perintah : "))
52         if perintah == "2" : # Jika player membuka potion maka player akan diberi pilihan potion sebelum attack
53             print_potion(user_inventory=user_item_inventory)
54             drink_potion(user_item_inventory=user_item_inventory,potion_used=potion_used,user_monster=user_monster,hp_max=hp_max)
55         elif perintah == "3" :
56             Arena = False
57             print("Pergi dari arena")
58         if Arena : # Arena pertarungan masih berlangsung
59             oc_reward = oc_rewards[stage-1] # hadiah tahap ini
60             duel(battle_status=battle_status,user_monster=user_monster,enemy_monster=enemy_monster,oc_reward=oc_reward) # monster bertarung

```

```

61     turns,Arena,win_battle = battle_status[0],battle_status[1],battle_status[2] # update keadaan battle
62     if win_battle and stage != 5:
63         print(f"Keren euy anda mengalahkan monster {enemy_monster[1]} ")
64         print(f"Anda mendapatkan {oc_reward} pada stage ini")
65         stage+=1
66         new_monster = level_calculation([random_monster_data(monster,stage)]) # Pengambilan monster bot untuk stage slanjutnya
67         # stage selanjutnya
68         Arena = True
69         win_battle = False
70         enemy_monster= new_monster[0] # Monster baru
71         potion_used = [] # battle selanjutnya potionnya reset
72         battle_status[0] = 1 # Turns awal stage selanjutnya
73         battle_status[1] = Arena # Arena masih berlangsung
74         battle_status[2] = win_battle # Belum menang pertarungan selanjutnya
75         user_monster[2],user_monster[3],user_monster[4] = base_atk,base_def,hp_max # Efek potion dan hp monster user ter reset
76         print(user_monster)
77         print("STAGE SELANJUTNYA : ")
78         print(f"=====STAGES {stage}=====")
79     elif win_battle and stage == 5 : # User telah menang hingga stage terakhir
80         Arena = False
81         print(f"%s{enemy_monster}")
82         print(f"=====STAGES {stage}=====")
83         print(f"anda telah mengalahkan final boss(monster {enemy_monster[1]})")
84         print(f"Anda mendapatkan {oc_reward} pada stage ini")
85     elif not win_battle and not Arena : # user telah kalah
86         stage -=1
87         print(f"MONSTER ANDA WAFATT ,TETAP SEMANGAT, ANDA KEREN TELAH BERHASIL MEMBANTAI HINGGA STAGES {stage}")
88
89
90     print("=====STATS=====")

```

```

91     print(f"Total hadiah : {battle_status[5]}")
92     print(f"Jumlah stage : {stage}")
93     print(f"Damage diberikan : {battle_status[3]}")
94     print(f"Damage diterima : {battle_status[4]}")
95     # update database
96     index_user = get_index_by_id(user_id,user)
97     user[index_user][4] = (int(user[index_user][4]) + battle_status[5]) # update oc
98     update_inventory_by_id(item_inventory,user_item_inventory,user_id) #update data quantity potion

```

## ● F10 - Shop & Currency

```

1     from src.F11 import get_user_inventory
2     from src.F11 import update_inventory_by_id
3     from src.F12 import merge_tables_by_id
4     from src.F12 import generate_id_potion
5     from src.validation import get_index_by_id,is_all_same_type
6     from src.F13 import table_interface
7
8     def beli(objek:str,data_shop,user_inventory,user_data,menu) :
9         # Procedure , I:5 user_inventory,oc diambil dari database . F:5 penambahan pada user_inventory dan pengurangan OC berdasarkan pembelian user
10         oc = user_data[4]
11         print(f"Jumlah O.W.C.A Coin mu sekarang {oc}\n")
12         could_purchase = True
13         id = str(input(f"Masukkan id {objek} : "))
14         index_in_shop = get_index_by_id(id,menu)
15         if objek == "potion" :
16             index_in_inventory = get_index_by_id(id,generate_id_potion(user_inventory))
17         elif objek == "monster" :
18             index_in_inventory = get_index_by_id(id,user_inventory)
19         else :
20             print("Fitur belum ada")
21             could_purchase = False
22         if index_in_shop == -999 : # i == -999 berarti index tidak ada
23             print(f"{objek} tidak terdapat pada shop")
24             could_purchase = False
25         elif index_in_inventory != -999 and objek == "monster":
26             print(f"Monster {menu[index_in_shop][1]} sudah terdapat di inventory")
27             could_purchase = False
28         if could_purchase : # purchase objek (kondisi telah tervalidasi)
29             stock = int(data_shop[index_in_shop][1])
30             price = int(data_shop[index_in_shop][2])

```



```

31         purchase = 1
32         if objek == "potion":
33             purchase = str(input("Mau beli berapa banyak potion? : "))
34             while not is_all_same_type(input=purchase,check_integer=True) :
35                 print("Input Tidak Valid, Masukkan Bilangan Bulat")
36                 purchase = str(input("Mau beli berapa banyak potion? : "))
37             purchase = int(purchase)
38             total_price = price * purchase
39             if purchase > stock :
40                 print("Stock tidak cukup")
41             elif total_price > oc :
42                 print("Saldo oc tidak cukup")
43             else : # stock tersedia dan saldo cukup
44                 user_data[4] -= total_price
45                 data_shop[index_in_shop][1] = stock-purchase
46                 print(data_shop)
47                 nama_objek = menu[index_in_shop][1]
48                 print(f"{objek} {nama_objek} berhasil di beli dengan harga {total_price}")
49                 if objek == "monster" :
50                     user_inventory.append([id,purchase])
51                 elif (objek == "potion" and index_in_inventory == -999):
52                     user_inventory.append([nama_objek,purchase])
53                 else : #objek == "potion" dan potion ada di inventory
54                     current_amount = user_inventory[index_in_inventory][1]
55                     new_amount = purchase + int(current_amount)
56                     user_inventory[index_in_inventory][1] = new_amount

```

```

60 def shop(database,user_id,user_role) : # Program Utama
61     # Inialisasi data dari database
62     user = database["user"]
63     monster = database["monster"]
64     monster_shop = database["monster_shop"]
65     monster_inventory = database["monster_inventory"]
66     item_shop = database["item_shop"]
67     item_inventory = database["item_inventory"]
68     user_items = get_user_inventory(item_inventory,user_id)
69     user_monsters = get_user_inventory(monster_inventory,user_id)
70
71     if user_role.lower() != "agent" :
72         return print("Ini shop untuk role agent")
73     index_user = get_index_by_id(user_id,user)
74     user_data = user[index_user]
75     print("Selamat datang di indoseptember, Selamat belanja :) \n")
76     is_Shopping = True
77     while is_Shopping :
78         menu_potion = generate_id_potion(item_shop)
79         menu_monster = merge_tables_by_id(monster,monster_shop)
80         aksi = str(input("Pilih aksi (lihat/beli/keluar \n:)).lower()
81         if aksi != "lihat" and aksi != "beli" and aksi != "keluar" : # cek input
82             print("Aksi yang anda berikan tidak valid, silahkan input lagi ")
83         elif aksi == "lihat" :
84             objek = str(input(("Mau lihat apa (monster/potion) : ")).lower()
85             if objek == "monster" :
86                 table_interface(menu_monster,"SHOP MONSTER KOSONG")
87             elif objek == "potion" :
88                 table_interface(menu_potion,"SHOP POTION KOSONG")
89             else :

```

```

91
92     elif aksi == "beli" :
93         objek = str(input(("Mau beli apa (monster/potion :)")).lower())
94         if objek == "monster" :
95             beli(objek="monster",data_shop=monster_shop,user_inventory=user_monsters,user_data=user_data,menu=menu_monster)
96             update_inventory_by_id(monster_inventory,user_monsters,user_id) # perbarui database monster_inventory
97         elif objek == "potion" :
98             beli(objek="potion",data_shop=item_shop,user_inventory=user_items,user_data=user_data,menu=menu_potion)
99             update_inventory_by_id(item_inventory,user_items,user_id) # perbarui database item_inventory
100
101         else :
102             print("Hanya terdapat shop potion dan monster ")
103     elif aksi == "keluar":
104         is_Shopping = False
105         print("Terimakasih sudah berbelanja")
106     else :
107         print('Fitur belum ada')
108     user[index_user] =user_data #update oc

```

## ● F11 - Laboratory

```

1  from src.validation import input_unill_range_valid
2  from src.validation import is_range_valid
3  from src.validation import is_all_same_type
4  from src.validation import input_unill_type_valid, get_index_by_id
5
6  def update_inventory_by_id(general_inventory,user_inventory,user_id:int) :
7      # I.S general_inventory dari database(general_inventory), user_inventory telah update
8      # F.S jika user_id terdapat pada general_inventory, maka row pada user_id akan diupdate dengan data user_inventory
9      # jika tidak maka akan menambahkan row baru dari row data user_inventory
10     for j in range(1,len(user_inventory)) :
11         found = False
12         user_row = user_inventory[j]
13         for i in range(1,len(general_inventory)) :
14             general_row = general_inventory[i]
15             if (general_row[0] == user_id) and (general_row[1] == user_row[0]) :
16                 general_inventory[i] = [user_id] + user_row
17                 found = True
18                 break
19         if not found :
20             new_data = [user_id] + user_row
21             general_inventory.append(new_data)
22
23     def get_user_inventory(inventory,user_id:int) :
24         # Fungsi mengeluarkan output data user dari database inventory yang memiliki user_id
25         user_inventory = [inventory[0][1:]]
26         for row in inventory[1:] :
27             if str(row[0]) == str(user_id) :
28                 user_inventory.append(row[1:])
29         return user_inventory
30

```

```

31 def get_monster_name(user_monster_inventory,table_monster) :
32     # Fungsi untuk mendapatkan nama monster yang terdapat pada user_monster_inventory , nama monster diambil dari table_monster
33     monster_names = ["Monster Names : "]
34     for i in range(1,len(user_monster_inventory)) :
35         id_monster = user_monster_inventory[i][0]
36         found = False
37         j = 0
38         while not found and j < len(table_monster):
39             if id_monster == table_monster[j][0] :
40                 monster_name = table_monster[j][1]
41                 monster_names.append(monster_name)
42                 found = True
43                 j+=1
44                 break
45             j+=1
46     return monster_names
47
48
49 def print_monster_list(user_monsters,monster_names) : # Procedure cetak monster list
50     print("===== MONSTER LIST =====")
51     for i in range(1,len(user_monsters)) :
52         | print(f"{i}. {monster_names[i]} (Level : {user_monsters[i][1]})")
53     print()
54
55 def print_upgrade_price (upgrade_prices) : # Procedure cetak harga upgrade level
56     for i in range(4):
57         | print(f"Level {i+1} -> Level{i+2} : {upgrade_prices[i]}")
58     print()
59
60 def upgrading_monster(monster_level:int,monster_name:str,oc:int,upgrade_prices) :
61     # I.S monster level belum terupgrade atau sudah maksimum (level 5)
62     # F.S Jika awalnya monster level belum max maka monster level terupgrade sampai keinginan user dengan level maks,yaitu level 5
63     while True :
64         upgrade_price = upgrade_prices[monster_level-1]
65         if monster_level == 5 :
66             print(f"Level {monster_name} sudah maksimal")
67             break
68         if upgrade_price > oc :
69             print("OC anda tidak cukup")
70             break
71
72         print(f"{monster_name} akan di upgrade ke level {monster_level+1}")
73         print(f"Harga untuk melakukan upgrade adalah {upgrade_price}")
74         next_upgrade = str(input("Lanjutkan upgrade (y/n) : ")).lower()

```

```

75         if next_upgrade.lower() == "y" :
76             oc -= int(upgrade_price)
77             monster_level += 1
78             print(f"Selamat, {monster_name} berhasil diupgrade menjadi level {monster_level}")
79             next_upgrade = str(input("Lanjutkan upgrade (y/n) : ")).lower()
80             if next_upgrade.lower() == "n" :
81                 print("Upgrade selesai")
82                 break
83             elif next_upgrade.lower() != "y" and next_upgrade.lower() != "n":
84                 print("error,input salahh")
85                 break
86
87         elif next_upgrade == "n" :
88             print("Upgrade dibatalkan")
89             break
90         else:
91             print("erorr, input salah")
92             break
93
94     return monster_level,oc
95 def laboratory(database,user_id,user_role) : # Program Utama
96     if user_role.lower() != "agent" : # cek role
97         return print("Anda bukan agent")
98     # Pengambilan data
99     monster_inventory = database["monster_inventory"]
100     monster = database["monster"]
101     user = database["user"]
102     index_user = get_index_by_id(user_id,user) # index letak data user di database user
103     oc = user[index_user][4] # oc yang dimiliki user
104     upgrade_prices = [300,500,800,1000] #hardcode harga upgrade
105
106
107     user_monsters = get_user_inventory(monster_inventory,user_id) # inventory item user
108     monster_names = get_monster_name(user_monsters,monster) # nama monster user, terurut sesuai pada user monsters inventory
109
110     print("SELAMAT DATANG DI LAB DR ASEPP")
111     islaboratoring = True
112     while islaboratoring :
113         print_monster_list(user_monsters,monster_names) # monster list name ebrdasarkan database monster
114         print_upgrade_price(upgrade_prices) # harga upgrade tiap upgrde level
115         banyak_monster = len(monster_names)-1
116         index_monster = int(input(f"Pilih monster (1 - {banyak_monster}) : "))
117
118         # check apakah user telah memasukkan pilihan monster yang benar sesuai user monsters dan check apakah masukan user merupakan integer
119         while not is_range_valid(str(index_monster),banyak_monster+1) or not is_all_same_type(str(index_monster),check_integer=True):
120             index_monster = input_untill_type_valid(str(index_monster), message="Input harus berupa integer",type="integer")
121             index_monster= input_untill_range_valid(index_monster,len(monster_names),f"Range pilihan tidak valid \nPilih monster (1-{len(monster_names)}): ")
122         index_monster = int(index_monster)
123         monster_level = user_monsters[index_monster][1]
124         monster_name = monster_names[index_monster]
125         monster_level,oc = upgrading_monster(int(monster_level),monster_name,int(oc),upgrade_prices) # akan mengupgrade monster
126         user_monsters[index_monster][1] = monster_level # update level monster pada user monsters(inventory)
127         next_monster = str(input("Ingin mengupgrade monster lain (y/n)?")).lower()
128         if next_monster == "n" :
129             user[index_user][4] = oc
130             update_inventory_by_id(monster_inventory,user_monsters,user_id) # update database monster inventory general dengan user inventory
131             islaboratoring = False

```

## ● F12 - Shop Management

```

1  from src.validation import *
2
3
4
5  def filter_table_by_id(table1, table2):
6      new_table = []
7      new_table.append(table1[0])
8      for i in range(1, len(table1)): # Mulai dari baris kedua
9          elemen_exist = False
10         for j in range(1, len(table2)): # Mulai dari baris kedua
11             if table1[i][0] == table2[j][0]:
12                 elemen_exist = True
13                 break
14
15         if not elemen_exist:
16             new_table.append(table1[i])
17     return new_table
18
19  def table_interface(table, message):
20      # Cari panjang maks tiap kolom
21      column_length = []
22      if len(table) == 1 :
23          print(message)
24      else :
25          for col in range(len(table[0])):
26              max_column_length = max(len(str(row[col])) for row in table)
27              column_length.append(max_column_length)
28          # Cetak tabel
29          for row in table:
30              for i in range(len(row)):
31                  item = row[i]

```

```

34         print()
35  def merge_tables_by_id(table1, table2) :
36      header = table1[0] + table2[0][1:]
37      merged_table = [header]
38
39      for row1 in table1[1:]:
40          id1 = row1[0]
41          for row2 in table2[1:]:
42              id2 = row2[0]
43              if id1 == id2:
44                  merged_table.append(row1 + row2[1:])
45                  break
46      return merged_table
47  def generate_id_potion(table_potion) :
48      new_table = [["ID"] + table_potion[0]]
49      i = 1
50      for row in table_potion[1:] :
51          new_table.append([str(i)] + row)
52          i += 1
53      return new_table
54
55
56  def delete_row(indeks: int, table) :
57      indeks_maks = len(table) - 1
58      if indeks_maks < indeks or indeks < 0 :
59          print("indeks tidak valid")
60          return table
61      else :
62          new_table = []
63          for i in range(indeks_maks + 1):

```

```

64         if i != indeks:
65             new_table.append(table[i])
66         return new_table
67
68     def input_data(objek:str,status_input:str,shop_data) ->int :
69         stock,harga = "0","0" # status_input == hapus
70         id = str(input(f"Masukkan id {objek} : ")) # Pada potion masukkan type
71         while (get_index_by_id(id,shop_data) == -999) :
72             id = str(input(f"Id tidak terdapat pada shop \n Masukkan id {objek} : "))
73
74         if status_input != "hapus" :
75             stock = str(input(f"Masukkan stok {status_input} : "))
76             while not is_all_same_type(stock,check_integer=True) or not is_range_valid(stock,99999) :
77                 stock = input_untill_type_valid(stock,f"Masukkan stock {status_input} :",type="integer")
78             stock = input_untill_range_valid(stock,99999,f"Input Tidak Valid, stock harus >= 0\nMasukkan stok{status_input} :")
79         if status_input != "hapus" :
80             harga = str(input(f"Masukkan harga {status_input} : "))
81             while not is_all_same_type(harga,check_integer=True) or not is_range_valid(harga,99999) :
82                 harga = input_untill_type_valid(harga,message=f"Masukkan harga {status_input} :",type="integer")
83             harga = input_untill_range_valid(harga,99999,message=f"Input Tidak Valid, Harga harus >= 0\nMasukkan harga {status_input} :")
84         return int(stock),int(harga),str(id)
85
86     def shop_management(database,user_role) :
87         monster = database['monster']
88         monster_shop = database['monster_shop']
89         item_shop = database['item_shop']
90         if user_role.lower() != "admin" : # cek akses
91             print("Anda tidak memiliki akses ke shop management")
92         return monster_shop,item_shop
93

```

```

94     isManaging = True
95     while isManaging :
96
97         menu_monsters = merge_tables_by_id(monster,monster_shop)
98         menu_items = generate_id_potion(item_shop)
99         aksi = str(input("Pilih aksi(lihat/tambah/ubah/hapus/keluar) :")).lower()
100        if aksi == "lihat" :
101            objek = str(input(f"Mau {aksi} apa?(monster/potion)?")).lower()
102            if objek == "monster" :
103                table_interface(menu_monsters,"TIDAK ADA MONSTER DI DALAM SHOP")
104            elif objek == "potion" :
105                table_interface(menu_items,"TIDAK ADA POTION DI DALAM SHOP")
106            else :
107                print("Input salah")
108        elif aksi.lower() == "tambah" :
109            objek = str(input(f"Mau {aksi} apa?(monster/potion)?")).lower()
110            if objek == "monster" :
111                not_added = filter_table_by_id(monster,monster_shop)
112                if len(not_added) == 1 :
113                    print("Semua jenis monster telah ada di di shop")
114                else :
115                    table_interface(not_added,"")
116                stock,harga,id_monster = input_data(objek,"awal",not_added)
117                monster_shop.append([int(id_monster),stock,harga])
118
119            elif objek == "potion" :
120                potion = [{"type"},["strength"],["resilience"],["healing"]] #jenis potion hardcode
121                not_added= (filter_table_by_id(potion,item_shop))
122                id_not_added = generate_id_potion(not_added)

```

```

126         table_interface(id_not_added,"")
127         stock ,harga ,id_potion = input_data(objek,"awal",not_added)
128         item_shop.append([id_potion,stock,harga])
129     else :
130         print("input tidak valid")
131     elif aksi == "ubah" :
132         objek = str(input(f"Mau {aksi} apa?(monster/potion?)).lower())
133         if objek == "monster" :
134             if len(monster_shop) == 1 :
135                 print("Belum terdapat monster apapun pada monster shop, silahkan tambahkan terlebih dahulu pada fitur tambah")
136             else :
137                 stock ,harga ,id_monster = input_data(objek,"baru",menu_monsters)
138                 index_monster = get_index_by_id(id_monster,monster_shop)
139                 monster_shop[index_monster] = [id_monster,stock,harga]
140
141         elif objek == "potion" :
142             if len(item_shop) == 1 :
143                 print("Belum terdapat potion apapun pada potion shop, silahkan tambahkan terlebih dahulu pada fitur tambah")
144             else :
145                 stock,harga, id_potion = input_data(objek,"baru",menu_items)
146                 index_potion = id_potion
147                 item_shop[index_potion][1] = stock
148                 item_shop[index_potion][2] = harga
149
150         else :
151             print("input salah")
152     elif aksi == "hapus" :
153         objek = str(input("Mau hapus apa min ? (monster/potion?)).lower())
154         if objek == "monster" :
155             if len(monster_shop) == 1 :
156                 print("MONSTER PADA SHOP MONSTER TELAH KOSONG")
157             else :
158                 table_interface(menu_monsters,"")
159
160                 _,id_monster = input_data(objek,"hapus",menu_monsters)
161                 yakin = str(input("Yakin mengubah (y/n) ").lower())
162                 if yakin == "y" :
163                     index_monster = get_index_by_id(id_monster,monster_shop)
164                     monster_shop.delete_row(index_monster,monster_shop)
165                 elif yakin == "n" :
166                     print("Tidak jadi mengubah")
167                 else :
168                     print("input salah")
169
170         elif objek == "potion" :
171             if len(item_shop) == 1 :
172                 print("POTION PADA SHOP POTION TELAH KOSONG")
173             else :
174                 table_interface(menu_items, "POTION PADA SHOP POTION TELAH KOSONG")
175                 _,id_potion = input_data(objek,"hapus",menu_items)
176                 yakin = str(input("Yakin mengubah (y/n) ").lower())
177                 if yakin == "y" :
178                     print(item_shop)
179                     item_shop = delete_row(int(id_potion),item_shop)
180                     print(item_shop)
181
182                 else :
183                     print("Tidak jadi mengubah")
184
185         else :
186             print("input salah")
187     elif aksi == "keluar" :
188         isManaging = False
189         print("terimakasih")
190     else :
191         print("Fitur belum ada")
192     return monster_shop,item_shop
193

```

## ● F13 - Monster Management

```

1
2 from src.validation import *
3 from src.F12 import table_interface
4
5 def is_monster_exist(monster:str,table_monster:list[list])>-> bool :
6     # Fungsi mengecek apakah monster merupakan bagian dari database monster
7     found = False
8     i = 1
9     while not found and (i<len(table_monster)) :
10         if monster.lower() == table_monster[i][1].lower() :
11             found = True
12             i +=1
13     return found
14
15 def input_monster_not_exist(monster:str,table_monster)>->str :
16     # Fungsi meminta user untuk memberi masukan monster yang belum ada di database monster
17     while is_monster_exist(monster,table_monster) :
18         monster = str(input("Masukkan monster yang belum ada : "))
19     return monster
20
21 def monster_management(database,user_role:str) :
22
23
24     if user_role.lower() == "admin" : # cek akses
25         isManaging = True
26         tabel_monster = database['monster']
27
28         while isManaging :
29             print("SELAMAT DATANG DI DATABASE PARA MONSTER !!!\n1. Tampilkan semua Monster \n2. Tambah Monster baru\n3.Cancel")
30             aksi = str(input("Pilih aksi(1/2) :"))

```

```

31         if aksi == "1" :
32             table_interface(tabel_monster,"DATAABASE MONSTER KOSONG")
33         elif aksi == "2" :
34             nama_monster = str(input("Masukkan type/nama monster :"))
35             # Meminta user untuk memberikan masukan nama monster dengan type string dan belum ada di database
36             while not is_all_same_type(nama_monster,check_integer=False) or is_monster_exist(nama_monster,tabel_monster) :
37                 nama_monster = input_unill_type_valid(nama_monster,message="Masukkan type/nama monster : ",type="string")
38             nama_monster = input_monster_not_exist(nama_monster,tabel_monster)
39
40             attack_power = str(input("Masukkan attack power : "))
41             # Meminta user untuk memberikan masukan attack power dengan type int
42             while not is_all_same_type(attack_power,check_integer=True) :
43                 attack_power = input_unill_type_valid(attack_power,message="Masukkan attack power : ",type="integer")
44
45             def_power = str(input("Masukkan def power (range 0 - 50) : "))
46             # Meminta user untuk memberikan masukan adef power dengan type int dan dengan range 0-50
47             while not is_all_same_type(def_power,check_integer=True) or not is_range_valid(def_power,51) :
48                 def_power = input_unill_type_valid(def_power,message="Masukkan def_power (range 0 - 50) : ",type="integer")
49             def_power = input_unill_range_valid(def_power,51,"Masukkan range def yang benar (0-50) : ")
50
51             hp_monster = str(input("Masukkan HP monster : "))
52             # Meminta user untuk memberikan masukan hp dengan type int
53             while not is_all_same_type(hp_monster,check_integer=True) :
54                 hp_monster = input_unill_type_valid(hp_monster,message="Masukkan HP monster : ",type="integer")
55
56             id_monster = str(len(tabel_monster)) # id monster baru terurut
57             new_monster = [int(id_monster),nama_monster,int(attack_power),int(def_power),int(hp_monster)]
58
59             # cetak monster yang berhasil dibuat
60             print("\n \n Monster baru telah berhasil dibuat")

```



```

61         print(f"Type : {nama_monster} ")
62         print(f"ATK Power : {attack_power}")
63         print(f"DEF Power : {def_power}")
64         print(f"HP : {hp_monster}\n")
65
66         tambahkan = str(input("Tambahkan monster ke database(Y/N) : ")).lower()
67         if tambahkan == "y" :
68             isManaging = False
69             tabel_monster.append(new_monster)
70             print("Monster berhasil ditambahkan")
71             next_command = str(input("Ingin menambahkan monster lain :?"))
72             if next_command == "y" :
73                 isManaging = True
74             else : # tidak jadi ditambahkan
75                 print("Monster gagal ditambahkan")
76         if aksi == "3" :
77             isManaging = False
78         else :
79             print("Input tidak valid")
80     else :
81
82         print("Anda tidak memiliki akses pada monster management" )

```

## ● F14 - Load

```

1  import sys
2  import argparse
3  import os
4  import time
5  from src.utilityfunction import read_csv
6  from src.validation import check_path_validity
7
8  def integer_adjustment (tables,column_exception) :
9      for i in range(1,len(tables)) :
10         for j in range(len(tables[0])) :
11             if j not in column_exception :
12                 tables[i][j] = int(tables[i][j])
13         return tables
14
15
16
17  def converting(folder_name:str)-> dict :
18
19      path = "C:/Users/ASUS/TUBESDAS/" + folder_name
20
21      if check_path_validity(path):
22          print("LOADING...")
23          time.sleep(2)
24          # Baca csv ubah ke tables , type blm di adjust (masih string)
25          user = read_csv(path + "/user.csv")
26          monster = read_csv(path + "/monster.csv")
27          monster_shop = read_csv(path + "/monster_shop.csv")
28          monster_inventory = read_csv(path+"/monster_inventory.csv")
29          item_shop = read_csv(path + "/item_shop.csv")
30          item_inventory =read_csv(path + "/item_inventory.csv")
31

```

```

33     database = {
34         "user" : integer_adjustment(user,[1,2,3]),
35         "monster" : integer_adjustment(monster,[1]),
36         "monster_shop" : integer_adjustment(monster_shop,[1]),
37         "monster_inventory" : integer_adjustment(monster_inventory,[1]),
38         "item_shop" : integer_adjustment(item_shop,[0]),
39         "item_inventory" : integer_adjustment(item_inventory,[1])
40     }
41     elif folder_name == "":
42         print("Tidak ada nama folder diberikan!")
43         #program berhenti
44     else:
45         print(f"Folder {folder_name} tidak ditemukan!")
46         #program berhenti
47     return database
48
49
50 def load () :
51     parser = argparse.ArgumentParser()
52     parser.add_argument("name", type=str, help="name of saved game", nargs="?", const='')
53     args = parser.parse_args()
54     folder = args.name
55
56     if folder == None:
57         print('Tidak ada nama folder yang diberikan!\nUsage : python main.py <nama folder>\nJika kamu belum pernah bermain, silahkan masukkan "python main.py database.'.')
58         sys.exit()
59     else:
60         path = "C:/Users/ASUS/TUBESDAS/" + folder
61         if (os.path.exists(path)):

```

```

62         database = converting(folder)
63     else:
64         print(f"Folder {folder} tidak ditemukan!")
65         sys.exit()
66     print('Selamat datang di Program OWCA!\nMasukkan command "HELP" untuk melihat daftar command yang dapat kamu panggil.')
67     return database

```

## ● F15 - Save

```

1     import os
2     import time
3     from src import utilityfunction as uf
4
5     def save(temp_data):
6         folder_name = input("Masukkan nama folder: ")
7         path = "C:/Users/ASUS/TUBESDAS/" + folder_name
8         if not(os.path.exists(path)):
9             os.mkdir(path)
10            print("Saving...\n")
11            time.sleep(2)
12            print(f"Membuat folder {folder_name}")
13            time.sleep(1)
14            print(f"Berhasil menyimpan data di folder {folder_name}!")
15        else:
16            print("Saving...\n")
17            time.sleep(2)
18            print(f"Berhasil menyimpan data di folder {folder_name}!")
19        uf.array_to_csv(temp_data["user"], path + "/user.csv")
20        uf.array_to_csv(temp_data["monster"], path + "/monster.csv")
21        uf.array_to_csv(temp_data["monster_shop"], path + "/monster_shop.csv")
22        uf.array_to_csv(temp_data["monster_inventory"], path + "/monster_inventory.csv")
23        uf.array_to_csv(temp_data["item_inventory"], path + "/item_inventory.csv")
24        uf.array_to_csv(temp_data["item_shop"], path + "/item_shop.csv")

```

## ● F16 - Exit

```

1     from src.F15 import save
2     import sys
3
4     def exit(database):
5         validation = input("Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) ")
6
7         while not ((validation.upper() == "Y") or (validation.upper() == "N")):
8             validation = input("Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) ")
9
10        if validation.upper() == "Y":
11            save(database)
12            sys.exit()
13        else:
14            sys.exit()

```

## ● B03 - Monster Ball

```

def MonsterBall (enemy_array:list)->bool:
    # monsterball
    success = False
    level = enemy_array[5]
    number = randomNumberGenerator(0,100)
    if(level == 1):
        if(number <= 75):
            success = True
    if(level == 2):
        if(number <= 50):
            success = True
    if(level == 3):
        if(number <= 25):
            success = True
    if(level == 4):
        if(number <= 10):
            success = True
    if(level == 5):
        if(number <= 5):
            success = True
    if (success == True):
        print("Swoosshhhh, Anda mengeluarkan Monster Ball !!!")
        print("Selamat, Anda berhasil mendapatkan Monster",enemy_array[1])
        print("Name          :",enemy_array[1])
        print("ATK Power       :",enemy_array[2])
        print("DEF Power       :",enemy_array[3])
        print("HP              :",enemy_array[4])
        print("Level           :",enemy_array[5])
    else:
        print("Swoosshhhh, Anda mengeluarkan Monster Ball !!!")
        print("Yahhh, Anda belum berhasil mendapatkan monster Zuko !!!")

```

● **B04 - Jackpot**

```

1 # Bonus function Jackpot
2 from src.validation import get_index_by_id
3 from src.F00 import randomNumberGenerator as rng
4
5 def Jackpotgraphics(number):
6     if(number == 1):
7         print("""
8
9             .: .:!!
10              "q#?"
11              .a#00000000..
12              .a#0000000000:
13              '0000000000'
14              '000000000'
15              '0000000' """)
16
17         if(number == 2):
18             print("""
19
20                 i#
21                 "q#..
22                 0001
23                 .a#0000:
24                 i#000000:
25                 '000000'
26                 '00000' """)
27
28         if(number == 3):
29             print("""
30
31                 ..
32                 .a#000..
33                 ./' 00000
34                 | 'q ""
35                 .a#0000000.
36                 .a#00000000:
37
38             if(number == 4):
39                 print("""
40
41                     i#0000..
42                     .... '00000'
43                     i#0000000000..
44                     '0000000000:
45                     '000000000'
46                     a# " 'q# "" """)
47
48             if(number == 5):
49                 print("""
50
51                     00000000
52                     00"0000"
53                     .0000
54                     0000:
55                     0000 """)
56
57         return(" ")
58
59 def Jackpot (user_id,user_role,database):
60     if user_role.lower() != "agent" :
61         return print("anda bukan agent kami")
62     user = database["user"]
63     index_user = get_index_by_id(user_id,user)
64     owcacoins = user[index_user][4]
65     item = [{"apple",5},{"pear",10},{"cherry",20},{"cherry",50},{"SEVEN",100}]
66     print("=====")
67     print("
68
69             WELCOME TO JACKPOT ")
70     print("=====")
71     print("1. Apple      = 5 oc")

```



## Lampiran Hasil Pengujian Program

Fitur	Hasil Pengujian
<u>F00 - RNG</u>	<p>1. Random Number 5 data dari range 1 sampai 10002</p> <pre>59 59 59 59 59</pre>
<u>F01 - Register</u>	<p>1. Username telah terpakai</p> <pre>&gt;&gt;&gt;&gt;&gt; register Masukkan username: adndax Masukkan password: miaw  Username adndax sudah terpakai, silahkan gunakan username lain! &gt;&gt;&gt;&gt;&gt; █</pre> <p>2. Username tidak valid</p> <pre>&gt;&gt;&gt;&gt;&gt; register Masukkan username: miaw! Masukkan password: miawmiaw  Username hanya boleh berisi alfabet, angka, underscore, dan strip! &gt;&gt;&gt;&gt;&gt; █</pre> <p>3. Username valid</p> <pre>&gt;&gt;&gt;&gt;&gt; register Masukkan username: hilo_miaw Masukkan password: miawmiaw123456  Silahkan pilih salah satu monster sebagai monster awalmu. 1. Centaur 2. Roga 3. Freya 4. Rika 5. Shrek  Monster pilihanmu: 5 Selamat datang Agent hilo_miaw. Maria kita mengalahkan Dr. Asep Spakbor dengan Shrek!</pre>
<u>F02 - Login</u>	<p>1. Username valid dan password salah</p> <pre>&gt;&gt;&gt;&gt;&gt; login Username: adndax Password: miaw Password salah!</pre>

	<p>2. Username tidak terdaftar</p> <pre> &gt;&gt;&gt;&gt;&gt; login Username: hm_miaw Password: miaw Username tidak terdaftar! </pre> <p>3. Username valid dan password benar</p> <pre> &gt;&gt;&gt;&gt;&gt; login Username: adndax Password: @Fjr0311 Selamat datang, Admin adndax! Masukkan command "HELP" untuk daftar command yang dapat kamu panggil. </pre> <p>4. Sudah login</p> <pre> &gt;&gt;&gt;&gt;&gt; login Login gagal! Anda telah login dengan username hi_miaw silahkan lakukan "LOGOUT" sebelum melakukan login kembali. &gt;&gt;&gt;&gt;&gt;  </pre>
<u>F03 - Logout</u>	<p>1. Sudah login</p> <pre> &gt;&gt;&gt;&gt;&gt; logout logout telah berhasil </pre> <p>2. Belum login</p> <pre> &gt;&gt;&gt;&gt;&gt; logout Logout gagal! Anda belum login, silahkan login terlebih dahulu sebelum melakukan logout. &gt;&gt;&gt;&gt;&gt;  </pre>
<u>F04 - Menu &amp; Help</u>	<p>1. Belum login</p> <pre> &gt;&gt;&gt;&gt;&gt; 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 &gt;&gt;&gt;&gt;&gt;  </pre> <p>2. Admin</p> <pre> &gt;&gt;&gt;&gt;&gt; help ===== HELP ===== Selamat datang, Admin adndax Berikut adalah hal-hal yang dapat kamu lakukan: 1. Logout: Keluar dari akun yang sedang digunakan 2. Shop Management: Melakukan manajemen pada SHOP sebagai tempat jual beli peralatan Agent 3. Monster Management : Menambahkan monster pada database 4. Save : Menyimpan data game 5. Exit : Keluar dari game Footnote: 1. Untuk menggunakan aplikasi, masukkan nama fungsi yang terdaftar 2. Pastikan memasukkan input yang valid &gt;&gt;&gt;&gt;&gt;  </pre>

	<h3>3. Agent</h3> <pre> &gt;&gt;&gt;&gt; help ===== HELP ===== Halo Agent hilo_miaw .Kamu memanggil command HELP. Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian. Berikut adalah hal-hal yang dapat kamu lakukan sekarang: 1. Logout : Keluar dari akun yang sedang digunakan 2. Inventory : Melihat monsters dan items yang dimiliki Agent 3. Owcadex : Membuka OWCADEX untuk melihat list monsters 4. Battle : Melakukan battle dengan monster lain 5. Arena : Menginisiasi Arena untuk melatih monster 6. Shop : Membuka shop untuk membeli items 7. Laboratory : Menaikan level monster yang dimiliki 8. Jackpot : Melakukan gacha 9. Save : Menyimpan data game 10. Exit : Keluar dari game Footnote: 1. Untuk menggunakan aplikasi, masukkan nama fungsi yang terdaftar 2. Pastikan memasukkan input yang valid </pre>
<p><u>F05 -</u> <u>Monster</u></p>	<h3>1. Admin</h3> <pre> &gt;&gt;&gt;&gt; owcadex Owcadex hanya ditujukan untuk Agent. Silahkan masukkan command "HELP" untuk melihat daftar command yang dapat kamu panggil. &gt;&gt;&gt;&gt; </pre> <h3>2. Agent</h3> <pre> &gt;&gt;&gt;&gt; owcadex  =====    OWCADEX     =====  Index 1 ----- Monster Type : Centaur Attack Power : 60 Defense Power : 6 Monster Health : 75 ===== Index 2 ----- Monster Type : Roga Attack Power : 25 Defense Power : 25 Monster Health : 100 ===== Index 3 ----- Monster Type : Freya Attack Power : 70 Defense Power : 40 Monster Health : 80 ===== Index 4 ----- Monster Type : Rika Attack Power : 100 Defense Power : 50 </pre>



<u>F06 - Potion</u>	<pre>Contoh monster yang digunakan ['4', 'Zuko', '100', '25', '800'] Atk power sebelum potion : 100.0 Def power sebelum potion : 25.0 Health sebelum potion : 60 ATK POWER SETELAH POTION 105.0 DEF POWER SETELAH POTION 26.25 HEALTH SETELAH POTION 75.0</pre>
<u>F07 - Inventory</u>	<pre>Input Command INVENTORY ===== INVENTORY LIST (User ID : 2) ===== Jumlah O.W.C.A. Coin-mu sekarang 100 1. Monster      (Name: Pikachow , Lvl: 1 , HP: 726.0) 2. Potion       (Type: strength , Qty: 5) 3. Potion       (Type: resilience , Qty: 3) Ketikkan id untuk menampilkan id item : 1 Monster Name           : Pikachow ATK Power      : 151.25 DEF Power      : 12.1 HP             : 726.0 Level          : 1 Ketikkan id untuk menampilkan id item : 2 Potion Type           : strength Quantity       : 5 Ketikkan id untuk menampilkan id item : KELUAR</pre>
<u>F08 - Battle</u>	<pre>&gt;&gt; Monster tidak ada =====MONSTER LIST===== 1. Pikachow (level 2) pilih monster :6 Monster tidak ada, Masukkan pilihan monster yang ada ! &gt;&gt; Monster ada</pre>

pilih monster : 1



HUFTTT Agen telah mengeluarkan monster Pikachow

Monster Type : Pikachow

Attack Power : 247

Defense Power : 12

Monster Health : 720

Level : 2

>> Attack

```
Pilih perintah : 1
Monster Type   : Bulbu
Attack Power   : 60
Defense Power  : 60
Monster Health : 619
Level : 2
=====
WHOOZZZ Monster Bulbu menyerang balik anda
===== Turns 1 (Bulbu)=====
Monster Type   : Pikachow
Attack Power   : 247
Defense Power  : 12
Monster Health : 662
Level : 2
=====
===== Turns 2 (Pikachow)=====
1. Attack
2. Use Potion
3. Monsterball
4. Quit
```

>> Potion ada dan belum digunakan

Pilih type : 2

Setelah meminum ramuan ini, def power meningkat menjadi 12

>> Potion tidak ada

```
=====POTION LIST=====
1.Strength Potion (Quantity : 5)----Increase ATK Power
2.Resilience Potion (Quantity : 3)----Increase DEF pow
3.Healing  Potion (Quantity : 0)---Restore Health
4.Cancel
=====
```

Pilih type : 3

>> Potion ada tetapi sudah digunakan

```
Pilih type : 2
Udah minum resilience bro
```

>> Monsterball

	<pre> Swoosshhhh, Anda mengeluarkan Monster Ball !!! Yahhh, Anda belum berhasil mendapatkan monster Zuko ! Monster Type    : Bulbu Attack Power    : 60 Defense Power    : 60 Monster Health   : 95 Level : 2 ===== WHOOOZZZ Monster Bulbu menyerang balik anda &gt;&gt; Perintah tidak ada Pilih perintah : wekww Fitur perintah tersebut belum ada </pre>
<u>F09 - Arena</u>	<pre> &gt;&gt; Perintah invalid Pilih perintah : 4 Fitur perintah tersebut belum ada &gt;&gt; Validasi potion Pilih type : 2 Setelah meminum ramuan ini, def power meningkat menjadi 12 =====POTION LIST===== 1.Strength Potion (Quantity : 5)----Increase ATK Power 2.Resilience Potion (Quantity : 3)----Increase DEF pow 3.Healing  Potion (Quantity : 0)---Restore Health 4.Cancel ===== Pilih type : 3 </pre>
<u>F10 - Shop &amp; Currency</u>	<pre> &gt;&gt; "Lihat" merupakan aksi yang terdaftar Pilih aksi (lihat/beli/keluar) :lihat Mau lihat apa (monster/potion) : &gt;&gt; "sksks" bukan merupakan aksi yang terdaftar Pilih aksi (lihat/beli/keluar) :sksks Aksi yang anda berikan tidak valid, silahkan input lagi &gt;&gt; Monster yang memiliki id 6 tidak ada </pre>

	<pre> Mau beli apa (monster/potion) :monster Jumlah O.W.C.A Coin mu sekarang 748  Masukkan id monster : 6 monster tidak terdapat pada shop &gt;&gt; Monster ada pada shop dan oc user cukup Masukkan id monster : 2 monster Bulbu berhasil di beli dengan harga 700 &gt;&gt; Potion ada pada shop tetapi oc user tidak cukup Mau beli apa (monster/potion) :potion Jumlah O.W.C.A Coin mu sekarang 48  Masukkan id potion : 1 Mau beli berapa banyak potion? : 10 Saldo oc tidak cukup </pre>
<p>F11 - Laboratory</p>	<pre> &gt;&gt; Hanya terdapat 1 monster  SELAMAT DATANG DI LAB DR ASEP ===== MONSTER LIST ===== 1. Pikachow (Level : 2)  Level 1 -&gt; Level2 : 300 Level 2 -&gt; Level3 : 500 Level 3 -&gt; Level4 : 800 Level 4 -&gt; Level5 : 1000  Pilih monster (1 - 1) : 2 Range pilihan tidak valid &gt;&gt; Range input valid </pre>



	<p>range pilihan tidak valid</p> <p>Pilih monster (1-2: )1</p> <p>Pikachow akan di upgrade ke level 3</p> <p>Harga untuk melakukan upgrade adalah 500</p> <p>&gt;&gt; Perintah tidak valid</p> <p>Lanjutkan upgrade (y/n) : slks</p> <p>erorr, input salah</p> <p>&gt;&gt; Perintah valid namun oc tidak cukup</p> <p>Selamat, Pikachow berhasil diupgrade menjadi level</p> <p>Lanjutkan upgrade (y/n) : y</p> <p>OC anda tidak cukup</p>																																																															
F12 - Shop Managemen t	<p>1. Input aksi salah</p> <p>Pilih aksi(lihat/tambah/ubah/hapus/keluar) :nds</p> <p>Fitur belum ada</p> <p>2. Input aksi benar</p> <p>Mau lihat apa?(monster/potion)?monster</p> <table><tr><th>id</th><th>type</th><th>atk_power</th><th>def_power</th><th>hp</th><th>stock</th><th>price</th></tr><tr><td>1</td><td>Pikachow</td><td>206</td><td>10</td><td>600</td><td>10</td><td>500</td></tr><tr><td>2</td><td>Bulbu</td><td>50</td><td>50</td><td>600</td><td>4</td><td>700</td></tr><tr><td>4</td><td>Zuko</td><td>100</td><td>25</td><td>800</td><td>8</td><td>550</td></tr><tr><td>5</td><td>Chacha</td><td>80</td><td>30</td><td>700</td><td>7</td><td>600</td></tr></table> <p>3. Validasi id monster/type potion pada database sukses</p> <p>Masukkan id monster : 2</p> <p>Yakin mengubah (y/n) y</p> <p>Pilih aksi(lihat/tambah/ubah/hapus/keluar) :lihat</p> <p>Mau lihat apa?(monster/potion)?monster</p> <table><tr><th>id</th><th>type</th><th>atk_power</th><th>def_power</th><th>hp</th><th>stock</th><th>price</th></tr><tr><td>1</td><td>Pikachow</td><td>206</td><td>10</td><td>600</td><td>10</td><td>500</td></tr><tr><td>4</td><td>Zuko</td><td>100</td><td>25</td><td>800</td><td>8</td><td>550</td></tr><tr><td>5</td><td>Chacha</td><td>80</td><td>30</td><td>700</td><td>7</td><td>600</td></tr></table> <p>4. Valdasi id monster/type potion pada database gagal</p> <p>Masukkan id monster : 4</p> <p>Id tidak terdapat pada tabel</p> <p>5. Validasi input type</p>	id	type	atk_power	def_power	hp	stock	price	1	Pikachow	206	10	600	10	500	2	Bulbu	50	50	600	4	700	4	Zuko	100	25	800	8	550	5	Chacha	80	30	700	7	600	id	type	atk_power	def_power	hp	stock	price	1	Pikachow	206	10	600	10	500	4	Zuko	100	25	800	8	550	5	Chacha	80	30	700	7	600
id	type	atk_power	def_power	hp	stock	price																																																										
1	Pikachow	206	10	600	10	500																																																										
2	Bulbu	50	50	600	4	700																																																										
4	Zuko	100	25	800	8	550																																																										
5	Chacha	80	30	700	7	600																																																										
id	type	atk_power	def_power	hp	stock	price																																																										
1	Pikachow	206	10	600	10	500																																																										
4	Zuko	100	25	800	8	550																																																										
5	Chacha	80	30	700	7	600																																																										

	<p>Masukkan stok awal : osjs</p> <p>Input harus bertipe integer</p>
<p>F13 - Monster Managemen t</p>	<p>1. Admin</p> <pre> &gt;&gt;&gt;&gt;&gt; monster management SELAMAT DATANG DI DATABASE PARA MONSTER !!! 1. Tampilkan semua Monster 2. Tambah Monster baru 3.Cancel Pilih aksi(1/2) :1 id   type        atk_power   def_power   hp   1    Centaur     60          6           75   2    Roga        25          25          100  3    Freya       70          40          80   4    Rika        100         50          100  5    Shrek       999         999         999  Input tidak valid SELAMAT DATANG DI DATABASE PARA MONSTER !!! 1. Tampilkan semua Monster 2. Tambah Monster baru 3.Cancel Pilih aksi(1/2) :2 Masukkan type/nama monster :Kaneki Masukkan attack power : 100 Masukkan def power (range 0 - 50) : 60 Masukkan range def yang benar (0-50) : 30 Masukkan HP monster : 99      Monster baru telah berhasil dibuat Type : Kaneki ATK Power : 100 DEF Power : 30 HP : 99  Tambahkan monster ke database(Y/N) : y Monster berhasil ditambahkan </pre> <p>2. Agent</p> <pre> &gt;&gt;&gt;&gt;&gt; monster management Anda tidak memiliki akses pada monster management </pre>
<p>F14 - Load</p>	<p>1. User tidak memasukkan nama folder</p> <pre> PS C:\Users\ASUS\TUBESDAS&gt; &amp; C:\Users\ASUS\AppData\Local\Programs\Python\Python311\python.exe c:/Users/ASUS/TUBESDAS/main.py Tidak ada nama folder yang diberikan! Usage : python main.py &lt;nama folder&gt; Jika kamu belum pernah bermain, silahkan masukkan "python main.py database". </pre> <p>2. User memasukkan nama folder yang tidak valid</p> <pre> PS C:\Users\ASUS\TUBESDAS&gt; python main.py dindacantik Folder dindacantik tidak ditemukan! PS C:\Users\ASUS\TUBESDAS&gt; </pre> <p>3. User memasukkan nama folder yang valid</p>

	<pre> PS C:\Users\ASUS\TUBESDAS&gt; python main.py database LOADING... Selamat datang di Program OWCA! Masukkan command "HELP" untuk melihat daftar command yang dapat kamu panggil. &gt;&gt;&gt;&gt; </pre>
F15 - Save	<p>1. Folder belum ada</p> <pre> &gt;&gt;&gt;&gt; save Masukkan nama folder: dinda_cantik Saving...  Membuat folder dinda_cantik Berhasil menyimpan data di folder dinda_cantik! &gt;&gt;&gt;&gt; </pre> <p>2. Folder sudah ada</p> <pre> &gt;&gt;&gt;&gt; save Masukkan nama folder: database Saving...  Berhasil menyimpan data di folder database! &gt;&gt;&gt;&gt; </pre>
F16 - Exit	<p>1. Menyimpan folder</p> <pre> &gt;&gt;&gt;&gt; exit Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) y Masukkan nama folder: database Saving...  Berhasil menyimpan data di folder database! PS C:\Users\ASUS\TUBESDAS&gt; </pre> <p>2. Tanpa menyimpan folder</p> <pre> &gt;&gt;&gt;&gt; exit Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) n PS C:\Users\ASUS\TUBESDAS&gt; </pre>
B03 – Monster Ball	<pre> &gt;&gt; Gagal  Swoosshhhh, Anda mengeluarkan Monster Ball !!! Yahhh, Anda belum berhasil mendapatkan monster Freya!!!  &gt;&gt; Berhasil  Swoosshhhh, Anda mengeluarkan Monster Ball !!! Selamat, Anda berhasil mendapatkan Monster Freya Name           : Freya ATK Power      : 70 DEF Power      : 40 HP             : 80 Level          : 4 </pre>



B04 -  
Jackpot

>> Admin

```
Role Admin
anda bukan agent kami
```

>> Agent

```
Role Agent
Owcacoin testing : 300
```

=====

WELCOME TO JACKPOT

=====

- 1. Apple = 5 oc
- 2. Pear = 10 oc
- 3. Cherry = 20 oc
- 4. Clover = 50 oc
- 5. SEVEN = 100 oc

WOULD YOU LIKE TO PLAY (-100 oc) ? (Y/N)

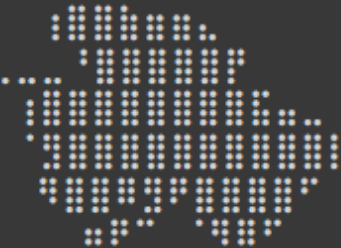

Y

=====

SLOT

=====



	<div>  =====</div> <div><p>JACKPOT</p><p>Anda Mendapatkan 500 OC dan Monster Roga</p><p>Anda mendapatkan : 500 owcacoïn</p><p>Total owcacoïn anda :</p><p>700</p></div>
--	---

# Lampiran

## Lampiran 1. Form Asistensi 1

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

Nomor Asistensi : 1  
No. Kelompok/Kelas : K / K-07  
Tanggal asistensi : 3 Mei 2024

Anggota kelompok	NIM / Nama (Hanya yang Hadir)	
	1	19623137 / Adinda Putri
	2	16523027 / Jerry Alexander Tjoa
	3	19623297 / Dzaky Aurelia Fawwaz
	4	16523257 / Haura Hafizha Harryson
	5	16523187 / Adicahya Sukma Kencana Putra S
	6	16523227 / Muhammad Zarfani
Asisten pembimbing	NIM / Nama	
	13521101 / Arsa Izdiyar Islam	

Catatan Asistensi:

<b>Rangkuman Diskusi</b>
Asisten memberi arahan terhadap fungsi-fungsi yang harus dibuat dalam tugas besar, asistensi membantu menjawab beberapa pertanyaan dari peserta asistensi mengenai tugas besar. Asisten menjelaskan mengenai spesifikasi dari laporan mengenai penggunaan notasi algoritmik dalam laporan.
<b>Tindak Lanjut</b>
Jika ada pertanyaan lebih lanjut mengenai tugas besar maka dapat bertanya secara langsung kepada asisten melalui private chat dan juga melalui forum QnA yang tersedia.

## Dokumentasi 1



## Lampiran 2. Form Asistensi 2

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

Nomor Asistensi : 2  
No. Kelompok/Kelas : K / K-07  
Tanggal asistensi : 11 Mei 2024

Anggota kelompok	NIM / Nama (Hanya yang Hadir)
1	19623137 / Adinda Putri
2	16523027 / Jerry Alexander Tjoa
3	19623297 / Dzaky Aurelia Fawwaz
4	16523257 / Haura Hafizha Harryson
5	16523187 / Adicahya Sukma Kencana Putra S
6	16523227 / Muhammad Zarfan
Asisten pembimbing	NIM / Nama
	13521101 / Arsa Izdiyar Islam

#### Catatan Asistensi:

<b>Rangkuman Diskusi</b>
Asisten menanyakan progress kelompok, untuk progress kelompok masih kurang code di battle & arena. Kelompok menanyakan beberapa pertanyaan mengenai laporan dan telah terjawab melalui asistensi
<b>Tindak Lanjut</b>
Semua pertanyaan terjawab, jika diperlukan asistensi lebih lanjut maka dapat menjadwalkan asistensi tambahan jika diperlukan

## Dokumentasi 2

