

LAPORAN TUGAS BESAR
IF1210 DASAR PEMROGRAMAN
K06-C



Naufal Fakhri Fadhlurrahman	(16523106)
Sacca Kovida Kasmaji	(16523146)
Daniel A. M. Sipayung	(19623076)
Z. Nayaka Athadiansyah	(19623116)
Muhammad Ra'if Alkautsar	(19623296)

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.”

Naufal Fakhri Fadhlurrahman	(16523106)
Sacca Kovida Kasmaji	(16523146)
Daniel A. M. Sipayung	(19623076)
Z. Nayaka Athadiansyah	(19623116)
Muhammad Ra'if Alkautsar	(19623296)

Daftar Isi

Pernyataan Kelompok.....	1
Daftar Isi	2
Daftar Tabel.....	3
Daftar Gambar	4
I. Deskripsi Persoalan.....	5
II. Daftar Pembagian Tugas.....	6
III. Checklist	9
IV. Desain Command.....	10
V. Desain Kamus Data	22
VI. Desain Dekomposisi Algoritmik dan Fungsional Program.....	32
Lampiran.....	57

Daftar Tabel

Tabel 1. Pembagian beban tugas dalam pembuatan program.....	6
Tabel 2. Pembagian beban tugas dalam penulisan laporan.....	7
Tabel 3. Daftar checklist hasil rancangan, implementasi, dan testing setiap primitif	9

Daftar Gambar

I. Deskripsi Persoalan

Tugas besar ini meminta kita untuk membantu Purry si Platypus alias Agen P untuk mengalahkan Dr. Asep Spakbor, seorang ilmuwan jahat yang telah menciptakan berbagai monster mengerikan yang mengancam kedamaian kota Danville. Purry sadar bahwa monster-monster yang diciptakan oleh Dr. Asep Spakbor terlalu kuat dan ia tidak akan mampu melawannya sendirian. Oleh karena itu, Purry meminta bantuan kepada O.W.C.A (Organisasi Warga Cool Abiez) dimana ia bertemu dengan agen-agen hebat lainnya, yaitu para player. Purry menjelaskan kepada para agen, bahwa untuk mengalahkan Dr. Asep Spakbor, para agen dan Purry harus bekerja sama dan merencanakan strategi dengan matang. Purry kemudian meminta para agen untuk mencari dan melatih monster-monster untuk digunakan dalam pertempuran melawan Dr. Asep Spakbor. Dengan tekad yang kuat serta kerja sama tim yang baik, para agen (pemain) yakin bahwa mereka akan mampu menyelesaikan misi ini dan mengalahkan Dr. Asep Spakbor. Para agen (pemain) memutuskan untuk memasuki hutan terpencil yang diyakini sebagai tempat tinggal berbagai jenis monster. Petualangan ini tidaklah mudah, akan ada banyak tantangan, hambatan, serta bahaya yang menanti para agen, tetapi tidak ada pilihan lain, para agen harus terus berjuang demi keselamatan kota Danville. Dari deskripsi persoalan yang ada, dibuatlah sebuah gim yang mensimulasikan para pemain sebagai para agen yang akan membantu Purry mengalahkan Dr. Asep Spakbor dengan berusaha menangkap serta melatih monster-monster yang ada untuk mengalahkan Dr. Asep Spakbor.

II. Daftar Pembagian Tugas

Tabel 1. Pembagian beban tugas dalam pembuatan program

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F00 – Random Number Generator	Fungsi <code>lcg</code> dan <code>rng</code>	19623116	19623116	19623076 19623296
F01 – Register	Fungsi <code>register</code>	19623116	19623116	19623076 19623296
F02 – Login	Fungsi <code>login</code>	19623116	19623116	19623076 19623296
F03 – Logout	Prosedur <code>logout</code>	19623116	19623116	19623076 19623296
F04 – Menu & Help	Prosedur <code>menu</code>	16523146	16523146 19623116 19623296	19623116 19623296
F05 – Monster		19623296	19623296	19623116
F06 – Potion	Fungsi <code>strength_potion</code> , <code>resilience_potion</code> , dan <code>healing_potion</code>	16523106	16523106 19623296	
F07 – Inventory		19623296	19623296	19623116
F08 – Battle		19623296	19623296	19623116
F09 – Arena	Prosedur <code>arena</code>	19623076	19623076 19623296	19623116
F10 – Shop & Currency	Fungsi <code>shop</code>	19623296	19623296 19623116	19623116
F11 – Laboratory	Fungsi <code>laboratory</code> dan <code>upgrade_monster</code>	16523146	19623296	19623076
F12 – Shop Management		19623116	19623296	19623116
F13 – Monster Management		19623076	19623296	19623116
F14 – Load	Fungsi <code>load_data</code>	19623116	19623116	19623076 19623296
F15 – Save	Prosedur <code>save</code>	19623116	19623116	19623076 19623296
F16 – Exit	Prosedur <code>exit</code>	19623116	19623116	19623296
B01 – Git Best Practice	Penggunaan <i>commit message</i> dan <i>branching</i> dalam Git	16523106 16523146 19623076	19623076 19623296	16523106 16523146 19623116

		19623116 19623296		
B02 – Typing	<i>Type checking</i> dan <i>type safing</i> variabel-variabel dan fungsi-fungsi	16523106 16523146 19623076 19623116 19623296	19623116	19623116
B03 – Monster Ball	Fungsi use_monsterball check_monsterball, dan monsterball_success	19623296	19623296	16523106 16523146
B04 – Jackpot	Prosedur jackpot	16523146	16523146 19623296	
B05 – Peta Kota Danville	Fungsi read_map, move_up, move_down, move_left, move_right, check_proximity, serta prosedur print_map	19623296	19623296	19623116
Penggabungan kode dan alur utama permainan	main.py global_var.py	19623116 19623296	19623076 19623116 19623296	16523106 16523146
Fungsi-fungsi pembantu	x01.py	19623116 19623296	19623116 19623296	16523106 16523146 19623076

Tabel 2. Pembagian beban tugas dalam penulisan laporan

No	Bagian	NIM
1	Halaman Sampul	16523146
2	Daftar Isi	19623116
3	Daftar Tabel	19623116
4	Daftar Gambar	19623116
5	Deskripsi Persoalan	16523106
6	Daftar Pembagian Tugas	16523146, 19623116
7	Checklist Hasil Tugas Besar	16523146
8	Desain Command	16523146, 16523106

9	Desain Kamus Data	19623116, 19623296
10	Desain Dekomposisi Algoritmik dan Fungsional Program	16523106
11	Spesifikasi	16523146, 16523106
12	Hasil Pengujian Program	19623116, 19623296
13	Catatan dan Lampiran	16523146

III. Checklist

Tabel 3. Daftar checklist hasil rancangan, implementasi, dan testing setiap primitif

Fitur	Desain	Implementasi	Testing
F00 - Random Number Generator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F01 - Register	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F02 - Login	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F03 - Logout	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F04 - Menu & Help	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F05 - Monster	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F06 - Potion	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F07 - Inventory	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F08 - Battle	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F09 - Arena	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F10 - Shop & Currency	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F11 - Laboratory	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F12 - Shop Management	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F13 - Monster Management	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F14 - Load	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F15 - Save	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F16 - Exit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
B01 - Git Best Practice	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
B02 - Typing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
B03 - Monster Ball	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
B04 - JACKPOT!	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
B05 - Peta Kota Danville	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

IV. Desain Command

1. F00 - Random Number Generator

function lcg (a,c,m,x_0: integer) → integer

{fungsi menghitung rumus LCG}

{I.S. belum dihitung a,c,m,x_0 dengan rumus LCG}

F.S. a,c,m,x_0 dihitung dengan rumus LCG}

function rng (start,stop,x_0: integer) → integer

{fungsi menciptakan bilangan acak}

{I.S. Belum terbentuk bilangan acak}

F.S. Sudah terbentuk bilangan acak}

{DESKRIPSI}

Pembangkit bilangan acak semu (pseudo-random number generator) yang kita gunakan di Tubes cool abiez ini adalah linear congruential generator (LCG) yang dideskripsikan dengan relasi rekurens berikut [1, 2]:

$$x_{(n+1)} = (a * x_n + c) \% m$$

dengan a adalah pengali (multiplier), c adalah penambah (increment), dan m adalah modulus. Suku pertama, x_0, disebut sebagai "seed". Diasumsikan $m > 0$ serta $0 < a, c, x_0 < m$.

Akibat penggunaan modulo, LCG sebenarnya menghasilkan barisan bilangan yang berulang (mempunyai periode) sehingga kualitas algoritma LCG dilihat dari seberapa besar periode tersebut. Keempat "angka aneh" tadi dipilih berdasarkan teorema berikut [1]:

TEOREMA Hull-Dobell. Untuk $c \neq 0$, LCG akan mempunyai periode penuh m jika dan hanya jika

1. c dan m koprima, yang sama saja dengan $\text{fpb}(c, m) == 1$;
2. $a \% p == 1$ jika p adalah faktor prima m;
3. $a \% 4 == 1$ jika 4 adalah faktor m.

Jika kita pilih m sebagai pangkat 2, maka kita cukup pilih c ganjil dan $a = 4k + 1$ untuk sembarang bilangan asli k.

Pemilihan m ini efisien karena komputer bekerja dengan sistem bilangan biner. Agar periodenya besar, kita pilih

pangkat 2 yang besar, misalnya $2^{(11 + 10)}$ karena saya lahir di tanggal 11 Oktober. Selanjutnya, dipilih $c = 1945$

agar terdengar nasionalis serta $a = 69420177013$ yang dipilih secara acak tanpa alasan.

Supaya sederhana, kita pilih nilai default seed-nya 0.

2. F01 – Register

function register (username, pasword: **string**) → **string**

{fungsi untuk registrasi username dan password pengguna baru}

{I.S: database masih kosong belum terisi data pengguna baru

F.S: telah dimasukkan username dan password pengguna baru}

function input_username_reg (username: **string**) → **string**

{fungsi mengecek apakah karakter yang digunakan untuk membuat username sudah valid atau belum}

{I.S: fungsi telah menerima karakter apa saja yang boleh digunakan dalam pembuatan username

F.S: fungsi mengecek apakah karakter yang digunakan dalam username sudah sesuai dengan karakter yang diperbolehkan atau belum dan akan meminta pengguna untuk memasukkan username baru bila username tidak memnuhi ketentuan}

{DESKRIPSI}

Fungsi register() akan mendaftarkan akun baru untuk agent dengan username dan password.

sesuai dengan yang diinput oleh pengguna. Fungsi input_username_reg() digunakan untuk

menerima username dari akun yang akan dibuat dan memvalidasinya.

3. F02 – Login

function login (db: **dictionary**) → **boolean**

{fungsi ini sebagai tempat user untuk login}

{I.S: memasukkan username dan password

F.S: username dan password akan dicek valid atau tidak dan jika benar maka user akan masuk ke program}

function input_username (username: **string**) → **boolean**

{mengecek apakah username telah menggunakan karakter yang valid dan diperbolehkan}

{I.S: username akan dicek apakah sudah menggunakan karakter yang diperbolehkan/tidak dan sudah terdaftar atau belum

F.S: jika username tidak valid akan diberi peringatan untuk mengganti karakter atau mendaftarkan username terlebih dahulu}

function input_password (username: **string**) → **boolean**

{mengecek apakah password sudah sesuai dengan username yang diinput}

{I.S: menerima username dan menyamakan password username tersebut dengan yang tersimpan di database

F.S: jika password valid maka user dapat melanjutkan ke dalam program dan jika salah program akan memberi peringatan bahwa password salah dan meminta untuk input password lagi}

{DESKRIPSI}

Fungsi login() akan menerima username dan password dari pengguna dan mengecek validitasnya. Fungsi akan terus menerima input hingga username atau password sesuai. Selanjutnya, variable logged_in pada global_var.py akan diubah nilainya menjadi True, mengindikasikan pengguna telah berhasil login ke dalam program.

4. F03 – Logout

function logout (logged_in: **boolean**) → **integer**

{fungsi akan meng-logout-kan user dari program}

{I.S: fungsi belum menerima input apapun {I.S: fungsi belum menerima input apapun

F.S: menerima input apakah user sudah melakukan log in atau belum, jika sudah melakukan login maka user akan logout dari program tetapi jika

5. F04 - Menu & Help

procedure help (**input** username : **string**)

{prosedur akan memberi penjelasan bagi jenis perintah dalam game tergantung dari program}

{I.S: fungsi menerima username dan mengecek role user

F.S: penjelasan fungsi jika role admin, user dan belum login }

6. F05 – Monster

```
function attribute_monster (id: string, monster_level: string,  
monster_db) → integer  
{fungsi mengubah stats tiap monster berdasarkan level dengan rumus yang  
telah diberikan}  
{I.S: data csv dibaca dengan semua monster memiliki stats dasar (lvl 1)  
F.S: data monster sesuai dengan level yang diinginkan}  
  
function atk_power (id: string, monster_db) → integer  
{fungsi mengalkulasi atk_power monster berdasarkan dengan fungsi rng}  
{I.S: nilai atk_power yang sudah dikalkulasikan di fungsi  
attribute_monster dimasukkan  
F.S: dihasilkan nilai atk yang telah diacak oleh fungsi rng}  
  
function atk_result (attacker_id: string, defender_id: string,  
monster_db) → integer  
{menghitung nilai atk yang diterima oleh musuh, setelah dikurangi oleh  
nilai def-power musuh}  
{I.S: menerima nilai atk_power monster yang mneyerang  
F.S: mengurnagi atk_power berdasarkan def_power musuh dnegan rumus  
kalkulasi yang ada}
```

7. F06 – Potion

```
function strength_potion (atk_power: integer) → integer  
{fungsi menambahkan atk_power monster}  
{I.S: memilih strength potion  
F.S: nilai atk_power akan ditambahkan sebanyak 5% dari nilai atk_power  
sekarang}  
  
function resilience_potion (def_power: integer) → integer  
{fungsi menambahkan def_power monster}  
{I.S: memilih resilience potion  
F.S: nilai nilai def_power akan ditambahkan sebanyak 5% dari nilai  
def_power sekarang}  
  
function healing_potion (current_hp: integer, max_hp: integer) → integer  
{fungsi menambahkan hp monster}  
{I.S: memilih healing potion  
F.S: nilai hp akan ditambahakn nilai hp sekarang yang dikalikan 25%}
```

8. F07 – Inventory

```
function get_monster (i,lv : integer) → list
{fungsi mengambil data monster yaitu nama monster, atk power, def power
 dan HP monster}
{I.S: list monster kosong dan memilih strength potion
 F.S: list monster terisi data nama monster, atk power, def power dan HP
 monster}

procedure show_items ()
{prosedur menunjukan item yang dimiliki pengguna}
{I.S: tampilan layar kosong
 F.S: ditampilkan item - item yang dimiliki pengguna}

procedure show_monsters ()
{prosedur menunjukan monster yang dimiliki pengguna}
{I.S: tampilan layar kosong
 F.S: ditampilkan monster - monster yang dimiliki pengguna}

function get_start_index(db, current_user : integer) -> integer
{Fungsi dibuat untuk mendapatkan indeks pertama dari barang pengguna,
 digunakan pada monster inventory dan item inventory}
{I.S: list monster kosong dan memilih strength potion
 F.S: list monster terisi data nama monster, atk power, def power dan HP
 monster}

procedure show_inventory ()
{prosedur menunjukan monster yang dimiliki pengguna}
{I.S: tampilan layar kosong
 F.S: ditampilkan monster - monster dan item - item yang dimiliki
 pengguna}
```

9. F08 – Battle

```
function use_potion ()
{fungsi untuk memilih potion yang ingin digunakan dalam battle}
{I.S:menerima data potion yang dimiliki oleh tiap agent
 F.S:menerima input potion yang dipilih oleh pemain dan memberikan efek
 potion yang dipilih pada monster pemain}

function battle ()
{menampilkan statistik monster dalam pertaraungan}
{I.S: menerima input statistik monster player dan monster yang ingin
 dilawan
 F.S: menampilkan hasil pertarungan berupa statistik monster musuh atau
 mosnter player setelah melakukan "atk" dan menentukan menang/kalah}
```

10. F09 – Arena

```
function arena ()  
{fungsi menambahkan kesempatan untuk pengguna bertarung dengan monster dan  
mendapat oc }  
{I.S: menerima input pengguna memainkan arena  
F.S: menampilkan hasil permainan di arena berupa oc yang tambahkan ke oc  
pengguna sesuai jumlah menang}
```

11. F10 - Shop & Currency

```
function add_monster(user_id, index, enemy_level : integer  
,monster_inv_db : list )→ list  
{fungsi menambahkan monster baru kedalam database}  
{I.S: tidak ada monster baru  
F.S: terdapat monster baru}  
  
function add_item(user_id : integer, item : string , item_inv_db :  
list)→ list  
{fungsi menambahkan monster baru kedalam database}  
{I.S: tidak ada item baru  
F.S: terdapat item baru}  
  
function remove_item(index : integer, item_inv_db : list)→ list  
{fungsi menghilangkan item dari database}  
{I.S: item ada dalam database  
F.S: item hilang dari database}  
  
procedure monster_shop_list()  
{prosedur menunjukkan semua monster yang tersedia di shop}  
{I.S: tampilan layar kosong  
F.S: ditampilkan monster - monster yang tersedia di shop }  
  
procedure item_shop_list()  
{prosedur menunjukkan semua item yang tersedia di shop}  
{I.S: tampilan layar kosong  
F.S: ditampilkan item - item yang tersedia di shop }
```



```

function monster_shop(monster_inv_db, monster_shop_db, monster_db, oc :
list user_id : integer) → list
{fungsi memungkinkan pengguna membeli monster yang tersedia di shop}
{I.S: pengguna tidak dapat membeli monster baru
F.S: pengguna tidak dapat membeli monster baru }

function item_shop(item_inv_db, item_shop_db, oc : list user_id :
integer) → list
{fungsi memungkinkan pengguna membeli monster yang tersedia di shop}
{I.S: pengguna tidak dapat membeli item baru
F.S: pengguna tidak dapat membeli item baru }

function shop(monster_inv_db, item_inv_db, monster_shop_db, item_shop_db,
monster_db, oc : list user_id : integer) → list
{fungsi shop utama memungkinkan pengguna memilih mau membeli item atau
membeli monster}
{I.S: pengguna tidak dapat membeli item dan monster baru
F.S: pengguna dapat memilih membeli item baru membeli monster baru}

```

12. F11 – Laboratory

```

function monster_upgrade(monster_inv_db, monster_db, oc : list i :
integer) → list
{fungsi shop utama memungkinkan pengguna memilih mau membeli item atau
membeli monster}
{I.S: pengguna tidak dapat membeli item dan monster baru
F.S: pengguna dapat memilih membeli item baru membeli monster baru}

procedure labortary()
{prosedur menampilkan semua monster yang dimiliki pengguna dan
memungkinkan pengguna menaikkan level monster dengan membayar sejumlah
oc}
{I.S: tampilan layar kosong
F.S: ditampilkan monster yang dimiliki pengguna dan dinaikkan level
monster bila pengguna ingin menaikannya}

```

13. F12 - Shop Management

```

procedure input_validation() procedure input_validation()
{prosedur memeriksa masukan pengguna}
{I.S: menerima masukan pengguna
F.S: jika benar program lanjut berjalan, bila salah maka akan
ditampilkan error message }

function add_monster_shop(monster_shop_db : list monster_id, stock,
price, : integer) → list
{fungsi menambahkan monster dalam shop}
{I.S: pengguna memanggil fungsi add_monster_shop
F.S: pengguna menambahkan monster baru dalam shop jika berhasil }

```

```

function remove_monster_shop(monster_shop_db : list monster_id :
integer) → list
{fungsi menghapus monster dalam shop}
{I.S: pengguna memanggil fungsi remove_monster_shop
 F.S: pengguna menghapus monster dalam shop jika berhasil }

function change_monster_shop(monster_shop_db : list monster_id, stock,
price, : integer) → list
{fungsi menambahkan stock dan harga monster dalam shop}
{I.S: pengguna memanggil fungsi change_monster_shop
 F.S: pengguna menambahkan stock dan harga monster dalam shop
 jika berhasil }

function add_item_shop(item_shop_db : list item_type, stock, price, :
integer) → list

{fungsi menambahkan item dalam shop}
{I.S: pengguna memanggil fungsi add_item_shop
 F.S: pengguna menambahkan item baru dalam shop jika berhasil }

function remove_item_shop(item_shop_db : list item_type :
integer) → list
{fungsi menghapus item dalam shop}
{I.S: pengguna memanggil fungsi remove_item_shop
 F.S: pengguna menghapus item dalam shop jika berhasil }

function change_item_shop(monster_shop_db : list item_type, stock, price,
: integer) → list
{fungsi menambahkan stock dan harga item dalam shop}
{I.S: pengguna memanggil fungsi change_item_shop
 F.S: pengguna menambahkan stock dan harga item dalam shop
 jika berhasil }

function item_shop_management(item_shop_db : list) → list
{fungsi memberi pilihan pengguna untuk menambah , ubah atau hapus item
 dalam shop}
{I.S: pengguna memanggil fungsi item_shop_management
 F.S: pengguna menambahkan, mengubah atau menghapus item dalam
 menambahkan stock dan harga item dalam shop jika berhasil }

function monster_shop_management(monster_db ,monster_shop_db : list) →
list
{fungsi memberi pilihan pengguna untuk menambah , ubah atau hapus
 monster dalam shop}
{I.S: pengguna memanggil fungsi monster_shop_management
 F.S: pengguna menambahkan, mengubah atau menghapus item dalam
 menambahkan stock dan harga monster dalam shop jika berhasil }

```

```

procedure shop_management()
{prosedur memberi pilihan pada pengguna untuk mengubah item atau monster
 dalam shop}
{prosedur memberi pilihan pada pengguna untuk mengubah  }
{prosedur memeriksa masukan pengguna}
{I.S: prosedur shop_management di panggil pengguna
 F.S: program akan mengarah ke monster_shop_managmenet
 ,item_shop_managmenet atau exit sesuai pilihan pengguna}

```

14. F13 - Monster Management

```

function show_monster_database(path: string) → list
{fungsi melihat semua monster dan menambahkan monster pada file csv}
{I.S: tidak ada monster baru dalam csv
 F.S: ditunjukkan semua monster dalam csv atau ditambahkan monster baru
 dalam csv}
function add_monster (monster_db: string) → list
{I.S: dalam kondisi kosong
 {I.S: dalam kondisi kosong
 F.S: sudah menerima input type dan statistik monster }
 F.S: }
 F.S: sudah menerima input type dan statdjsalkdjalkdjalkdjalkdistik
 monster}

function remove_monster (monster_db: string) → list
{I.S: menerima input monster yang ingin dihapus
 F.S: monster terhapus}

function change_monster (monster_db: string) → list
{I.S: menerima input monster yang ingin dubah statistiknya
 F.S: statistik monster berhasil diubah}
function monster_management(monster_db: string) → list
{I.S: menyediakan pilihan untuk menghilangkan, menambah, atau
 mengubah statistik monster
 F.S: memberikan output sesuai pilihan pemain }

```

15. F14 – Load

```
function read_header (path: string) → string
{fungsi menerima masukan argumen pada file csv}
{I.S: menerima path dan membaca file csv
 F.S: mengekstrak atribut pada tiap baris pertama sebagai array}

function load_csv (path: string) → dictionary
{fungsi memuat isi file csv}
{I.S: data belum dimuat lalu menerima path file csv yang dibutuhkan
 F.S: data dari tiap kolom dipasangkan terhadap atribut yang sesuai }
```

function load_data (csv_dir: string) → list[dictionary]
{sebagai jalan pintas untuk memanggil file csv}
{I.S: menerima file csv
F.S: mengubah isi file csv menjadi list yang berisikan dictionary}
{DESKRIPSI}
Skema load untuk memuat data dari csv ke dalam dictionary of lists sebagai database temporer dalam game terdiri atas beberapa langkah. Pertama-tama, header yang memuat parameter-parameter (kolom) dalam database akan dimuat. Selanjutnya, barulah baris-baris yang tersisa dimuat dan dipasangkan sesuai dengan parameter/kolomnya.

16. F15 – Save

```
function save (db: dictionary, path: string)
{fungsi menyimpan data ke dalam file yang terstruktur dan eksternal}
{I.S: menerima input nama folder yang digunakan untuk menyimpan data
 F.S: menyimpan data ke dalam folder yang ada atau membuat folder baru
 jika folder nama folder tidak ditemukan}
```

17. F16 – Exit

```
function exit (dir: string)
{fungsi akan menyimpan progres para pemain dalam game ini}
{I.S: memberikan output berupa pertanyaan ingin menyimpan data/tidak
 kepada pemain output berupa pertanyaan ingin menyimpan data/tidak kepada
 pemain
 F.S: jika pemain memilih menyimpan data maka semua data akan disimpan ke
 dalam file csv yang ada, namun jika memilih tidak maka data tidak akan
 disimpan }
```

{DESKRIPSI}
Ketika pengguna ingin keluar dari program, pengguna diberi pilihan untuk menyimpan progresnya atau tidak. Jika pengguna berkenan, seluruh database dalam program yang tersimpan dalam bentuk dictionary akan di-parsing ulang menjadi baris-baris yang dipisahkan dengan semikolon (;) lalu file .csv tujuan akan ditimpa baris per baris dengan data yang baru. Jika input yang diterima (terkait kesediaan pengguna untuk menyimpan data) tidak valid, yakni selain Y/N (case insensitive), pengguna terus diminta untuk memberikan input hingga valid.

18. B03 - Monster Ball

function monsterball_succes (level: **integer**) → **boolean**
{fungsi digunakan untuk menangkap monster saat BATTLE}
{I.S: tiap level monster memiliki nilai value yang berebeda
F.S: keberhasilan monsterball untuk menangkap monster ditentukan oleh persentase kemungkinan berdasarkan level monster yang ditangkap}

function check_monsterball(item_inv_db, user_id)
{mencari item monsterball di inventory }
{I.S: membuka kolom "type""monsterball dalam inventory
F.S: mencari item bernama "monsterball" di dalam inventory }

function use_monsterball (item_inv_db, user_id)
{untuk mengambil monsterball yang dimiliki}
{I.S: menerima jumlah monsterball yang dimiliki berdasarkan inventory player
F.S: mengurangi jumlah monsterball di inventory saat monsterball digunakan}

19. B04 - JACKPOT!

function jackpot (username: **string**) → **none**
{sebagai permainan sampingan agar tidak bosan dan ~~pemuas hasrat gacha~~}
{I.S: terdapat lima buah hadiah yang bisa didapatkan oleh pemain
F.S: memberikan hadiah kepada pemain sesuai dengan total value item yang didapat}

20. B05 - Peta Kota Danville

```
function read_map(N, M, posx, posy: integer, fn : string) → array
  read_map(N, M, posx, posy: integer, fn : string) → array
{fungsi berfungsi sebagai pembuat array peta awal dan posisi awal
  pengguna }
{I.S: array kosong
  F.S: array terbuat degan penghalang dan posisi awal pengguna
  tercipta}

procedure print_map()
{prosedur menampilkan peta pada layar pengguna}
{I.S: tampilan layar kosong
  F.S: ditampilkan peta pada layar pengguna}

function moveUp(worldmap : array posx, posy: string) → array
{fungsi berfungsi memindahkan pengguna ke posisi atas}
{I.S: pengguna dalam array dalam keadaan diam
  F.S: pengguna dalam array bergerak ke atas}

function moveDown(worldmap : array posx, posy: string) → array
{fungsi berfungsi memindahkan pengguna ke posisi bawah}
{I.S: pengguna dalam array dalam keadaan diam
  F.S: pengguna dalam array bergerak ke bawah}

function moveRight(worldmap : array posx, posy: string) → array
{fungsi berfungsi memindahkan pengguna ke posisi kanan}
{I.S: pengguna dalam array dalam keadaan diam
  F.S: pengguna dalam array bergerak ke kanan}

function moveLeft(worldmap : array posx, posy: string) → array
{fungsi berfungsi memindahkan pengguna ke posisi kiri}
{I.S: pengguna dalam array dalam keadaan diam
  F.S: pengguna dalam array bergerak ke kiri}

function checkProximity(worldmap : array action, posx, posy: string) →
  array
{fungsi berfungsi mengecek apakah pergerakan valid}
{I.S: pengguna dalam array dalam keadaan ingin bergerak
  F.S: pengguna dalam array bergerak ke arah yang dipilih setelah
  divalidasi}
```

V. Desain Kamus Data

a. main.py (mencakup global_var.py)

KAMUS

```
type users <id: integer,
            username: string,
            password: string,
            role: string,
            oc: integer>

type monsters <id: integer,
               type: string,
               atk_power: string,
               def_power: string,
               hp: integer>

type item_inventory <user_id: integer,
                    type: string,
                    quantity: integer>

type monster_inventory <user_id: integer,
                       monster_id: integer,
                       level: integer>

type item_shop <type: string,
               stock: integer,
               price: integer>

type monster_shop <monster_id: integer,
                  stock: integer,
                  price: integer>

parser      : ArgumentParser    {type bawaan argparse}
args        : Namespace         {idem}
csv_dir     : string
logged_in   : Boolean
database    : array[0..5] of dictionary of string to array[0..NEff]
              of string/integer
user_db     : users
monster_db  : monsters
item_inv_db : item_inventory
monster_inv_db : monster_inventory
item_shop_db : item_shop
monster_shop_db : monster_shop

procedure add_argument(input parser: ArgumentParser, input
folder_name: str, input nargs: str, input help: str)
{Fungsi bawaan dari argparse untuk menambah argumen}

procedure title_screen()
{Prosedur untuk mencetak judul game}

procedure main_gameplay(user_db, monster_db, monster_shop_db,
monster_inv_db, item_shop_db, item_inv_db : dictionary of string to
array[0..NEff] of string/integer)
{Prosedur yang menjalankan bagian utama permainan}
```

b. F00 – Random Number Generator

KAMUS

function lcg(input a: integer, input c: integer, input m: integer,
input x_0: integer, output rand_num: integer)

function rng(input start: integer, input stop: integer, input x_0:
integer, output rand_num: integer)

KAMUS LOKAL

a, c, m, x_n : integer

c. F01 – Register

KAMUS LOKAL

parent_path : string
data_path : string
user_db : users
isExist : Boolean
username : string
password : string

d. F02 – Login

KAMUS LOKAL

parent_path : string
data_path : string

I. Fungsi login()

KAMUS LOKAL

user_db : users
username : string

II. Fungsi input_username()

KAMUS LOKAL

user_db : users
valid_char : string
valid1 : Boolean
valid2 : Boolean
username : string

III. Prosedur input_password()

KAMUS LOKAL

username : string
user_db : users
user_idx : integer
password : string
valid : Boolean

e. F03 – Logout

KAMUS LOKAL

logged_in : Boolean

f. F04 – Menu & Help

KAMUS LOKAL

username : string
login_status : Boolean
user_db : users

g. F05 – Monster

KAMUS LOKAL

parent_path : string
data_path : string

i. Fungsi attribute_monster

KAMUS LOKAL

id : integer
monster_level : integer
monster_db : monsters
monster_type : string
monster_atk : integer
monster_def : integer
monster_hp : integer

ii. Fungsi atk_power

KAMUS LOKAL

atk_value : integer
atk : integer

iii. Fungsi atk_result

KAMUS LOKAL

attacker_atk_value : integer
defender_def_value : integer
damage : integer

h. F06 – Potion

i. Fungsi strength_potion

KAMUS LOKAL

atk_power : integer
increase : integer

ii. Fungsi resilience_potion

KAMUS LOKAL

def_power : integer
increase : integer

iii. Fungsi healing_potion

KAMUS LOKAL

current_hp : integer
max_hp : integer
new_hp : integer
heal_amount : integer

i. F07 – Inventory

i. Fungsi get_monster

KAMUS LOKAL

i, lv, user_id : integer

ii. Fungsi show_items

KAMUS LOKAL

item_inv_db : item inventory
user_id, i, j : integer
found : Boolean

iii. Fungsi show_monsters

KAMUS LOKAL

monster_inv_db : monster inventory
monster_db : monsters
user_id, i, j : integer
found : Boolean

iv. Fungsi get_start_index

KAMUS LOKAL

db : users or monsters or item inventory or
monster inventory or item shop or monster shop
user_id : integer
i : integer

v. Fungsi show_inventory

KAMUS LOKAL

item_inv_db : item inventory
monster_inv_db : monster inventory
monster_db : monsters
user_id : integer

j. F08 – Battle

i. Fungsi show_state

KAMUS LOKAL

type1, type2 : string
hp1, hp2, atk1, atk2, def1, def2 : integer

ii. Fungsi use_potion

KAMUS LOKAL

item_inv_db : item inventory
user_id, your_monster, your_atk, your_def, your_hp : integer
pot_choice : string

iii. Fungsi battle

KAMUS LOKAL

monster_db : monsters
monster_inv_db : monster inventory
item_inv_db : item inventory

```

battle_type      : string
user_id, enemy_level, oc, random_index, enemy_monster, enemy_type,
enemy_atk, enemy_def, enemy_hp, enemy_damage, your_index,
your_monster_idx, your_monster, your_type, your_atk, your_def,
your_hp, damage_dealt, damage_received: integer
isBattle: Boolean

```

k. F09 – Arena

KAMUS LOKAL

```

parent_path      : string
data_path        : string
monster_db       : monsters
monster_inv_db   : monster inventory
user_id, oc, total_dealt, total_received, reward, stage: integer
defeat           : Boolean

```

l. F10 – Shop & Currency

i. Fungsi add_monster

KAMUS LOKAL

```

user_id          : integer
monster_index    : integer
enemy_level      : integer
monster_inv_db   : monster inventory
i                : integer
temp_user        : string
temp_index       : integer
temp_level       : integer

```

ii. Fungsi add_item

KAMUS LOKAL

```

user_id          : integer
item             : string
item_inv_db      : item inventory
temp_user        : string
temp_type        : string
temp_quantity    : integer
found            : Boolean

```

iii. Fungsi remove_item

KAMUS LOKAL

```

index            : integer
item_inv_db      : item inventory

```

iv. Fungsi monster_shop_list

KAMUS LOKAL

```

monster_shop_db : monster shop
monster_db       : monsters
monster_idx      : integer

```

v. Fungsi item_shop_list

KAMUS LOKAL

```

item_shop_db : item shop

```

vi. Fungsi monster_shop

KAMUS LOKAL

monster_inv_db : monster inventory
monster_shop_db: monster shop
monster_db : monsters
oc : integer
user_id : integer

vii. Fungsi item_shop

KAMUS LOKAL

item_inv_db : item inventory
item_shop_db : item shop
oc : integer
user_id : integer

viii. Fungsi shop

KAMUS LOKAL

monster_inv_db : monster inventory
monster_shop_db: monster shop
monster_db : monsters
item_inv_db : item inventory
item_shop_db : item shop
oc : integer
user_id : integer

m. F11 – Laboratory

i. Fungsi upgrade_monster

KAMUS LOKAL

monster_inv_db : monster inventory
monster_db : monsters
i, oc, monster_index_in_monster_db, monster_name, upgrade_cost,
current_level : integer

ii. Fungsi laboratory

KAMUS LOKAL

monster_inv_db : monster inventory
monster_db : monsters
user_id : integer

n. F12 – Shop Management

i. Fungsi add_monster_shop

KAMUS LOKAL

monster_shop_db : monster shop
monster_id, stock, price : integer

ii. Fungsi remove_monster_shop

KAMUS LOKAL

monster_shop_db: monster shop
monster_id, idx: integer

iii. Fungsi change_monster_shop

KAMUS LOKAL

monster_shop_db : monster_shop
monster_id, idx, stock, price : integer

iv. Fungsi add_item_shop

KAMUS LOKAL

item_shop_db : item_shop
item_type : string
stock, price : integer

v. Fungsi change_item_shop

KAMUS LOKAL

item_shop_db : item_shop
item_type : string
price : integer

vi. Fungsi item_shop_management

KAMUS LOKAL

item_shop_db : item_shop
choice, item_type : string
user_id, stock, price: integer

vii. Fungsi monster_shop_management

KAMUS LOKAL

monster_shop_db : monster_shop
choice : string
monster_id, stock, price: integer

viii. Fungsi shop_management

KAMUS LOKAL

monster_shop_db: monster_shop
item_shop_db : item_shop
choice : string

o. F13 – Monster Management

KAMUS LOKAL

monster_db : monsters
choice, monster_type : string
monster_idx, monster_atk, monster_def, monster_hp: integer

p. F14 – Load

i. Fungsi read_header

KAMUS LOKAL

path : string

ii. Fungsi load_csv

KAMUS LOKAL

```

path      : string
headers   : array[0..NEff] of string
entries   : string
data      : dictionary

```

iii. Fungsi load_data

KAMUS LOKAL

```

csv_dir      : string
user_db      : users
monster_db   : monsters
monster_shop_db : monster shop
monster_inv_db : monster inventory
item_shop_db  : item shop
item_inv_db   : item inventory

```

q. F15 – Save

KAMUS LOKAL

```

parent_path : string
data_path   : string
db          : users or monsters or item inventory or
              monster inventory or item shop or monster shop
logged_in   : Boolean

```

r. F16 – Exit

KAMUS LOKAL

```

dir : string

```

s. B03 – Monster Ball

i. Fungsi monsterball_success

KAMUS LOKAL

```

level, result : integer

```

ii. Fungsi check_monsterball

KAMUS LOKAL

```

item_inv_db : item inventory
i, user_id  : integer

```

iii. Fungsi use_monsterball

KAMUS LOKAL

```

item_inv_db : item inventory
i, user_id  : integer

```

t. B04 – JACKPOT!

KAMUS LOKAL

```

parent_path, data_path, monster : string
monster_db                      : monsters
monster_inv_db                  : monsters inventory
items                          : arr[0..4] of string
oc, user_id, item_values, item1_idx, item2_idx, item3_idx,
item1, item2, item3, random_monster_index, monster_level: integer

```

u. B05 – Peta Kota Danville

KAMUS LOKAL

```
worldmap : arr[0..9] of arr[0..9] of string
N, M, fn, posx, posy, i, j : integer
arr : arr[0..N+1] of arr[0..N+1] of string
symbol : string
```

v. X01 – Fungsi-fungsi Pembantu

```
function maxi(input/output a: integer, input/output b: integer)
{Fungsi untuk mencari yang terbesar di antara dua bilangan bulat}
```

```
function mini(input/output a: integer, input/output b: integer)
{Fungsi untuk mencari yang terkecil di antara dua bilangan bulat}
```

```
function last(input arr: array[0..NEff] of string/integer, output
last_elmt : string/integer)
{Fungsi untuk mencari elemen terakhir dari sebuah array}
```

```
function is_in(input e: integer/string, input arr: array[0..NEff],  
output: Boolean)  
{Fungsi untuk memeriksa apakah suatu elemen (string/integer) terdapat  
dalam suatu array}
```

```
function get_idx(input e: integer/string, input arr: array[0..NEff],  
output idx: integer)  
{Fungsi untuk mengambil indeks dari suatu elemen dalam array.  
Mengembalikan -1 jika tidak ada.}
```

```
function lower(input/output text: string)
    KAMUS LOKAL
    text_len : integer
{Fungsi untuk mengubah seluruh huruf dalam teks menjadi huruf kecil}
```

```
function subset(input arr1: array[0..Neff] of string/integer, input
arr2: array[0..Neff] of string/integer, output result: boolean){}
{Fungsi untuk memeriksa apakah arr1 adalah subset dari arr2}
```

```
procedure remove_nth_line(input n: integer)
{Menghapus baris yang terletak n baris di atas kursor terminal}
```

```
function is_space(input char: string)
{Memeriksa apakah suatu karakter adalah spasi (' ')}

```

```
function is_numerical(input text: string, output result: Boolean)
{Memeriksa apakah suatu teks sepenuhnya berisi angka}
```

```
function split_str(input line: string, output row: array[0..NEff] of
string)
{Memecah string menjadi list berdasarkan pemisah semikolon (;)}
```

```
function strip_str(input_line: string, output_stripped_line: string)
{Menghilangkan karakter \n dan \t}
```

```
function remove_ele(input index: integer, input arr: arr[0..Neff] of integer, output new_arr: arr[0..Neff] of integer)
{Menghapus elemen pada indeks tertentu dalam array}
```

```
function procedure print text(input file path: string)
```

```
{Mencetak seluruh konten dari file .txt pada file_path}  
  
function validate_dir(input dir: string, output result: Boolean)  
{Memvalidasi keberadaan direktori (folder)}
```


VI. Desain Dekomposisi Algoritmik dan Fungsional Program

Dekomposisi Algoritmik

{I.S: Disiapkan file csv sebagai database utama, (monster, user, item, shop, dll)}

Dibuat fungsi parser

{F.S/I.S: fungsi memuat file csv ke dalam source code}

Dibuat fungsi penunjang dari F00-F16

{F.S/I.S: melalui file csv yang sudah dimuat ke dalam source code, dibuat fungsi yang mengatur jalannya permainan}

Dibuat fungsi bonus dari B03-B05

{F.S/I.S: melalui file csv yang sudah dimuat ke dalam source code, dibuat fungsi tambahan yang berperan sebagai fitur tambahan untuk pemain di luar permainan utama}

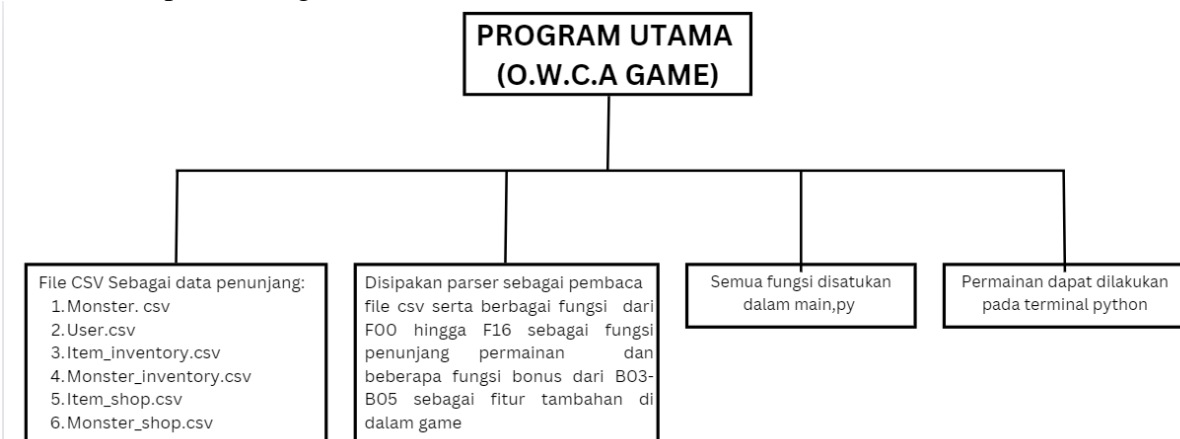
Disatukan dalam main.py

{F.S/I.S: fungsi di F00-16 dan B03-B05 disatukan ke dalam program utama dan dihubungkan satu sama lain sehingga dapat bekerja sebagai sebuah game yang utuh}

Program dapat dijalankan

{F.S: program dapat dijalankan di terminal}

a. Dekomposisi Fungsional



b. Spesifikasi Fungsi/Prosedur dalam Program

1. F00 - Random Number Generator

ALGORITMA

```
function lcg() {Mengitung dengan rumus LCG}
    → (a * x_0 + c) % m
    {Mengembalikan (a * x_0 + c) % m seperti pada rumus LCG}
```

```
function rng () {Menghasilkan bilangan acak}
    {KAMUS LOKAL}
    {a,c,m,x_n,rand_num = integer}
    a ← 69420177013
    c ← 1945
    m ← 2 ** 21
    x_n ← x_0
    x_n ← lcg(a, c, m, x_n) {Menciptakan bilangan acak}
    rand_num ← x_n % (stop - start) + start
    {Memastikan bilangan acak ada di selang [start, stop]}
    → rand_num
```

2. F01 – Register

ALGORITMA

```
import os, sys
from global_var import *
function register()
    print text (register_screen)
    output ("<=====>")
    output (" O. W. C. A. ")
    output (" We serve. ")
    output ("<=====>")

    isExist ← False
    username ← input username reg()
    i traversal n [user_db]["username"]:
        if i == username then
            isExist ← True
            output(f"Username sudah ada, login dengan username
            '{username}' atau gunakan username lain !")
    if isExist == False then
        password ← input("Password: ")
        {Menambahkan data pengguna ke dalam database}
        user_db["id"].append(last(user_db["id"]) + 1)
        user_db["username"].append(username)
        user_db["password"].append(password)
        user_db["role"].append('agent')
        user_db["oc"].append(0)
    → user_db
```

```

function input_username_reg ()
  valid_char ←
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-_"
  username ← input("Username: ")

  valid ← subset(username, valid_char)
  while not valid do
    remove_nth_line(1)
    output("Username hanya boleh berisi alfabet, angka, tanda
  hubung bawah (_), dan tanda strip (-)!")
    username ← input("Username: ")
    remove_nth_line(1)
    valid ← subset(username, valid_char)
    if valid then
      remove_nth_line(1)
  → username

```

3. F02 – Login

ALGORITMA

```

import sys, os

from global_var import *
function login()
  output("<=====>")
  output(" O. W. C. A. ")
  output(" We serve. ")
  output("<=====>")
  {Bagian utama fungsi}
  username ← input_username(db)
  input_password(username, db)

function input_username ()
  valid_char ←
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-_"
  username ← input("Username: ")
  remove_nth_line(1)
  valid1 ← subset(username, valid_char)
  valid2 ← is_in(username, db["username"])
  while not valid1 or not valid2:
    if not valid1 then
      output("Username hanya boleh berisi alfabet, angka, tanda
  hubung bawah (_), dan tanda strip (-)!")
    else {not valid2}
      output("Username tidak terdaftar. ")
      username ← input("Username: ")
      remove_nth_line(1)
      valid1 ← subset(username, valid_char)
      valid2 ← is_in(username, db["username"])
      remove_nth_line(1)
  → username

function input_password()
  user_idx ← get_idx(username, db["username"])

  password ← input("Password: ")
  remove_neth_line (1)

```

```

valid ← db["password"][user_idx] == password
if valid then
    → password
else
    while not valid do
        output ("Password salah!")
        password ← input ("Password: ")
        remove_nth_line(1)
        valid ← db["password"][user_idx] == password
        remove_nth_line(1)

```

4. F03 – Logout

```

ALGORITMA
from global_var import *

procedure logout()
    if logged_in then
        output ("Berhasil logout!")
        logged_in ← False
    else {not logged_in}
        output ("Logout gagal karena Anda belum login!")

```

5. F04 - Menu & Help

```

ALGORITMA
from global_var import *
current_user ← get_idx("Asep_Spakbor",user_db["username"])

function help()
    if user_db["role"][current_user] == "admin" then
        output ("""
===== HELP =====

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

1. Logout           : Keluar dari akun yang sedang digunakan.
2. Shop             : Melakukan manajemen pada SHOP sebagai tempat
jual beli peralatan Agent.
3. Monster Management: Menambahkan monster baru ke dalam database.

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

```

```

else if user_db["role"][current_user] == "agent" then
  output ((f"""
=====HELP=====

Halo Agent {user_db["username"][current_user]}.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. Shop        : Membeli agent, monster dan potion.
3. Inventory   : Menampilkan monster dan potion serta spesifikasi yang
dimiliki .
4. Battle      : Melakukan pertarungan melawan monster.
5. Arena       : Melatih monster dan mendapat koin oc saat menang .
6. Laboratory: Mengupgrade monster yang ada di inventory dengan koin
oc.

Footnote:
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang
terdaftar.
2. Jangan lupa untuk memasukkan input yang valid.
""")
  else {belum login}
  output("""
=====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.
""")

help(current_user)

```

6. F05 – Monster

```

ALGORITMA
from src.F00_RandomNumberGenerator import *
from src.x01 import *
import sys, os
from global_var import *

function attribute_monster()
  monster ← monster_db['id'].index(id)
  monster_type ← monster_db['type'][monster]
  monster_atk ← int(monster_db['atk_power'][monster] + (monster_level -
1) * 0.1 * monster_db['atk_power'][monster])
  monster_def ← int(monster_db['def_power'][monster] + (monster_level -
1) * 0.1 * monster_db['def_power'][monster])
  monster_hp ← int(monster_db['hp'][monster] + (monster_level - 1) * 0.1
* monster_db['hp'][monster])
  → (monster type, monster atk, monster def, monster hp)

```

```

function atk_power()
    atk ← int(atk_value + ((rng(-30, 30, 2) / 100) * atk_value))
    → atk

function atk_result()
    attacker_atk ← atk_power(attacker_atk_value)
    damage ← attacker_atk * (1 - int(defender_def_value) / 100)
    output(f"ATK Value: {attacker_atk_value}. DEF Value:
{defender_def_value}.")

    if damage < 0 then
        damage ← 0

    → damage

```

7. F06 – Potion

ALGORITMA

```

from src.F05_Monster import *

```

```

import sys, os

```

```

from global_var import *

```

```

function strength_potion()

```

```

{Menghitung penambahan atk_power dengan strength potion}

```

```

    increase ← (atk_power * 0.05)

```

```

    {menyimpan nilai tambahan atk_power yang telah diberi potion}

```

```

    → atk_power + increase

```

```

    {mengembalikan nilai atk_power}

```

```

function resilience_potion ()

```

```

{Menghitung penambahan def_power dengan resilience potion }

```

```

    increase ← (def_power * 0.05)

```

```

    → def_power + increase

```

```

    {mengembalikan nilai def_power}

```

```

function healing_potion ()

```

```

{Menghitung penambahan hp dengan healing potion}

```

```

    heal_amount ← (max_hp * 0.25)

```

```

    {jumlah penambahan hp yang didapat}

```

```

    new_hp ← current_hp + heal_amount

```

```

    {nilai hp yang sudah ditambah}

```

```

    → min(new_hp, max_hp)

```

```

    {memastikan nilai hp tidak melebihi hp maksimum}

```

8. F07 – Inventory

ALGORITMA

```
from src.x01 import *
from src.F14_Load import *
from src.F05_Monster import *

from global_var import *

function get_monster ()
    monster_to_get  $\leftarrow$  attribute_monster(i, lv, monster_db)
     $\rightarrow$  (f"{monster_to_get[0]} (ATK Power: {monster_to_get[1]} | DEF Power:
{monster_to_get[2]} | HP: {monster_to_get[3]})")

function show_items ()
    i  $\leftarrow$  0
    j  $\leftarrow$  1
    while item_inv_db['user_id'][i]  $\neq$  current_user do
        i  $\leftarrow$  i + 1
        pass
    while item_inv_db['user_id'][i] == current_user do
        output (f"{j}. {(item_inv_db['type'][i]).title()} (Quantity:
{item_inv_db['quantity'][i]})")
        i  $\leftarrow$  i + 1
        j  $\leftarrow$  i + 1

function show_monster ()
    i  $\leftarrow$  0
    j  $\leftarrow$  1
    while monster_inv_db['user_id'][i]  $\neq$  current_user do
        i  $\leftarrow$  i + 1
        pass
    while monster_inv_db['user_id'][i] == current_user do
        output (f"{j}. {get_monster(monster_inv_db['monster_id'][i],
monster_inv_db['level'][i])}")
        i  $\leftarrow$  i + 1
        i  $\leftarrow$  i + 1

function get_start_index ()
    i  $\leftarrow$  0
    while db['user_id'][i]  $\neq$  current_user do
        i  $\leftarrow$  i + 1
    j  $\leftarrow$  i
     $\rightarrow$  (j)

function show_inventory ()
    output ("Potion yang kamu miliki: ")
    show_items()
    output ("Monster yang kamu miliki:")
    show_monsters()
```

9. F08 – Battle

```
ALGORITMA
from src.x01 import *
from src.F00_RandomNumberGenerator import *
from src.F14_Load import *
from src.B03_MonsterBall import *
from src.F05_Monster import *
from src.F06_Potion import *
from src.F07_Inventory import *
from colorama import *
from global_var import *

function show_state()
    output(f"{type1:<17} | {type2:>17}")
    output(f"HP      / ATK / DEF | HP      / ATK / DEF")
    output(f"{hp1:<5} / {atk1:<3} / {def1:<3} | {hp2:>5} / {atk2:>3} / {def2:>3}")

function use_potion()
    output("Ramuan-ramuan yang kamu miliki saat ini: ")
    show_items(item_inv_db, user_id)
    while True do
        pot_choice ← str(input("Ramuan apa yang ingin kamu pilih? Pilih
dengan angka. Ketik 'cancel' untuk batal: ")).lower()
        if pot_choice == "cancel" then
            output("Kamu kembali ke pertarungan!")
            break
        else if
is_numerical(pot_choice) then
            output("Pilihan bukan angka!")
        else:
            pot_index ← integer(pot_choice) - 1 +
get_start_index(item_inv_db, user_id)
            if item_inv_db["user_id"][pot_index] != user_id then
                output
("Pilihan di luar jangkauan!")
            else:
                if item_inv_db["type"][pot_index] == "strength":
                    your_atk = strength_potion(your_atk)
                    item_inv_db["quantity"][pot_index] -= 1
                    break
                else if     else if
item_inv_db["type"][pot_index] == "resilience":
                    your_def = resilience_potion(your_def)
                    item_inv_db["quantity"][pot_index] -= 1
                    break
                elif item_inv_db["type"][pot_index] ==
"healing":
                    else if
item_inv_db["type"][pot_index] == "healing":
                        your_hp = healing_potion(your_hp, your_monster[3])
                        item_inv_db["quantity"][pot_index] -= 1
                        break
                    else:
                        else
                        output("Ramuan tidak valid!")
                →(item_inv_db, your_atk, your_def, your_hp)

function battle ()
    random_index ← int(rng(0, len(monster_db['id']), int(time.time())))
    {Meminta index untuk monster random}
```



```

enemy_monster ← attribute_monster(random_index + 1, enemy_level,
monster_db)
enemy_type ← enemy_monster[0]
enemy_atk ← enemy_monster[1]
enemy_def ← enemy_monster[2]
enemy_hp ← enemy_monster[3]
output (f"{enemy_type} mendekat!")
output (f"HP: {enemy_hp}. DEF: {enemy_def}. ATK: {enemy_atk}.")

output ("Monster-monster yang kamu miliki saat ini: ")
show_monsters(monster_inv_db, monster_db, user_id)
your_index = int(input("Monster apa yang ingin kamu pilih? ")) - 1 +
get_start_index(monster_inv_db, user_id)
your_monster_idx ← monster_inv_db["monster_id"][your_index]
your_monster ← attribute_monster(your_monster_idx,
monster_inv_db["level"][your_index], monster_db)
your_type ← your_monster[0]
your_atk ← your_monster[1]
your_def ← your_monster[2]
your_hp ← your_monster[3]
output (f"Kamu memilih {your_type}.")
output (f"HP: {your_hp}. DEF: {your_def}. ATK: {your_atk}.")

damage_dealt ← 0
damage_received ← 0

isBattle ← True
while isBattle == True do
    while True do
        output ("Pilihan Aksi")
        output ("1. Attack")
        output ("2. Use Potion")
        output ("3. Monsterball")
        output ("4. Escape")
        action ← input("Apa yang akan kamu lakukan? (Ketik pilihanmu!)")
    match action
        case "1":
            your_damage ← atk_result(your_atk, enemy_def)
            damage_dealt ← damage_dealt + your_damage
            enemy_hp ← enemy_hp - your_damage
            break
        case "2":
            item_inv_db, your_atk, your_def, your_hp ←
            use_potion(item_inv_db, user_id, your_monster,
            your_atk, your_def, your_hp)
            break
        case "3":
            if check_monsterball(item_inv_db, user_id) then
                item_inv_db ← use_monsterball(item_inv_db,
                user_id)
                if monsterball_success(enemy_level) then
                    print("Monster berhasil
tertangkap!")

```

```

monster_inv_db ← add_monster(user_id,
random_index, enemy_level, monster_inv_db)
enemy_hp ← 0

    else
        output ("Monster lepas!")
    else
        output ("Kamu tidak
        memiliki monsterball!")
    break
case "4":
    if battle_type == "wild" then
        isBattle == False
        output ("Kamu berhasil kabur ...")
        →(monster_inv_db, item_inv_db, oc, damage_dealt,
        damage_received)
    else
        output ("Tidak bisa kabur dari pertarungan di
        arena!")
    break
case
    output ("Pilihan tidak valid! \n")

enemy_damage ← atk_result(enemy_atk, your_def)
damage_received ← damage_received enemy_damage
enemy_hp ← enemy_hp - enemy_damage
your_hp ← your_hp - enemy_damage

if enemy_hp <= 0 then
    reward ← rng(50, 100, int(time.time()))
    oc ← oc + reward
    isBattle ← False
    output (f"Selamat, kamu menang! Kamu memperoleh {reward}
    OWCA Coins.")
    if battle_type == "wild" then
        →(monster_inv_db, item_inv_db, oc, damage_dealt,
        damage_received)
    else {battle_type == "arena" }
        →(monster_inv_db, item_inv_db, oc, True, damage_dealt,
        damage_received)
else if your_hp <= 0 then
    output ("Monstermu tumbang. Kamu kalah.")
    isBattle ← False
    if battle_type == "wild" then
        →(
        monster_inv_db, item_inv_db, oc, damage_dealt,
        damage_received)
    else {battle_type == "arena"}
        →(
        monster_inv_db, item_inv_db, oc, False, damage_dealt,
        damage_received)
    else
        print(f"Pertarungan terus
        berlangsung!
        print output
        (f"Pertarungan terus berlangsung! \n HP Monster-mu:
        {your_hp} \n HP Monster Musuh: {enemy_hp}")

```

10. F09 – Arena

ALGORITMA

```
from src.F05_Monster import *
import sys, os
from global_var import *
from src.x01 import *
from src.F00_RandomNumberGenerator import *
from src.F08_Battle import *

function arena()
    total_dealt  $\leftarrow$  0
    total_received  $\leftarrow$  0
    reward  $\leftarrow$  0
    stage  $\leftarrow$  1
    defeat  $\leftarrow$  False
    while defeat == False and stage <= 5 do
        monster_inv_db, item_inv_db, oc, victory, damage_dealt,
        damage_received = battle(monster_db, monster_inv_db, user_id, stage,
        item_inv_db, oc, "arena")

        total_dealt  $\leftarrow$  total_dealt + damage_dealt
        total_received  $\leftarrow$  total_received + damage_received

        if victory == True then
            if stage == 1:
                reward  $\leftarrow$  reward + 30
            else if stage == 2:
                reward  $\leftarrow$  reward + 50
            else if stage == 3:
                reward  $\leftarrow$  reward + 100
            else if stage == 4:
                reward  $\leftarrow$  reward + 200
            else if stage == 5:
                reward  $\leftarrow$  reward + 400
            stage  $\leftarrow$  stage + 1
        else
            output ("Yah, kamu kalah.")
            defeat  $\leftarrow$  True

    if defeat == False then
        output ("Selamat, kamu berhasil menyelesaikan arena!")
    else
        output ("Kamu mencapai stage", stage, "dari 5.")

    output (f"Kamu mendapatkan {reward} OC.")
    output (f"Total damage yang kamu berikan: {total_dealt}")
    output (f"Total damage yang kamu terima: {total_received}")
     $\rightarrow$  (monster_inv_db, item_inv_db, oc + reward)  $\rightarrow$  (monster_inv_db,
    item_inv_db, oc + reward)
```

```

ALGORITMA
ALGORITMA

from global_var import *
from src.x01 import *

function add_monster ()
    i ← 0
    temp_user ← integer(user_id)
    temp_index ← integer (monster_index + 1)
    temp_level ← integer (enemy_level).
    while monster_inv_db["user_id"][i] <= user_id and i <
len(monster_inv_db["user_id"])do

        i ← i + 1

        (monster_inv_db["user_id"][i], temp_user) ← (temp_user,
monster_inv_db["user_id"][i])
        (monster_inv_db["monster_id"][i], temp_index) ← (temp_index,
monster_inv_db["monster_id"][i])
        (monster_inv_db["level"][i], temp_level) ← (temp_level,
monster_inv_db["level"][i])
        i ← i + 1

    while i < (len(monster_inv_db["user_id"]))do

        (monster_inv_db["user_id"][i], temp_user) = (temp_user,
monster_inv_db["user_id"][i])
        (monster_inv_db["monster_id"][i], temp_index) = (temp_index,
monster_inv_db["monster_id"][i])
        (monster_inv_db["level"][i], temp_level) = (temp_level,
monster_inv_db["level"][i])
        i ← i + 1

    monster_inv_db["user_id"].append(temp_user)
    monster_inv_db["monster_id"].append(temp_index)
    monster_inv_db["level"].append(temp_level)
    → (monster_inv_db)

function add_item ()
    i          i = i + 1
    i          i = i + 1
    i = i + 1
    i = i + 1

    if item_inv_db["type"][i] == item:
    if item_inv_db["type"][i] == item:

        found ← True
        found ← True found = True found ← True
        found ← True found = True found ← True
        found ← True
        i ← i + 1
        i ← i + 1

```

12. F11 – Laboratory

ALGORITMA

```
from global_var import *
```

```
function upgrade mosnter ()
    selected_monster = monsters[i]
    if selected_monster['level'] >= 5 then
        output("Maaf, monster yang Anda pilih sudah memiliki level
maksimum.")
    → owca_coins
    upgrade_costs ← [300, 500, 800, 1000]
    current_level ← selected_monster['level']
    upgrade_cost ← upgrade_costs[current_level - 1]
    output(f"\n{selected_monster['name']} akan di-upgrade ke level
{selected_monster['level'] + 1}.")
    output(f"Harga untuk melakukan upgrade {selected_monster['name']}
adalah {upgrade_cost} OC.")

    if owca_coins >= upgrade_cost then
        confirm ← input(">>> Lanjutkan upgrade (Y/N): ")
        if confirm.lower() == 'y' then
            selected_monster['level'] ← selected_monster['level'] + 1
            owca_coins ← owca_coins - upgrade_cost
            output(f"Selamat, {selected_monster['name']} berhasil di-
upgrade ke level {selected_monster['level']}!")
        else if confirm.lower() == 'n' then
            output ("Upgrade dibatalkan.")
        else
            output , ("Pilihan tidak valid. Silakan masukkan Y atau N.")
    else
        output("Maaf, OC Anda tidak mencukupi untuk melakukan upgrade.")
    → owca_coins
```

```
function laboratory ()
```

```
    while True do
```

```
        output("Selamat datang di Lab Dokter Asep !!!\n")
```

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

```
        output (showMonsters())
```

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

```
        output ("1. Level 1 -> Level 2: 300 OC")
```

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

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

```
        output ("4. Level 4 -> Level 5: 1000 OC")
```

```
        user_input ← input(">>> Pilih monster (atau ketik 'exit' untuk
keluar): ")
```

```
        if user_input == 'exit' then
            break
```

```
        else if user_input.isdigit() then
```

```
            i ← integer(user_input) - 1
```

```
            if 1 <= i + 1 <= len(monsters) then
```

```
                owca_coins ← upgrade_monster(monsters, i, owca_coins)
```

```
            else
```

```
                output ("Pilihan tidak valid, silakan pilih nomor monster
yang valid.")
```

```
        else
```

```
            output ("Pilihan tidak valid, silakan pilih nomor monster
yang valid.")
```

```

function getMonster (i)
    → (f""{monster_db["type"][i]} (Level:
{monster_inv_db["level"][i]}) """)

function showMonsters ()
    i ← 0
    if i < len(monster_inv_db['user_id']) then
        while monster_inv_db['user_id'][i] ≠ current_user do
            i ← i + 1
        pass
        while monster_inv_db['user_id'][i] == current_user do
            i ← i + 1
        → (f"{getMonster(monster_inv_db['monster_id'][i])}")

```

13. F12 - Shop Management

```

ALGORITMA
from src.x01 import *
from src.F10_ShopCurrency import *
from src.F05_Monster import *

function input_validation(masukan, validation, error_message):
    while True do
        value ← input(masukan)
        if validation(value) then
            → value
        else
            print(error_message)

function add_monster_shop(monster_shop_db, monster_id, stock, price):
    if monster_id not in monster_shop_db["monster_id"] then
        monster_shop_db["monster_id"].append(monster_id)
        monster_shop_db["stock"].append(stock)
        monster_shop_db["price"].append(price)
        output("Monster ditambahkan ke shop")
    else
        output("Monster sudah ada di shop!")
    → (monster_shop_db)

function remove_monster_shop(monster_shop_db, monster_id):
    if monster_id in monster_shop_db['monster_id'] then
        idx ← get_idx(monster_shop_db['monster_id'],
monster_id)
        idx ← get_idx(monster_shop_db['monster_id'],
monster_id)
        monster_db["id"] ← remove_ele(idx,
monster_db["monster_id"])
        monster_db["id"] ← remove_ele(idx,
monster_db["monster_id"])
        monster_db["type"] ← remove_ele(idx,
monster_db["stock"])
        monster_db["type"] ← remove_ele(idx,
monster_db["stock"])
        monster_db["atk_power"] ← remove_ele(idx,
monster_db["price"])
        monster_db["atk_power"] ← remove_ele(idx,
monster_db["price"])
        output("Monster dihapus dari shop.")
    else
        output
        output
("ID monster tidak ditemukan!")

```

ALGORITMA

```
else else
    output("ID monster tidak ditemukan!")

function add_item_shop()
    item_shop_db['type'].append(item_type)
    item_shop_db['stock'].append(stock)
    item_shop_db['price'].append(price)
    output(f"{item_type.capitalize()} berhasil ditambahkan.")
    →(item_shop_db)

function remove_item_shop()
    if item_type in item_shop_db['type'] then
        index ← item_shop_db['type'].index(item_type)
        key traversal [item_shop_db.keys()]
        del item_shop_db[key][index]
        output("Item dihapus dari toko.")
    else
        output ("Item tidak ditemukan.")
    →(item_shop_db)

function change_item_shop()
    if item_type in monster_shop_db['monster_id'] then
        idx ← get_idx(item_shop_db["type"][item_type])
        item_shop_db['stock'][idx] ← stock
        item_shop_db['price'][idx] ← price
        output ("Data di shop berhasil diubah.")
    else
        output ("Item tidak ditemukan!")
    →(item_shop_db)

function item_shop_management():
    item_shop_list(item_shop_db)
    output("Mau tambah, hapus, atau ubah item di shop?")
    while True do
        choice ← string(input("Ketik 'tambah' untuk menambahkan item,
        'hapus' untuk menghapus item, 'ubah' untuk mengubah item, dan 'exit'
        untuk keluar.")).lower())
        if is_numerical(choice) then
            output("Input harus berupa teks!")
        else
            if choice == "tambah" then
                item_type ← input_validation("Masukkan tipe item: ",
                is_string, "Tipe harus berupa teks!")
                stock ← input_validation("Masukkan stock item: ",
                is_numerical, "Stock harus berupa angka!")
                price = input_validation("Masukkan harga item: ",
                is_numerical, "Harga harus berupa angka!")
                item_shop_db ← add_item_shop(item_shop_db, item_type,
                stock, price)
            else if choice == "hapus":
                item_type ← input_validation("Masukkan tipe item yang
                ingin dihapus: ", is_string, "Tipe item harus berupa teks!")
                item_shop_db ← remove_item_shop(item_shop_db, item_type)
```

```

        else if choice == "ubah" then
            monster_id ← input_validation("Masukkan ID monster yang
            ingin diubah: ", is_numerical, "ID monster harus berupa
            angka!")
            stock ← input_validation("Masukkan stock monster baru: ",
            is_numerical, "Stock monster harus berupa angka!")
            price ← input_validation("Masukkan harga monster baru: ",
            is_numerical, "Harga monster harus berupa angka!")
            monster_shop_db ← change_monster_shop(monster_shop_db,
            monster_id, stock, price)
        elif choice ==
"exit":
            else print("Mau edit database monster
atau shop?")
            elif choice == "exit":
                else if choice == "exit"
then
                elif choice ==
"exit":
                    else
                    else:
                        else
"exit":
                    else
                    else:
                        else
                        elif choice == "exit":
                            else
                            else:
                                else
                                print("Input tidak valid!")
                                output ("Input
tidak valid!")
                                return(monster_shop_db) → (monster_shop_db)
                                def shop_management(monster_shop_db, item_shop_db):
                                print("Selamat datang di shop management!")
                                output ("Selamat datang
di shop management!")
                                print("Mau edit database monster atau shop?")
                                output ("Mau edit
database monster atau shop?")
                                while True:
                                    while True:
                                        while True do
                                        while True:
                                            choice = str(input("Ketik 'monster' untuk mengelola
monster, 'item' untuk mengelola item, dan 'exit' untuk
keluar.").lower().lower())
                                            choice ← string(input("Ketik 'monster'
untuk mengelola monster, 'item' untuk mengelola item, dan 'exit'
untuk keluar.").lower())
                                            if is_numerical(choice):
                                                if is_numerical(choice):
                                                    if
is_numerical(choice) then
                                                    if is_numerical(choice):
                                                        print("Input harus berupa teks!")
                                                        output ("Input
harus berupa teks!")
                                                        else:
                                                            else:
                                                                else
                                                                else:
                                                                    if choice == "monster":
                                                                        if choice ==
"monster":
                                                                            if choice == "monster" then
                                                                            if choice == "monster":
                                                                                else if if
choice == "item": then
                                                                                else if
choice == "exit": then
                                                                                else:
                                                                                    print("Input if Input tidak valid!")

function shop_management(monster_shop_db, item_shop_db):
    print("Selamat datang di shop management!")
    print("Mau edit database monster atau shop?")
    while True:
        choice = str(input("Ketik 'monster' untuk mengelola monster, 'item'
untuk mengelola item, dan 'exit' untuk keluar.").lower())

```


14. F13 - Monster Management

ALGORITMA

```
from F05_Monster import *
import sys, os
from global_var import *
from x01 import *

function monsterManagement()
    output ("""
        SELAMAT DATANG DI DATABASE PARA MONSTER !!!
        1. Tampilkan semua Monster
        2. Tambah Monster baru
        """)
    pilihan  $\leftarrow$  integer(input(">>> Pilih aksi : "))
    if pilihan == 1:
        if pilihan == 1 then
            output (""" ID | Type | ATK Power | DEF
                Power | HP | """)
            i traversal[len(monster_db["type"])]

            output (f"""
                {monster_db["id"][i]} |
                {monster_db["type"][i]} |
                {monster_db["atk_power"][i]} |
                {monster_db["def_power"][i]} |
                {monster_db["hp"][i]} | """)
        else if pilihan == 2 then
            else if pilihan == 2 then
                addType  $\leftarrow$  ''
                addATKP  $\leftarrow$  ''
                addDEFP  $\leftarrow$  ''
                addHP  $\leftarrow$  ''
                output ("Memulai pembuatan monster baru")
                added  $\leftarrow$  False
                while added == False do
                    if addType == '' then
                        addType  $\leftarrow$  input(">>> Masukkan Type / Nama : ")
                    else if addType not in monster_db["type"] then
                        monster_db["id"].append(last(monster_db["id"])+1)

                added  $\leftarrow$  True
            else
                output ("Nama sudah terdaftar, coba lagi")
```

15. F14 – Load

ALGORITMA

```
from src.x01 import *
from global_var import *

function read_header ()
    file  $\leftarrow$  open(path, 'r')
    line traversal [file]
         $\rightarrow$  strip_str(line)
    File.close ()
     $\rightarrow$  ""

function load_csv ()
    file  $\leftarrow$  open(path, 'r')
    headers  $\leftarrow$  split_str(read_header(path))
    data  $\leftarrow$  {header : []header traversal [headers]}

    {Menambahkan data pada tiap kolom satu per satu}
    line traversal [file]
        entries  $\leftarrow$  split_str(strip_str(line))
        {Error ketika ada panjang baris yang tidak sama dengan header}
        if len(entries)  $\neq$  len(headers) then
            raise ValueError("Ada data yang kosong atau melebihi kolom header!")
        {Memproses data csv untuk baris-baris setelah header}
        {Mengambil identitas dari identifier yang ditentukan}
        if entries  $\neq$  headers then
            i traversal [len(headers)]
                header, entry  $\leftarrow$  headers[i], entries[i]
                data[header].append(entry)
    file.close ()
     $\rightarrow$  data

function load_data()
    user_db  $\leftarrow$  load_csv('data/' + csv_dir + '/user.csv')
    monster_db  $\leftarrow$  load_csv('data/' + csv_dir + '/monster.csv')
    monster_shop_db  $\leftarrow$  load_csv('data/' + csv_dir + '/monster_shop.csv')
    monster_inv_db  $\leftarrow$  load_csv('data/' + csv_dir +
'/monster_inventory.csv')
    item_shop_db  $\leftarrow$  load_csv('data/' + csv_dir + '/item_shop.csv')
    item_inv_db  $\leftarrow$  load_csv('data/' + csv_dir + '/item_inventory.csv')
     $\rightarrow$  [user_db, monster_db, monster_shop_db, monster_inv_db, item_shop_db,
item_inv_db]
```

16. F15 – Save

ALGORITMA

```
import os,sys

function save ()
    csv ← open(path, "w")
    db_width ← len(db)
    headers ← []
    headers_row ← ""

    i ← 0
    header traversal [db.keys()]
    headers.append(header)

    if i ≠ db_width - 1 do
        headers_row ← f"{headers_row}{header};"
    else
        headers_row ← f"{headers_row}{header}"

    i ← i + 1
    csv.write(headers_row + "\n")

    db_length ← len(db[headers[0]])
    i traversal[db_length]
        row ← ""
        j traversal [db_width]
            if j ≠ db_width - 1:
                row ← f"{row}{db[headers[j]][i]};"
            else
                row ← f"{row}{db[headers[j]][i]}"
        csv.write(row)
        if i ≠ db_length - 1 do
            csv.write("\\n")
    csv.close()
```

17. F16 – Exit

ALGORITMA

```
from src.F15Save import *
from src.x01 import *
from global_var import *

function exit()
    global user_db, a, monster_shop_db, monster_inv_db, item_shop_db,
item_inv_db
    save_prompt ← input("Apakah Anda ingin menyimpan progres Anda? (y/n)")
    match lower(save_prompt):
        case "y"
            output("Menyimpan database ke dalam csv... [user.csv]")
            save(user_db, 'data/' + dir + '/user.csv')
            output("Menyimpan database ke dalam csv... [monster.csv]")
            save(monster_db, 'data/' + dir + '/monster.csv')
            output("Menyimpan database ke dalam csv...
[monster_shop.csv]")
            save(monster_shop_db, 'data/' + dir + '/monster_shop.csv')
            output("Menyimpan database ke dalam csv...
[monster_inventory.csv]")
            save(monster_inv_db, 'data/' + dir + '/monster_inventory.csv')
            output("Menyimpan database ke dalam csv... [item_shop.csv]")
            save(item_shop_db, 'data/' + dir + '/item_shop.csv')
            output("Menyimpan database ke dalam csv...
[item_inventory.csv]")
            save(item_inv_db, 'data/' + dir + '/item_inventory.csv')
            output ("Proses penyimpanan selesai.")
            time.sleep(1)
            output("Sampai jumpa di lain waktu :D")
        case "n":
            output("Sampai jumpa di lain waktu :D")
        case _:
            remove_nth_line(1)
            exit(dir)
```

18. B03 - Monster Ball

ALGORITMA

```
from src.F00_RandomNumberGenerator import *
from src.F10_ShopCurrency import *
from global_var import *

function mosnterball_success ():
    result  $\leftarrow$  rng (0, 20, time.time())
    if level == 1 then
        if 0 < result < 15 then
             $\rightarrow$  (True)
        else:
             $\rightarrow$  (False)
    else if level == 2 then
        if 0 < result < 10 then
             $\rightarrow$  (True)
        else:
             $\rightarrow$  (False)
    else if level == 3 then
        if 0 < result < 5 then
             $\rightarrow$  (True)
        else:
             $\rightarrow$  (False)
    else if level == 4 then
        if 0 < result < 2 then
             $\rightarrow$  (True)
        else:
             $\rightarrow$  (False)
    else
        if 0 < result < 1 then
             $\rightarrow$  (True)
        else:
             $\rightarrow$  (False)

function check_monsterball ()
    i  $\leftarrow$  0
    while i < len(item_inv_db["type"]) do
        if item_inv_db["type"][i] == "monsterball"
        and item_inv_db["user_id"][i] == user_id then
             $\rightarrow$  (True)
        else
            i  $\rightarrow$  i + 1
     $\rightarrow$  (False)

Function use_monsterball ()
    i  $\leftarrow$  0
    while i < len(item_inv_db["type"]) do
        if item_inv_db["type"][i] == "monsterball" and
item_inv_db["user_id"][i] == user_id then
            item_inv_db["quantity"][i]  $\leftarrow$  item_inv_db["quantity"][i] - 1
            if item_inv_db["quantity"][i] == 0 then
                item_inv_db = remove_item(i, item_inv_db)
            break
```

```

        else
            i ← i + 1
    →(item_inv_db)

```

B04 - JACKPOT!

ALGORITMA

```

from F00_RandomNumberGenerator import *
from F14_Load import *
from global_var import *
from F10_ShopCurrency import add_monster

function jackpot ()
    {Cek OC pengguna}
    user ← username
    current_user ← get_idx(user, user_db["username"])
    owca_coins ← user_db["oc"][current_user]

    {Cek jika oc cukup untuk main}
    if owca_coins < 1000 then
        output ("Maaf, anda tidak memiliki cukup OC untuk bermain JACKPOT.")
        →
        {Kurangi OC untuk main}
        user_db["oc"][current_user] ← user_db["oc"][current_user]-1000

        {Harga item yang ada dan Namanya}
        items ← ["Topi", "Pedang", "Koin", "Potion", "Monster"]
        item_values ← [50, 100, 200, 300, 500]

        {Generate 3 item random}
        item1_idx ← rng(0, len(items), integer(working_time))
        item2_idx ← rng(0, len(items), integer (working_time))
        item3_idx ← rng(0, len(items), integer (working_time))

        {Mnedapatakan nama item}
        item1 ← items[item1_idx]
        item2 ← items[item2_idx]
        item3 ← items[item3_idx]
        {Print nama item yang pengguna menang}
        output (f"Anda Mendapatkan:\n{item1} | {item2} | {item3}")

        {Check jika pengguna mnedapat jackpot}
        if item1 == item2 == item3 then
            monster ← monster_db["type"][item2_idx] {Dapatakan nama monster}
            enemy_level ← 1 {Buat jadi level 1}
            {Tambahkan monster ke user inventory}
            add_monster (current_user, item2_idx, enemy_level,
monster_inv_db)

```

```

    {Print ouput}
    output(f"JACKPOT!!! Selamat, Anda mendapatkan monster
{monster}.")

else
    {Hitung oc yang dimenangkan}
    item1_value ← item_values[item1_idx]
    item2_value ← item_values[item2_idx]
    item3_value ← item_values[item3_idx]
    total_oc ← item1_value + item2_value + item3_value
    {Tambahkan oc ke oc pengguna}
    user_db["oc"][current_user] ← user_db["oc"][current_user] +
total_oc
    {Print output}
    output(f"Didapat {total_oc} OC.")

```

B05 - Peta Kota Danville

ALGORITMA

```

function read_map ()
    arr ← ["" for j in range(M+2)] for i in range(N+2)]
    {Array diinisialisasi dengan ruang tambahan untuk garis pembatas tepian map}
    with open(fn, 'r') as map
        p traversal[M+2]
        arr[0][p] ← "*"
        i ← integer(1)
        line traversal [map]:
            arr[i][0] ← "*"
            j ← integer(1)
            symbol traversal line.strip():
                arr[i][j] ← symbol
                j ← j + 1
            arr[i][M+1] ← "*"
            i ← i + 1
        p traversal[M+2]:
            arr[N+1][p] ← "*"
    arr[posx][posy] ← "P" {Posisi pemain diinisialisasi}
    → (arr)

```

```

function print_map()
  i traversal[M+2]
    j traversal[N+2]
      obj ← arr[i][j]
      if obj == "#" then
        output(" ", end="")
      else
        output(f"{arr[i][j]} ", end="")
    output()

```

```

function moveUp()
  if (worldmap[posy-1][posx]) ≠ '#' then
    output("Ada penghalang!")
  else
    worldmap[posy][posx] = "#"
    posy ← posy - 1
    worldmap[posy][posx] = "P"
    →(worldmap, posx, posy)
    →(worldmap, posx, posy)

```

```

function moveDown(worldmap, posx, posy):
  if (worldmap[posy+1][posx]) ≠ '#' then
    print("Ada penghalang!")
  else
    worldmap[posy][posx] ← "#"
    posy ← posy + 1
    worldmap[posy][posx] ← "P"
    →(worldmap, posx, posy)

```

```

function moveLeft(worldmap, posx, posy):
  if (worldmap[posy][posx-1]) ≠ '#' then
    output("Ada penghalang!")
  else
    worldmap[posy][posx] ← "#"
    posx ← posx - 1
    worldmap[posy][posx] ← "P"
    →(worldmap, posx, posy)

```

```

function moveRight(worldmap, posx, posy):
  if (worldmap[posy][posx+1]) ≠ '#' then
    output("Ada penghalang!")
  else
    worldmap[posy][posx] ← "#"
    posx ← posx + 1
    worldmap[posy][posx] ← "P"
    →(worldmap, posx, posy)

```



```

function checkProximity()
  symbol ← action.title()[0]
  {Aksi dan objek dicocokkan dengan mengambil huruf pertama di aksi,
   karena pada data map, objek disimpan sebagai huruf awal}
  i traversal [-1, 2]
    if worldmap[posy-1][posx+i] == symbol then
      →(True)
  if worldmap[posy][posx-1] == symbol then
    return(True)
  if worldmap[posy][posx+1] == symbol then
    →(True)
  i traversal [-1, 2]
    if worldmap[posy+1][posx+i] == symbol then
      →(True)
  →(False)

```

Lampiran

1. Lampiran Program

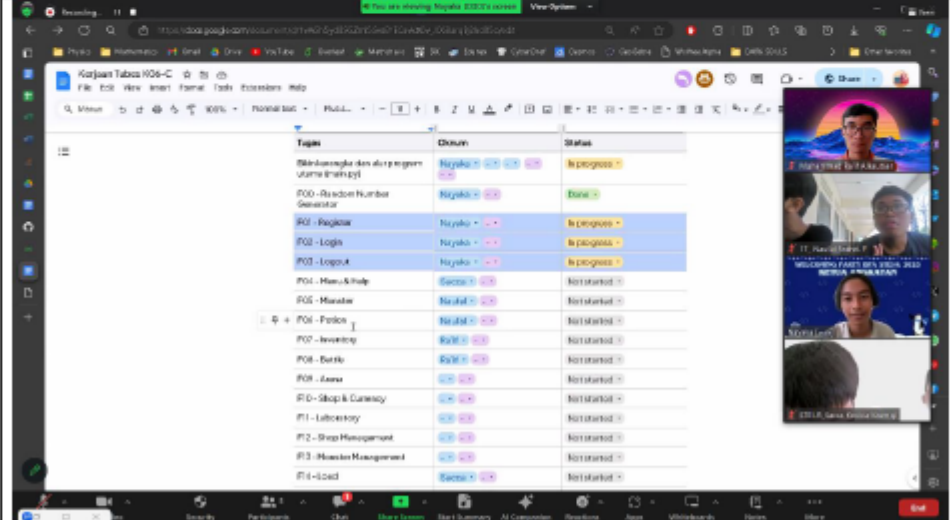
- 2.
3. Lampiran Form Asistensi

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

Nomor Asistensi	:	<u>1</u>												
No. Kelompok/Kelas	:	<u>K-06-C</u>												
Tanggal asistensi	:	<u>1 Mei 2024</u>												
Anggota kelompok	<table border="1"> <thead> <tr> <th></th> <th>NIM / Nama (Hanya yang Hadir)</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>19623116 / Zulfaggar Nayaka Athadiansyah</td> </tr> <tr> <td>2</td> <td>19623296 / Muhammad Ra'if Alkautsar</td> </tr> <tr> <td>3</td> <td>19623076 / Daniel A. M. Sipayung</td> </tr> <tr> <td>4</td> <td>16523106 / Naufal Fakhri Fadhlurrahman</td> </tr> <tr> <td>5</td> <td>16523146 / Sacca Kovida Kasmaji</td> </tr> </tbody> </table>			NIM / Nama (Hanya yang Hadir)	1	19623116 / Zulfaggar Nayaka Athadiansyah	2	19623296 / Muhammad Ra'if Alkautsar	3	19623076 / Daniel A. M. Sipayung	4	16523106 / Naufal Fakhri Fadhlurrahman	5	16523146 / Sacca Kovida Kasmaji
	NIM / Nama (Hanya yang Hadir)													
1	19623116 / Zulfaggar Nayaka Athadiansyah													
2	19623296 / Muhammad Ra'if Alkautsar													
3	19623076 / Daniel A. M. Sipayung													
4	16523106 / Naufal Fakhri Fadhlurrahman													
5	16523146 / Sacca Kovida Kasmaji													
Asisten pembimbing	<table border="1"> <thead> <tr> <th></th> <th>NIM / Nama</th> </tr> </thead> <tbody> <tr> <td></td> <td>13521129 / Chiquita Ahsanunnisa</td> </tr> </tbody> </table>			NIM / Nama		13521129 / Chiquita Ahsanunnisa								
	NIM / Nama													
	13521129 / Chiquita Ahsanunnisa													

Catatan Asistensi:

Rangkuman Diskusi
Bagian F05 dan F06 berisi database dan fungsi-fungsi yang digunakan untuk mengakses database tersebut. Colorama boleh digunakan supaya bisa terdapat warna di terminal. Terkait monster dan potion, diperbolehkan untuk berkreasi asal tidak mengubah mekanisme game. Penggunaan time diperbolehkan untuk seed pada fungsi <i>random number generator</i> . Pembagian tugas sudah dilakukan dan tugas besar sudah mulai dikerjakan.
Tindak Lanjut
Fungsi-fungsi pembantu, fungsi login, register, dan map akan segera dikerjakan. Pengerjaan fungsi-fungsi lain juga akan segera dimulai.
Dokumentasi



Gambar 1. Form Pertemuan Asistensi Pertama

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

Nomor Asistensi : 2
 No. Kelompok/Kelas : K-06-C
 Tanggal asistensi : 1 Mei 2024

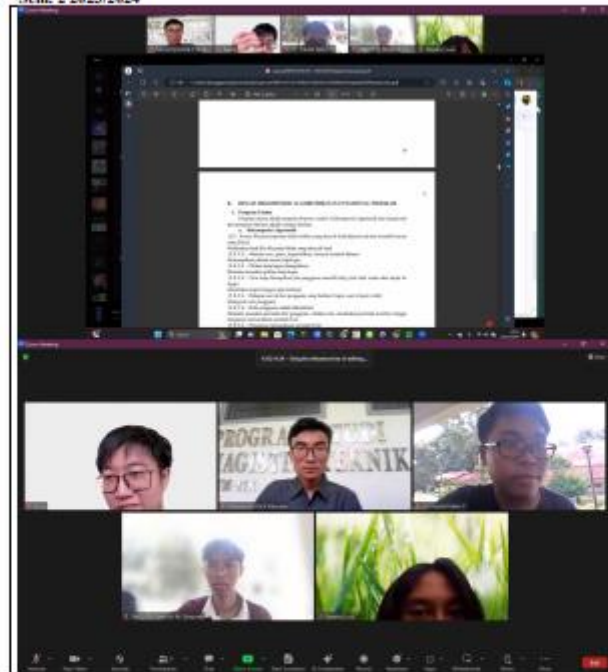
Anggota kelompok	NIM / Nama (Hanya yang Hadir)
1	196231116 / Zulfaqqar Nayaka Athadiansyah
2	19623296 / Muhammad Ra'if Alkautsar
3	19623076 / Daniel A. M. Sipayung
4	16523106 / Naufal Fakhri Fadhlurrahman
5	16523146 / Sacca Kovida Kasmaji
Asisten pembimbing	NIM / Nama
	13521129 / Chiquita Ahsanunnisa

Catatan Asistensi:

Rangkuman Diskusi
Meeting dimulai dengan progress report. Sebagian besar fungsi kurang lebih telah dituntaskan, serta fungsi-fungsi dan prosedur sudah mulai digabungkan ke main.py. Asisten menyarankan supaya nama-nama file source code diubah, semisal dari f01 menjadi f01_register supaya lebih jelas. Asisten juga mengimbau supaya pengerjaan tugas besar dipercepat karena tenggatnya hanya tinggal tujuh hari lagi.
Ada sejumlah pertanyaan yang ditanyakan oleh anggota, antara lain:
- Apakah flowchart dibutuhkan? Tidak dibutuhkan.
- Apa perbedaan kamus data dengan dekomposisi algoritmik dan fungsional? (Diberikan contoh oleh asisten lewat sharescreen.)
Tindak Lanjut
Pengerjaan tugas besar akan dipercepat dan prosedur serta fungsi akan mulai digabungkan di main.py. Laporan akan mulai ditulis.
Dokumentasi

Gambar 2. Form Pertemuan Asistensi Kedua Bagian Satu

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



Gambar 3. Form Pertemuan Asistensi Kedua Bagian Kedua